UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA MECÁNICA

ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

RECONOCIMIENTO Y CLASIFICACIÓN DE OBJETOS USANDO INTELIGENCIA ARTIFICIAL BASADA EN SVM Y VISIÓN ESTEREOSCÓPICA

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO MECATRÓNICO

ELVIS FRANKS CONDORI ARIAS

PROMOCIÓN 2010-I

LIMA – PERÚ

2013

Resumen

En esta tesis se desarrolla un sistema de reconocimiento y clasificación de objetos, el cual trata de emular la forma como los humanos percibimos la información visual mediante el uso de dos cámaras ópticas, que actúan como nuestros ojos, y un CPU que procesa la información en una “forma inteligente”. La compleja tarea de simular el sentido de la vista es dividida en un conjunto de tareas más simples, que abarcan desde la captura de las imágenes hasta el reconocimiento de objetos en la escena tridimensional.

Se utiliza visión estereoscópica para obtener información acerca de la profundidad de la escena a partir de un par estereoscópico, los procesos de segmentación utilizan esta información para obtener regiones de interés en el proceso de identificación de un objeto. El reconocimiento y clasificación de objetos se realiza mediante técnicas de inteligencia artificial ejecutadas en un computador. Específicamente en esta tesis se utiliza SVM, support vector machine, que es un método muy poderoso y que en pocos años desde su introducción ya ha superado a otras técnicas de inteligencia artificial como las redes neuronales.

Se presenta el desarrollo de los algoritmos utilizados en cada una de las fases de la tesis; algoritmos para la visión estereoscópica y realce de las características en las imágenes de entrada, considerados de bajo nivel; los algoritmos para segmentación y extracción de características, considerados de nivel intermedio; y finalmente los algoritmos de alto nivel, que realizan los procesos de reconocimiento y clasificación. El sistema de reconocimiento de objetos es entrenado mediante ejemplos previos, y se evalúa su comportamiento frente a nuevos objetos de la misma clase para los cuales ya ha sido entrenado.

A mis padres

Vidal y Natalia por su amor y soporte,

a mi hermana

Solansh Ruby,

a mi asesor

Ricardo Rodríguez Bustinza.

Contenido

Lista de figuras

Prólogo

La tecnología desarrollada por los seres humanos permite obtener y mejorar productos y servicios que están orientados a mejorar la calidad de vida de las personas, es por ello que la investigación tecnológica es importante a fin de contribuir con este objetivo. En la actualidad, podemos observar que una gran cantidad de productos que se utilizan diariamente toman el adjetivo de “inteligente”, desde smartphones, smartTVs, semáforos inteligentes, etc. Estos productos son el resultado de la tecnología desarrollada en los últimos años. Sin embargo, ¿Son realmente “inteligentes” todos estos nuevos equipos que intentan mejorar nuestra calidad de vida? En realidad no existe una escala objetiva para definir los niveles de inteligencia en las máquinas, por lo que no podríamos afirmar el grado de inteligencia que presentan estos equipos en la actualidad. El desarrollo de sistemas “inteligentes” ha recibido un impulso en los últimos años gracias a los paradigmas de programación que utilizan máquinas de aprendizaje para resolver problemas del mundo real.

En esta tesis se utilizan estos paradigmas de programación para realizar el reconocimiento y clasificación de objetos, por lo que podríamos decir que el sistema a implementarse presenta cierto grado de inteligencia. El desarrollo de sistemas

“inteligentes” tendrá probablemente un impacto notable sobre la humanidad en los próximos años, más de lo que ya nos ha cambiado la vida en la actualidad, es por ello que las investigaciones en este campo tienen una gran importancia debido al impacto que tienen sobre la vida de millones de personas. Esta tesis busca contribuir con investigaciones en la Universidad Nacional de Ingeniería en temas de inteligencia artificial aplicada al reconocimiento de objetos.

Capítulo 1

Introducción

1.1. Antecedentes

El reconocimiento de objetos, estructuras, entre otros, es una tarea fundamental que los humanos realizamos cada día de nuestras vidas, es por ello que varios proyectos de investigación se enfocan en desarrollar y construir máquinas y sistemas que realicen ésta tarea con mayor autonomía como los robots móviles para la exploración, sistemas de clasificación de productos en las industrias, equipos de entretenimiento para el hogar entre otros sistemas diversos. Ejemplos de estos trabajos son proyectos realizados por la NASA/JPL (JPL por su siglas en inglés, Jet Propulsion Laboratory) - Mars Exploration Rovers y el Mars Science Laboratory, Google - Search by Image, Microsoft - Kinect, NASA/DARPA - Robonaut, Honda - Asimo.

En el proyecto de la NASA/JPL – Mars Exploration Rovers, se desarrollaron dos robots autónomos llamados Spirit y Opportunity que llegaron a los lados opuestos del planeta Marte en enero del 2004, estos robots exploradores han

recorrido varios kilómetros sobre la superficie del planeta rojo con el objetivo de realizar geología de campo y observaciones atmosféricas entre otras metas. La parte de interés para esta tesis se refiere a la capacidad que tienen estos robots de percibir su entorno para la navegación y exploración planetaria mediante la visión estereoscópica.

La visión estereoscópica pasiva[1] autónoma brinda la habilidad de navegar en forma segura a través de terrenos no conocidos con potenciales obstáculos. El algoritmo que tienen implementados estos robots para la visión estereoscópica se divide en siete secciones principales que consisten en:

1. Redimensionar el tamaño de las imágenes para reducir la carga computacional.
2. Rectificar el par de imágenes para asegurar que las líneas epipolares[2] estén alineadas con la horizontal.
3. Calcular el laplaciano para remover la variación brusca en la intensidad de los pixeles.
4. Encontrar los puntos correspondientes en el par de imágenes realizando una correlación con ventana de 7x7.
5. Verificar las correspondencias halladas con algún tipo de control.
6. Eliminar las correspondencias incorrectas.
7. Finalmente mapear las disparidades encontradas a un mapa 3D usando el modelo geométrico de la cámara, [1].

[1] *La visión estereoscópica pasiva no actúa sobre la escena a diferencia de la activa que interviene externamente sobre la escena.*
[2] *Referente a la geometría epipolar que describe la relación entre dos imágenes de un par estereoscópico.*

En la Figura 1.1 se muestra la distribución de las cámaras que estos robots tienen incorporados para realizar su navegación autónoma, [2].

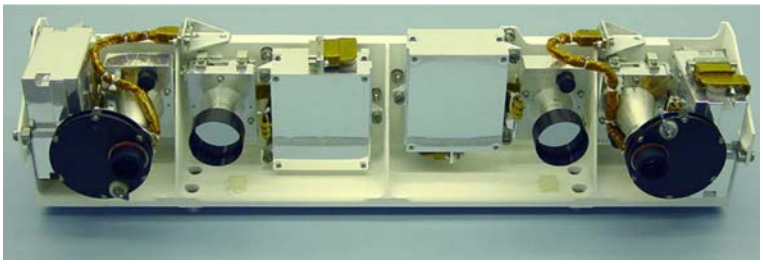
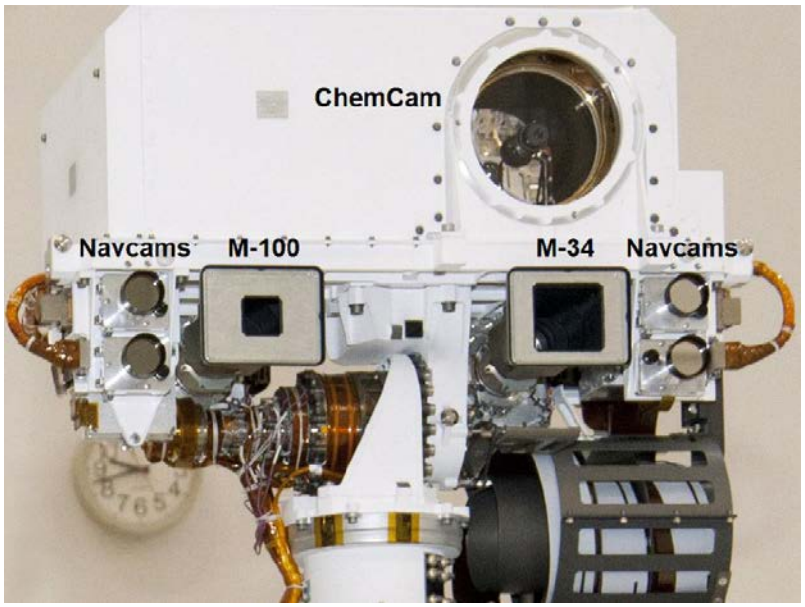


Figura 1.1*: Disposición de las cámaras utilizadas en los rovers Spirit y Opportunity.*

El Mars Science Laboratory es una misión de la NASA llevado a cabo por el JPL en un esfuerzo de largo plazo para la exploración robótica del planeta rojo. Esta misión aterrizó exitosamente al rover Curiosity el 06 de agosto del 2012 en el cráter Gale. El Curiosity fue diseñado para evaluar si Marte alguna vez tuvo un ambiente capaz de soportar formas pequeñas de vida como los microbios. En otras palabras, su misión es determinar la “habitabilidad” del planeta. El Curiosity hereda tecnología de misiones anteriores como el Mars Exploration Rovers y lleva incorporado un conjunto de cámaras para la navegación y la detección de obstáculos. De las diecisiete cámaras que tiene el rover, seis pares son cámaras de ingeniería; de ellas, cuatro pares se usan para evitar obstáculos y posibles daños (Hazcams), estas cámaras se encuentran ubicadas en la parte inferior frontal y trasera del rover y usan la luz visible para capturar imágenes estereoscópicas en escala de grises. Las escenas tridimensionales reconstruidas en base a los pares estereoscópicos son utilizadas para asegurar que el rover no se pierda o inadvertidamente choque con obstáculos inesperados, y trabaja conjuntamente con un software que permite que el rover tome sus propias decisiones de seguridad, de esta forma “piensa por sí mismo”. Los otros dos pares son cámaras de navegación (Navcams), y están montadas en el mástil (“cuello y cabeza” del rover), estas

cámaras también usan la luz visible para capturar imágenes estereoscópicas en escala de grises, sin embargo, son más precisas que las Hazcams en el cálculo de la profundidad de la escena. Las Navcams trabajan en cooperación con las cámaras usadas para evitar obstáculos, proveyendo una vista complementaria del terreno de exploración. Las diecisiete cámaras constituyen los “ojos” y otros “sentidos” del rover, cinco cámaras ejecutan investigaciones científicas [3].

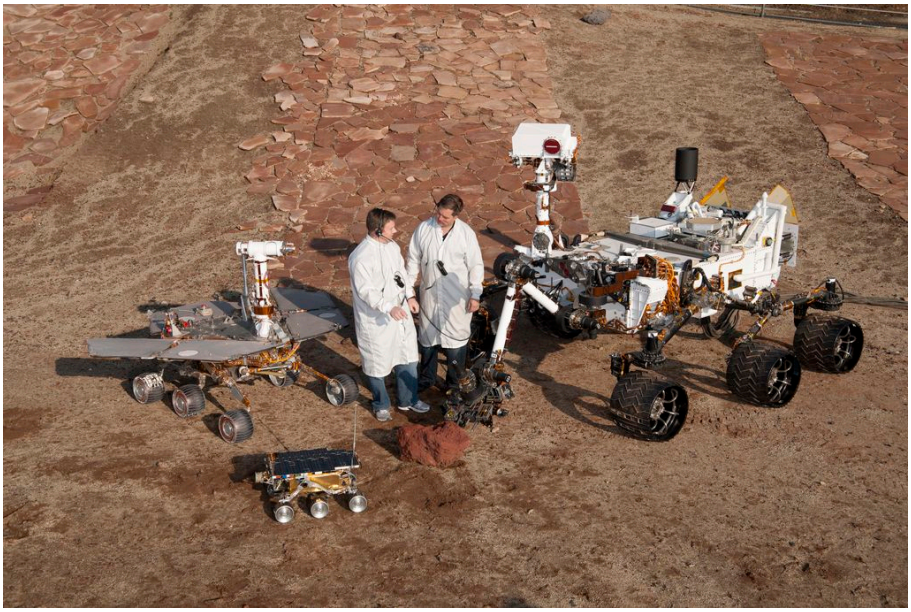


***Figura[3] 1.2**: Parte superior del mástil en el Mars Rover Curiosity.*

La Figura 1.2 muestra siete de las 17 cámaras en el rover Curiosity, los dos pares de cámaras para la navegación (Navcams), son las pequeñas aberturas circulares en ambos lados de la “cabeza”, las otras tres cámaras son para la investigación científica. Las Navcams están fijamente unidos a una placa de montaje para las cámaras con una línea base para el sistema estereoscópico de aproximadamente 42cm. El rover tiene a bordo dos computadoras idénticas llamadas “Rover Compute Element” (RCE), esto es así porque de los dos pares de Navcams en el Curiosity, un par está conectado a la electrónica del RCE “lado A” y

[3] *Image Credit: NASA/JPL-Caltech/LANL*

el otro par se conecta a la electrónica del RCE “lado B”. Solo un RCE se encuentra activo a la vez, el segundo RCE es para redundancia en caso de fallo.



Figura[4] 1.3*: Tres generaciones de rovers con dos ingenieros del JPL.*

La Figura 1.3 provee una comparación de las tres generaciones de rovers desarrollados por el Jet Propulsion Laboratory de la NASA. Al frente y en el centro se encuentra una muestra del primer rover construido para Marte, el Sojourner, que aterrizo en Marte en 1997 como parte del proyecto Mars Pathfinder. A la izquierda hay un rover de prueba del proyecto Mars Exploration Rover que aterrizo en Marte al Spirit y Opportunity en el 2004. En la derecha se muestra un rover de prueba del proyecto Mars Science Laboratory, el Curiosity, que aterrizo con éxito en el 2012.

El proyecto de Google - Search by Image [4], lanzado en junio del 2011 realiza el reconocimiento y clasificación de imágenes que el usuario brinda al buscador, dando como resultado una descripción de la imagen brindada e imágenes visualmente similares entre otros. Google usa técnicas de visión por

[4] *Image Credit: NASA/JPL-Caltech*

computador para relacionar la imagen con otras imágenes que se encuentran indexadas, de esta relación se intenta generar la más precisa “mejor conjetura” descripción textual de la imagen, además se busca otras imágenes con el mismo contenido de la imagen inicial. Esta es una tecnología que se encuentra en constante desarrollo y por la cual empresas como Google apuestan constantemente. La Figura 1.4 muestra la interfaz diseñada por Google para la búsqueda por imágenes.

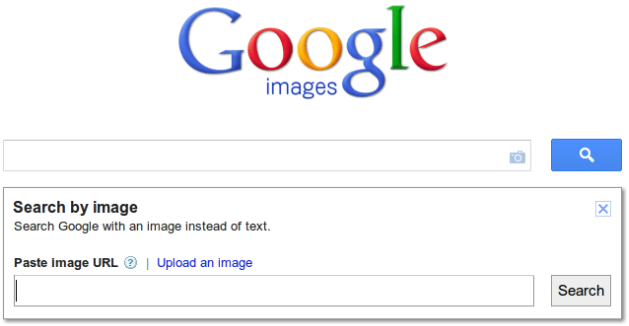


***Figura 1.4**: Cuadro de texto para iniciar una búsqueda por imagen.*

Kinect, el producto desarrollado por Microsoft y lanzado al mercado en noviembre del 2010 usa tecnología de visión por computador para realizar el control de los juegos mediante el cuerpo humano en la videoconsola. El sistema está equipado con un conjunto de sensores entre los que se encuentran un sensor de profundidad, el cual está constituido de un proyector infrarrojo combinado con un sensor monocromático CMOS, que permite al sistema ver en tres dimensiones lo que se encuentra frente a él, aun en condiciones de baja iluminación (véase Figura 1.5). Otro sensor que posee este sistema es una cámara RGB la cual entrega el color básico de tres componentes, además la cámara ayuda en el reconocimiento facial y otras funciones.

***Figura 1.5**: Disposición de cámaras en el Kinect.*

El algoritmo de visión que tiene Kinect toma un enfoque de reconocimiento de objetos que se encuentra entrenado para distinguir 32 partes del cuerpo humano [5], el amplio y altamente variado conjunto de datos de entrenamiento permite al clasificador estimar las partes del cuerpo invariantes a la posición, forma, ropa usada, etc. Finalmente el sistema predice las posiciones de cada una de las articulaciones del cuerpo en 3D, el proceso es repetido 30 veces por segundo.

Los robots humanoides Robonaut y ASIMO mostrados en la Figura 1.6 han sido desarrollados por la NASA/DARPA (DARPA por su siglas en inglés, Defense Advanced Research Projects Agency) y Honda respectivamente, tienen implementados sistemas de reconocimiento de objetos mediante visión estereoscópica que les permiten desarrollar tareas practicas en un entorno 3D que incluye múltiples objetos de diferentes tipos y con geometrías complejas, superficies oscuras, brillantes o traslucidas entre otros. Robonaut es el primer robot humanoide construido para realizar tareas en el espacio ofreciendo asistencia a los astronautas en la Estación Espacial Internacional, su algoritmo de visión le permite realizar el reconocimiento de objetos y la estimación de la posición, [6]. El concepto principal detrás de ASIMO fue el de crear un sistema móvil más viable que permita a los robots ayudar a las personas en sus tareas cotidianas y llegando a ser sus compañeros, el robot tendría que ser capaz de maniobrar entre objetos en una

habitación y ser capaz de subir y bajar escaleras, [7]. Usando la información visual capturada por las cámaras montadas en la cabeza, ASIMO puede detectar el movimiento de múltiples objetos evaluando la distancia y dirección, es capaz de interpretar las posturas y los gestos de los seres humanos, [8].

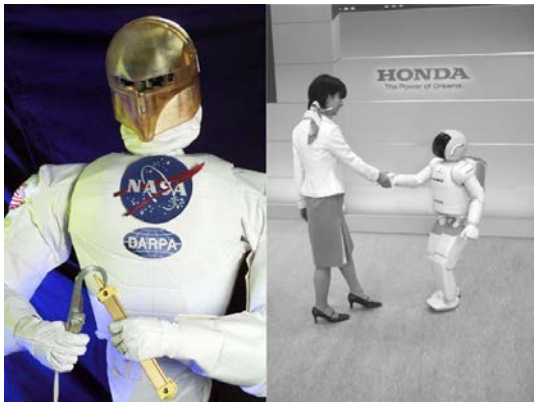


Figura 1.6*: Los robots humanoides Robonaut y ASIMO en la izquierda y derecha respectivamente.*

1.2. Justificación

El reconocimiento y clasificación de objetos que realizan las empresas que tienen ésta etapa en su línea de producción, o los robots móviles construidos para la exploración que normalmente realizan la tarea de reconocimiento de objetos, llevan a cabo el proceso de reconocimiento mediante sistemas tradicionales que son normalmente construidos para realizar una tarea específica, teniendo que modificar parte del sistema o el sistema por completo para realizan una nueva tarea de reconocimiento. Varios de estos sistemas usan un conjunto de sensores o la captura de una imagen individual sobre la escena para su operación utilizando técnicas clásicas de inteligencia artificial como las redes neuronales, más aun, algunas líneas de producción en las empresas realizan esta tarea manualmente, es por ello que hay la necesidad de desarrollar nuevos sistemas de reconocimiento y clasificación de objetos que utilicen nuevas técnicas de inteligencia artificial como

support vector machine, que sean capaces de resolver problemas genéricos para las tareas de reconocimiento con una mejor performance que las técnicas clásicas, y que además sean flexibles para que puedan adaptarse al problema concreto que se desea abordar como es requerido normalmente en los robots móviles para la exploración.

El uso de esta tecnología hace más eficientes y eficaces a las empresas que realizan una etapa de clasificación de objetos en todo el proceso productivo, es por ello que el desarrollo de sistemas autónomos para las tareas de reconocimiento y clasificación es importante para poder realizar una clasificación más objetiva, incrementar la productividad de la empresa y por lo tanto incrementar los ingresos de la misma, además de mejorar la calidad de vida de sus trabajadores.

1.3. Planteamiento del problema

Se tiene dos cámaras separadas mediante una distancia constante y colocadas en el mismo nivel horizontal, las cuales simulan los ojos de nuestro sentido de la vista, para la adquisición de imágenes de una escena en el espacio tridimensional; los algoritmos implementados para esta tesis son ejecutados por el CPU de un computador que interactúa con las dos cámaras.

Esta tesis busca demostrar que los algoritmos de inteligencia artificial desarrollados y propuestos, algoritmos para el reconocimiento de objetos basados en SVM, serán capaces de clasificar objetos de interés que ya le han sido enseñados al sistema de reconocimiento de un espacio tridimensional real.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar un sistema de reconocimiento capaz de aprender a clasificar objetos del espacio tridimensional real, mediante su entrenamiento a través de ejemplos previos, y que pueda adaptarse a nuevas situaciones utilizando visión estereoscópica e inteligencia artificial basada en SVM, ejecutada en un computador.

1.4.2. Objetivos específicos

En la tesis, se establecieron los siguientes objetivos específicos:

- Desarrollar algoritmos para el procesamiento individual de las dos imágenes obtenidas desde las cámaras, aplicando filtros u otras técnicas para el posterior cálculo de la disparidad.
- Desarrollar algoritmos capaces de obtener la disparidad de las imágenes para obtener información acerca de la profundidad.
- Segmentar la escena para la obtención de posibles objetos.
- Extraer diferentes características de los posibles objetos detectados en la escena para su posterior reconocimiento.
- Determinar la arquitectura más adecuada para los SVM como reconocedor de objetos, determinando las funciones kernel a utilizar.
- Realizar la programación de algoritmos de alto nivel tomando como principio la teoría de SVM para el adecuado desempeño del sistema como reconocedor y clasificador de objetos.

- Evaluar el desempeño del sistema de reconocimiento mediante la presentación de objetos en diferentes condiciones y la exposición ante nuevas situaciones.

1.5. Organización de la tesis

La tesis está organizada en seis capítulos, en donde el primer capítulo dio a conocer los antecedentes, la justificación, el planteamiento del problema y los objetivos. El segundo capítulo describe los SVM, que son la base para la implementación del sistema “inteligente” desarrollado en esta tesis. El tercer capítulo presenta los fundamentos para la visión estereoscópica que es usada en las primeras etapas del sistema. En el cuarto capítulo se encuentra el desarrollo de los algoritmos para el reconocimiento y clasificación de objetos. El quinto capítulo presenta las simulaciones y resultados obtenidos en la identificación de objetos presentados al sistema de reconocimiento desarrollado. En el último capítulo se encuentra las discusiones finales, las conclusiones y algunas recomendaciones.

El desarrollo de la tesis se ha basado en la **experimentación para obtención de nuevos productos**, que es un método corrientemente empleado en los casos en donde la investigación tiene por objeto provocar determinados fenómenos que no se presentan usualmente en la naturaleza y cuyo conocimiento puede ser interesante o importante en el avance de la ciencia o la tecnología. La implementación realizada en este trabajo se enfoca en el sistema de visión para el reconocimiento de objetos, el cual podría ser montado sobre un robot para su navegación o formar parte de una línea de producción que requiere la clasificación y reconocimiento de un objeto en particular.

Capítulo 2

Support vector machines como reconocedor de objetos

2.1. La capacidad de las máquinas de aprender

La construcción de máquinas capaces de aprender de la experiencia ha recibido un gran impulso con la llegada de la electrónica de computadoras que pueden realizar cada vez mayor cantidad de operaciones por segundo. Ello ha demostrado que las máquinas pueden mostrar un nivel significativo de la capacidad de aprendizaje, aunque los límites de esta facultad están lejos de ser claramente definidos.

La disponibilidad de sistemas fiables de aprendizaje es de importancia estratégica ya que hay diferentes tareas que no pueden ser resueltas con técnicas de programación clásica, pues no se disponen de modelos matemáticos para diversos problemas, o estos no se pueden modelar con precisión. Así, por ejemplo, no se sabe cómo escribir un programa de computadora para realizar el reconocimiento de caracteres escritos a mano, aunque hay un montón de ejemplos

disponibles. Por lo tanto, es natural preguntarse si una máquina puede ser entrenada para reconocer los caracteres como la letra 'A' a partir de ejemplos – después de todo está es la forma como los humanos aprenden a leer. Nos referiremos a este enfoque de resolución de problemas como la metodología de aprendizaje, [9].

Los support vector machine (SVM) son sistemas de aprendizaje que implementan el principio de minimización del riesgo estructural, usan un espacio de hipótesis de funciones lineales en un espacio de características de alta dimensión, y son entrenados con un algoritmo de aprendizaje de la teoría de optimización que implementa una tendencia de aprendizaje derivada de la teoría de aprendizaje estadístico. Esta estrategia de aprendizaje introducida por el Ph.D. Vladimir Vapnik [10] y sus compañeros de trabajo son un principio y un método muy poderoso que en pocos años desde su introducción ya ha superado a la mayoría de otros sistemas y tiene una amplia variedad de aplicaciones.

2.2. Teoría de aprendizaje estadístico

Los SVM son máquinas de aprendizaje lineales que resuelven de forma nativa problemas de clasificación binaria. En esta sección se dará las ideas de la teoría de aprendizaje estadístico, que constituye la base para la construcción de SVM.

Los modelos de aprendizaje supervisado, como los SVM, consisten de tres componentes interrelacionados y abstraídos en términos matemáticos que son el entorno, el profesor y la máquina de aprendizaje (Algoritmo). El problema del aprendizaje supervisado consiste en aprender las salidas deseadas dadas por el

profesor para cada patrón de entrada. Consideremos un problema general de clasificación binaria de N patrones con las siguientes características.

1. El conjunto de datos $\{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \dots, (\boldsymbol{x}_N, t_N)\}$, es compuesto por las muestras de entrenamiento $\boldsymbol{x}_i$ de longitud M, con elementos $\boldsymbol{x}_i = (x_1, x_2, \dots, x_M)$, y sus salidas deseadas $t_i \in \{-1, +1\}$.

2. El objetivo es encontrar un clasificador con una función de decisión, $f(\boldsymbol{x})$, tal que $f(\boldsymbol{x}_i) = t_i, \forall(\boldsymbol{x}_i, t_i)$

Sea $L(f(\boldsymbol{x}), t)$ una medida de la pérdida o discrepancia entre la respuesta deseada t correspondiente al vector de entrada $\boldsymbol{x}$ y la respuesta de una máquina de aprendizaje. Existen diferentes definiciones para la función de pérdida como por ejemplo la función de pérdida cuadrática, definida como la distancia al cuadrado entre la salida deseada y la salida dada por la máquina de aprendizaje. Para el problema de clasificación binaria se define a la función de pérdida de la siguiente forma:

$$L(f(\boldsymbol{x}), t) = \begin{cases} 0 & si\ f(\boldsymbol{x}) = t, \\ 1 & en\ otro\ caso. \end{cases} \tag{2.1}$$

Ahora, considerar una máquina de aprendizaje con un conjunto de parámetros ajustables $\boldsymbol{w}$ (por ejemplo, en una red neuronal los pesos y los umbrales). Dada la tarea de clasificación binaria mostrada arriba, la máquina busca encontrar w tal que aprenda el mapeo $\boldsymbol{x} \rightarrow t$. Esto resultará en un posible mapeo $\boldsymbol{x} \rightarrow f(\boldsymbol{x}, \boldsymbol{w})$ que define la máquina. Una elección particular de $\boldsymbol{w}$ genera lo que llamamos una máquina entrenada.

Supongamos que existe una distribución de probabilidad $P(x, t)$ del que se han tomado las muestras del problema de clasificación binaria. Esto es mas general

que asignar un t fijo a cada x. Con esta distribución, las etiquetas t_n se asignan según una distribución de probabilidad condicional en x_i. Sin embargo en las siguientes secciones se asume un t fijo para un x dado.

La esperanza del error de evaluación para una máquina entrenada, llamada también riesgo real, riesgo esperado o simplemente función de riesgo, es:

$$R(\boldsymbol{w}) = \int L(f(\boldsymbol{x}, \boldsymbol{w}), t_i) dP(x, t). \tag{2.2}$$

El objetivo del aprendizaje supervisado es minimizar la función de riesgo $R(\boldsymbol{w})$. Sin embargo, la evaluación de la función de riesgo es complicada porque la distribución de probabilidad $P(x, t)$ es usualmente desconocida. En el aprendizaje supervisado la información disponible solo está contenida en los datos de entrenamiento. Para superar esta dificultad matemática se usa el principio inductivo de minimización del riesgo empírico. Este principio se basa enteramente en los datos de entrenamiento, lo cual lo hace perfectamente apropiado en la filosofía de diseño de las redes neuronales.

2.2.1. Minimización del riesgo empírico

La función de riesgo empírico es definido en términos de la función de pedida $L(f(\boldsymbol{x}), t)$ como:

$$R_{emp}(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} L(f(\boldsymbol{x_i}, \boldsymbol{w}), t_i), \tag{2.3}$$

donde N es el tamaño del conjunto de entrenamiento y $\boldsymbol{w}$ es el conjunto de parámetros ajustables.

La idea básica del principio de minimización del riesgo empírico es trabajar con la función de riesgo empírico $R_{emp}(\boldsymbol{w})$. Esta nueva función difiere de la función de riesgo $R(\boldsymbol{w})$ en dos aspectos deseables:

- No depende de una distribución de probabilidad desconocida $P(x,t)$ de forma explícita.
- En teoría esta función puede ser minimizada con respecto al vector de pesos $\boldsymbol{w}$.

Se puede probar que si la cantidad de muestras N del conjunto de entrenamiento crece, espacio de entrada denso, el punto mínimo de la función de riesgo empírico $R_{emp}(\boldsymbol{w})$ converge en probabilidad al punto mínimo de la funcional de riesgo $R(\boldsymbol{w})$ [11]. Las máquinas de aprendizaje que implementan este principio en general son buenos clasificadores, sin embargo, si la complejidad de la máquina es alta, estas máquinas tienden a sobre ajustar los datos.

2.2.2. Minimización del riesgo estructural

En contraste con la minimización del riesgo empírico, el principio de minimización de riesgo estructural considera la complejidad de la máquina de aprendizaje cuando es estrenada en busca de los parámetros ajustables $\boldsymbol{w}$. Esto es hecho minimizando la función de riesgo $R(\boldsymbol{w})$. Como el cálculo de la función de riesgo es complicada, en lugar de ello se busca una cota superior para esta función. Sea ρ la probabilidad de que ocurra la siguiente desigualdad con $\epsilon > 0$:

$$\sup_{\boldsymbol{w}} \left| R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w}) \right| \geq \epsilon. \tag{2.4}$$

Luego, con probabilidad $1-\rho$, podemos afirmar que para todo conjunto de vectores $\boldsymbol{w}$ la siguiente desigualdad se cumple:

$$R(\boldsymbol{w}) < R_{emp}(\boldsymbol{w}) + \epsilon, \tag{2.5}$$

la definición de la probabilidad ρ se muestra en la siguiente ecuación:

$$\rho = \left(\frac{2\mathrm{e}N}{h}\right)^{h} \mathrm{e}^{-\epsilon^2 \mathrm{N}}, \tag{2.6}$$

donde e es la base de los logaritmos neperianos y h es un número entero no negativo llamado dimensión de Vapnik - Chervonenkis (dimensión VC). Reescribiendo la ecuación (2.5):

$$R(\boldsymbol{w}) < R_{emp}(\boldsymbol{w}) + \sqrt{\frac{h}{N}\left[ln\left(\frac{2N}{h}\right)+1\right] - \frac{1}{N} ln\,(\rho)}, \tag{2.7}$$

a la parte derecha de la desigualdad se le llama cota de riesgo y al segundo término de la cota de riesgo se le denomina intervalo de confianza. Notar que el intervalo de confianza depende del tamaño N del conjunto de entrenamiento, la dimensión VC, y la probabilidad ρ. En el contexto de clasificación de patrones binarios, como se viene realizando, la función de riesgo empírico $R_{emp}(\boldsymbol{w})$ es el error de entrenamiento de la máquina de aprendizaje (frecuencia de errores durante la etapa de entrenamiento). Para un número fijo de muestras de entrenamiento N, el error de entrenamiento decrece monótonamente, mientras que el intervalo de confianza crece

monótonamente conforme la dimensión VC se incrementa. Estas tendencias son mostradas en forma genérica en la Figura 2.1.

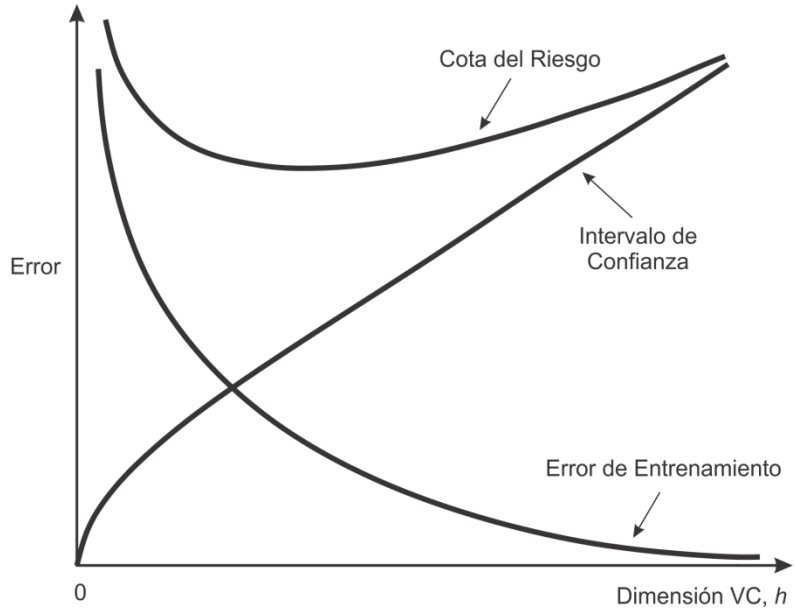


***Figura 2.1**: Relación del error de entrenamiento, intervalo de confianza y la cota del riesgo.*

El reto de resolver problemas de aprendizaje supervisado es obtener la mejor performance de generalización mediante la correcta relación entre la capacidad de la máquina de aprendizaje y la cantidad de datos de entrenamiento disponibles para el problema. El principio de minimización de riesgo estructural provee un procedimiento inductivo para lograr esto mediante el uso de la dimensión VC de la máquina de aprendizaje como una variable de control. El principio de minimización de riesgo estructural afirma que la mejor estructura para un conjunto de máquinas de aprendizaje es aquella para la cual la cota del riesgo es mínima. La implementación de la minimización del riesgo estructural provee una máquina de aprendizaje con un compromiso óptimo entre el error de entrenamiento (calidad de

aproximación de los datos de entrenamiento) y el intervalo de confianza (complejidad de la máquina de aprendizaje) que compiten entre sí [12].

2.2.3. Dimensión VC

La dimensión VC, llamada así en honor a sus autores Vapnik – Chervonenkis, es una medida de la capacidad o poder expresivo de una familia de máquinas de aprendizaje $\{f(\boldsymbol{w})\}$.

Si un conjunto de N patrones puede etiquetarse de 2^N formas distintas, y para cada etiquetado existe una máquina de aprendizaje de la familia $\{f(\boldsymbol{w})\}$ que asigne correctamente estas etiquetas, se dice que el conjunto de patrones es quebrado por esa familia de máquinas de aprendizaje. La dimensión VC de una familia de máquinas de aprendizaje $\{f(\boldsymbol{w})\}$ se define como el máximo número de patrones de entrenamiento que pueden ser quebrados por $\{f(\boldsymbol{w})\}$. La dimensión VC está íntimamente relacionada con la complejidad de la máquina de aprendizaje.

2.3. Support vector machine lineal

Ésta y la siguiente sección describen la formulación matemática para la construcción de SVM, la cual fue desarrollada por el Ph.D. Vladimir Vapnik [10]. En primer lugar se considera la construcción del hiperplano óptimo para el simple caso de patrones linealmente separables. Luego se describen clasificadores que tienen hiperplanos con margen suave para el caso de patrones no linealmente separables. Finalmente se presenta los clasificadores no lineales que son los más interesantes y útiles para la solución de un problema no separable.

2.3.1. Hiperplano de margen máximo

Considerar la muestra de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^{N}$ donde $\boldsymbol{x_i}$ es un patrón de entrada para el $i-esimo$ ejemplo y t_i es el valor de la respuesta deseada para el correspondiente ejemplo. Para empezar, se asume que las clases representadas por los subconjuntos $t_i = +1$ y los patrones del subconjunto $t_i = -1$ son "linealmente separables". La ecuación de la superficie de decisión en forma de hiperplano que hace la separación es:

$$\boldsymbol{w^T x} + b = 0, \tag{2.8}$$

donde $\boldsymbol{x}$ es el vector de entrada, $\boldsymbol{w}$ es el vector de parámetros ajustables, y b es el bias. Así se puede escribir:

$$\boldsymbol{w^T x_i} + b \geq 0 \quad para\ t_i = +1, \tag{2.9}$$

$$\boldsymbol{w^T x_i} + b < 0 \quad para\ t_i = -1. \tag{2.10}$$

Se ha asumido patrones linealmente separables solo para explicar la idea básica detrás de SVM; más adelante se muestra que los patrones a clasificar pueden ser linealmente no separables, como ocurre con la mayoría de los casos en los problemas reales.

Para un vector de pesos $\boldsymbol{w}$ y bias b, la separación entre el hiperplano definido por la ecuación (2.8) y la muestra de entrenamiento más cercana es llamado el margen de separación, denotado por ρ. La meta de un SVM es encontrar un hiperplano particular para el cual el margen de separación ρ es máximo, un ejemplo de esto se muestra en la Figura 2.2. Bajo estas condiciones, la superficie de decisión es referida como el hiperplano óptimo.

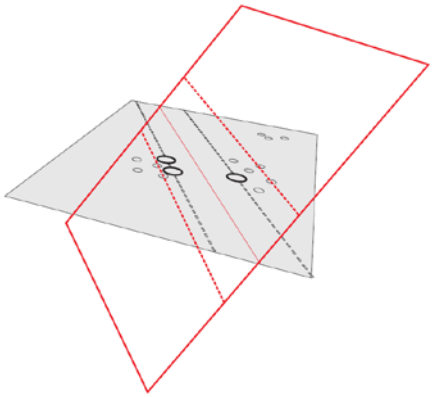


***Figura 2.2**: Hiperplano óptimo (en rojo) para la separación de las clases.*

Sean $\boldsymbol{w_0}$ y b_0 los valores óptimos del vector de pesos y el bias respectivamente, entonces, el hiperplano óptimo que representa a una superficie de decisión multidimensional en el espacio de entrada, es definido por:

$$\boldsymbol{w_0^T x} + b_0 = 0, \tag{2.11}$$

la función de discriminación:

$$g(\boldsymbol{x}) = \boldsymbol{w_0^T x} + b_0, \tag{2.12}$$

provee una medida algebraica de la distancia de $\boldsymbol{x}$ al hiperplano óptimo. Esto es más fácil de ver expresando $\boldsymbol{x}$ como:

$$\boldsymbol{x} = \boldsymbol{x_p} + r\frac{\boldsymbol{w_0}}{\|\boldsymbol{w_0}\|}, \tag{2.13}$$

donde $\boldsymbol{x_p}$ es la proyección normal de $\boldsymbol{x}$ sobre el hiperplano óptimo, y r es la distancia algebraica deseada; r es positivo si $\boldsymbol{x}$ se encuentra sobre el lado positivo del hiperplano óptimo y negativo si $\boldsymbol{x}$ esta en el lado negativo. Desde que, por definición, $g(\boldsymbol{x_p}) = 0$, se demuestra que:

$$g(\boldsymbol{x}) = \boldsymbol{w_0^T x} + b_0 = r\|\boldsymbol{w_0}\|, \tag{2.14}$$

lo que es equivalente a:

$$r = \frac{g(\boldsymbol{x})}{\|\boldsymbol{w_0}\|}. \tag{2.15}$$

Un caso particular es la distancia desde el origen ($\boldsymbol{x} = 0$) al hiperplano óptimo que está dado por $b_0/\|\boldsymbol{w_0}\|$. Si $b_0 > 0$, el origen se encuentra en el lado positivo del hiperplano óptimo; si $b_0 < 0$, se encuentra en el lado negativo. Si $b_0 = 0$, el hiperplano óptimo pasa por el origen. Una interpretación geométrica de estos resultados algebraicos es mostrada en la Figura 2.3.

El objetivo hasta ahora es encontrar los parámetros $\boldsymbol{w_0}$ y b_0 para el hiperplano óptimo, dado el conjunto de entrenamiento $T = \{(\boldsymbol{x}_i, t_i)\}$. De los resultados y ecuaciones anteriores, se puede ver que el par $(\boldsymbol{w_0}, b_0)$ satisface lo siguiente:

$$\boldsymbol{w_0^T x_i} + b_0 \geq 1 \quad para\ t_i = +1, \tag{2.16}$$

$$\boldsymbol{w_0^T x_i} + b_0 \leq -1 \quad para\ t_i = -1. \tag{2.17}$$

Notar que si las ecuaciones (2.9) y (2.10) se mantienen, esto es, los patrones son linealmente separables, siempre se puede reescalar $\boldsymbol{w_0}$ y b_0 tal que las ecuaciones (2.16) y (2.17) se cumplan; este escalamiento deja a la ecuación (2.11) sin ningún cambio.

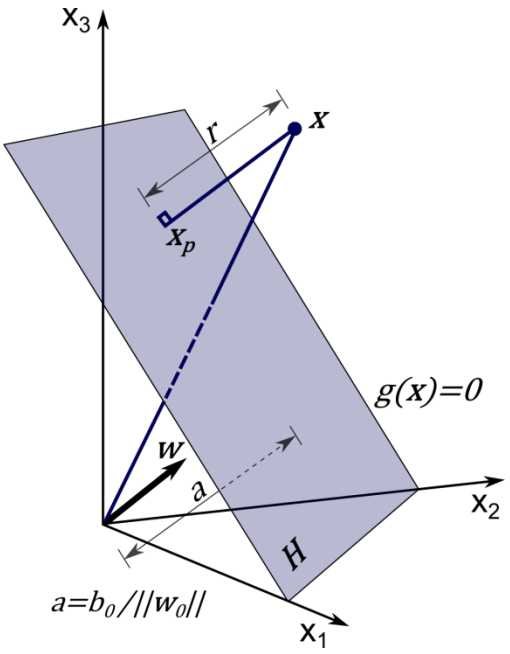


***Figura 2.3**: Interpretación geométrica de la distancia algebraica de los puntos al hiperplano óptimo para un caso en tres dimensiones.*

Los puntos particulares $(\boldsymbol{x}_i, t_i)$ para los que cualquiera de las ecuaciones (2.16) ó (2.17) satisface en la igualdad, son llamados *support vectors (vectores de soporte)*, he aquí la razón del nombre "support vector machine". En términos conceptuales, los vectores de soporte son aquellos puntos que se encuentran más cercanos a la superficie de decisión y por lo tanto son los que presentan mayor dificultad para ser clasificados. Como tales, tienen una influencia directa en la ubicación óptima de la superficie de decisión.

Considerar el vector de soporte $\boldsymbol{x}^{(s)}$ para el cual $t^{(s)} = +1$. Entonces, por definición tenemos:

$$g\left(\boldsymbol{x}^{(s)}\right) = \boldsymbol{w_0^T x}^{(s)} \mp b_0 = \mp 1 \qquad para\ t^{(s)} = \mp 1. \tag{2.18}$$

De la ecuación (2.15), la distancia algebraica desde el vector de soporte $\boldsymbol{x}^{(s)}$ al hiperplano óptimo es:

$$r = \frac{g(\boldsymbol{x}^{(s)})}{\|\boldsymbol{w_0}\|} = \begin{cases} \frac{1}{\|\boldsymbol{w_0}\|} & si\ t^{(s)} = +1, \\ \frac{-1}{\|\boldsymbol{w_0}\|} & si\ t^{(s)} = -1, \end{cases} \tag{2.19}$$

donde el signo positivo indica que $\boldsymbol{x}^{(s)}$ está en el lado positivo del hiperplano óptimo y el signo negativo indica que esta en el lado negativo. Sea ρ el valor óptimo del margen de separación entre las dos clases que constituyen el conjunto de entrenamiento T. Entonces de la ecuación (2.19) se desprende que:

$$\rho = 2r = \frac{2}{\|\boldsymbol{w_0}\|}. \tag{2.20}$$

Esta última ecuación establece que maximizar el margen de separación entre las clases es equivalente a minimizar la norma Euclidiana del vector de pesos $\boldsymbol{w}$.

En resumen, el hiperplano óptimo definido en la ecuación (2.11) es único en el sentido de que el óptimo vector de pesos $\boldsymbol{w_0}$ provee la máxima separación posible entre las muestras positivas y negativas. Esta condición óptima es alcanzada mediante la minimización de la norma Euclidiana del vector de pesos $\boldsymbol{w}$ [11].

Entrenamiento de la máquina de aprendizaje

El entrenamiento de la máquina de aprendizaje consiste en encontrar los valores del vector de pesos $\boldsymbol{w_0}$ y b_0 tal que la máquina sea capaz de clasificar correctamente los patrones que se le presenten en el futuro, para ello se provee a la máquina un conjunto de entrenamiento.

El entrenamiento de los SVM se realiza mediante la solución de un problema de optimización con restricciones que puede ser escrito de la siguiente forma:

Dado el conjunto de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^{N}$*, encontrar los valores óptimos del vector de pesos* $\boldsymbol{w}$ *y el bias* b *tal que satisfacen las restricciones:*

$$t_i(\boldsymbol{w^T x_i} + b) \geq 1, \quad i = 1,2,\dots,N \tag{2.21}$$

y el vector de pesos $\boldsymbol{w}$ *minimiza la siguiente función de costo:*

$$\varphi(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w^T w}. \tag{2.22}$$

El factor de escala $1/2$ en la función de costo es incluido por conveniencias de presentación. Este problema de optimización con restricciones es llamado el problema primario, el cual será resuelto mediante el método de los multiplicadores de Lagrange. Para ello construimos la función de Lagrange:

$$L_P(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\boldsymbol{w^T w} - \sum_{i=1}^{N} \alpha_i [t_i(\boldsymbol{w^T x_i} + b) - 1], \tag{2.23}$$

las variables auxiliares no negativas $\alpha_1, \alpha_2, \dots \alpha_N$ son los multiplicadores de Lagrange. Para solucionar el problema de optimización se debe minimizar la función de Lagrange, ecuación (2.23), con respecto a $\boldsymbol{w}$ y b; además de ser maximizado con respecto a α. Calculando las derivadas parciales de la función de Lagrange se obtienen las siguientes condiciones:

$$\frac{\partial L(\boldsymbol{w}, b, \alpha)}{\partial \boldsymbol{w}} = 0 \quad \rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \, t_i \boldsymbol{x_i}, \tag{2.24}$$

$$\frac{\partial L(\boldsymbol{w}, b, \alpha)}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{N} \alpha_i \, t_i = 0. \tag{2.25}$$

El vector de pesos $\boldsymbol{w}$ es definido en términos de una expansión que involucra los N ejemplos de entrenamiento, sin embargo, esta solución es única en virtud de la convexidad de la función de Lagrange para el problema que estamos analizando.

Dado que en el problema primario la función de costo es una función convexa y las restricciones son lineales, es posible construir otro problema llamado el problema dual. Este segundo problema tiene los mismos valores óptimos que el problema primario, pero con los multiplicadores de Lagrange proveyendo la solución óptima. Para transformar nuestro problema primario en el problema dual primero expandimos la ecuación (2.23):

$$L_P(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\boldsymbol{w^T w} - \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{w^T x_i} - b \sum_{i=1}^{N} \alpha_i t_i + \sum_{i=1}^{N} \alpha_i. \tag{2.26}$$

De la ecuación (2.25), el tercer miembro del lado derecho de la ecuación anterior es cero, además de la ecuación (2.24) se obtiene:

$$\boldsymbol{w^T w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{w^T x_i} = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x_i^T x_j}, \tag{2.27}$$

reformulando la ecuación (2.23) obtenemos la siguiente expresión:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x_i^T x_j}, \tag{2.28}$$

donde los α_i son valores no negativos. Ahora se puede plantear el problema dual de la siguiente forma:

Dado el conjunto de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^{N}$ *, encontrar los multiplicadores de Lagrange* $\{\alpha_i\}_{i=1}^{N}$ *que maximicé la función objetivo:*

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x_i^T x_j}, \tag{2.29}$$

sujeto a las restricciones:

$$\sum_{i=1}^{N} \alpha_i \, t_i = 0, \tag{2.30}$$

$$\alpha_i \geq 0 \quad para\ i = 1,2, \dots, N. \tag{2.31}$$

Notar que hay un multiplicador de Lagrange para cada patrón de entrenamiento. Aquellos puntos para los cuales $\alpha_i > 0$ son los vectores soporte, el resto de patrones tienen $\alpha_i = 0$. El hiperplano óptimo es definido por los vectores soporte, por lo que si todos los demás puntos fueran eliminados y se vuelve a realizar el entrenamiento, se obtendría el mismo hiperplano óptimo para la separación de las clases.

Luego de calcular los multiplicadores de Lagrange en el problema dual, denotados por $\alpha_{o,i}$ podemos calcular el vector de pesos óptimo $\boldsymbol{w_o}$ usando la ecuación (2.24):

$$\boldsymbol{w_o} = \sum_{i=1}^{N} \alpha_{o,i}\, t_i \boldsymbol{x_i}. \tag{2.32}$$

Para el cálculo del bias óptimo b_o, usamos el vector de pesos $\boldsymbol{w_o}$ que se acaba de calcular y con la ayuda de la ecuación (2.18), se puede escribir:

$$b_o = 1 - \boldsymbol{w_o^T x^{(s)}} \quad para\ t^{(s)} = 1. \tag{2.33}$$

2.3.2. Hiperplano con margen suave

El hiperplano de margen máximo es un concepto importante como punto de inicio para el análisis y construcción de support vector machines de mayor complejidad, pero no puede ser usado en problemas comunes del mundo real: si los datos presentan ruido, estos serán en general linealmente no separables en el espacio de entrada. Por lo tanto, dado un conjunto de datos de entrenamiento no linealmente separables (como la mayoría de casos en el mundo real), no es posible construir un hiperplano de separación con margen máximo sin errores de clasificación. No obstante, el objetivo es encontrar un hiperplano óptimo que minimiza la probabilidad del error de clasificación.

El margen de separación entre las clases se dice que es suave si los datos $(\boldsymbol{x}_i, t_i)$ no cumplen la siguiente condición:

$$t_i(\boldsymbol{w^T x_i} + b) \geq +1, \quad i = 1,2, \dots, N. \tag{2.34}$$

Para permitir un margen suave entre las clases se introduce un nuevo conjunto de variables escalares no negativas, $\{\xi_i\}$, en la definición del hiperplano de separación:

$$t_i(\boldsymbol{w^T x_i} + b) \geq 1 - \xi_i, \quad i = 1,2, \dots, N. \tag{2.35}$$

Las ξ_i son llamadas variables de holgura, y miden la desviación de los datos de una situación ideal de separación de patrones. En la Figura 2.4 se muestra las variables de holgura y un hiperplano de separación con margen suave.

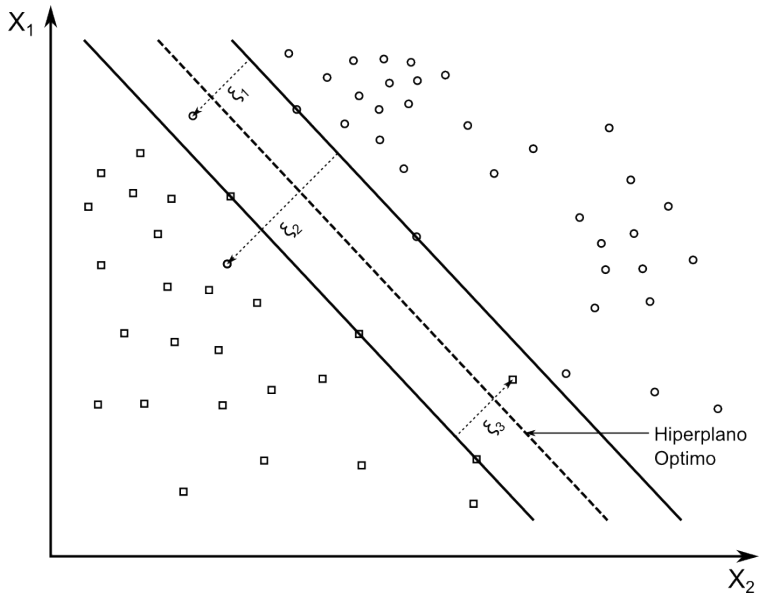


***Figura 2.4**: Hiperplano con margen suave.*

Para que un patrón quede mal clasificado, el correspondiente ξ_i debe ser mayor a la unidad. La suma $\sum_{i=1}^{N} \xi_i$ es, por lo tanto, una cota superior en el número de errores sobre el entrenamiento y es usado para añadir el costo de los errores a la función de costo. Por lo tanto, podemos plantear el problema primario para un clasificador de margen suave como sigue:

Dado el conjunto de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^{N}$, *encontrar los valores óptimos del vector de pesos* $\boldsymbol{w}$ *y el bias* b *tal que satisfacen las restricciones:*

$$t_i(\boldsymbol{w^T x_i} + b) \geq 1 - \xi_i, \quad i = 1,2, \dots, N$$
$$\xi_i \geq 0 \quad \forall\, i, \tag{2.36}$$

y el vector de pesos $\boldsymbol{w}$ *y las variables de holgura* ξ_i *minimizan la siguiente función de costo:*

$$\varphi(\boldsymbol{w}, \xi) = \frac{1}{2}\boldsymbol{w^T w} + C\sum_{i=1}^{N}\xi_i, \tag{2.37}$$

donde C *es un parámetro positivo ajustable.*

El parámetro C controla el equilibrio entre la complejidad de la máquina de aprendizaje y el número de patrones clasificados incorrectamente. Para valores de C cercanos a cero la solución converge a un clasificador con hiperplano de margen máximo. Por otro lado, si el parámetro C tiende a infinito se hace énfasis en minimizar los errores de clasificación y la máquina de aprendizaje puede perder capacidad de generalización. El parámetro C es normalmente determinado de forma experimental por el usuario.

Usando el método de los multiplicadores de Lagrange de forma similar como se hizo para el clasificador con hiperplano de margen máximo, se puede plantear el problema dual para un clasificador de margen suave como sigue:

Dado el conjunto de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^{N}$ *, encontrar los multiplicadores de Lagrange* $\{\alpha_i\}_{i=1}^{N}$ *que maximize la función objetivo:*

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x_i^T x_j}, \tag{2.38}$$

sujeto a las restricciones:

$$\sum_{i=1}^{N} \alpha_i \, t_i = 0, \tag{2.39}$$

$$0 \leq \alpha_i \leq C, \quad para\ i = 1,2,\dots,N \tag{2.40}$$

donde C *es un parámetro positivo ajustable.*

Notar que ni las variables de holgura ξ_i ni sus multiplicadores de Lagrange aparecen en el problema dual, de esta forma, el problema dual para el caso de un hiperplano con margen suave es muy similar al caso más simple de un hiperplano de margen máximo, con la diferencia de que la restricción $\alpha_i \geq 0$ en el hiperplano de margen máximo es reemplazado con la restricción $0 \leq \alpha_i \leq C$ en un clasificador con hiperplano de margen suave. Excepto por esta modificación, el problema de optimización con restricciones para un hiperplano de margen suave y el cálculo de los valores óptimos del vector de pesos $\boldsymbol{w}$ y el bias b se realiza de la misma forma que en el caso de un hiperplano de margen máximo.

2.4. Support vector machine no lineal

En las secciones anteriores se utilizaron hiperplanos para la clasificación de los patrones, diversos de los problemas de clasificación no pueden ser resueltos

eficazmente con los métodos presentados anteriormente, estos problemas necesitan superficies de decisión no lineales para la correcta clasificación de los patrones. Por lo que, en esta sección se desarrolla SVM no lineal que resuelven este tipo de problemas.

La idea para realizar esto es hacer una transformación del espacio de entrada en otro espacio de una dimensión mayor, en donde los patrones son linealmente separables y se pueden clasificar mediante un hiperplano. Considerar el siguiente mapeo:

$$\Phi: \mathfrak{R}^{N} \rightarrow \boldsymbol{H}, \tag{2.41}$$

donde N es la dimensión del espacio de entrada, y H es un espacio de alta dimensión, llamado el espacio de características. Una vez que los patrones se encuentren en el espacio de características se pueden usar las técnicas descritas en las secciones anteriores para encontrar un hiperplano óptimo de separación. Luego se debe realizar una transformación inversa hacia el espacio de entrada. Si se utiliza una transformación no lineal Φ, la superficie resultante en el espacio de entrada será no lineal.

En la Figura 2.5 se aprecia en la parte superior izquierda un espacio de entrada en dos dimensiones con patrones de dos clases linealmente no separables, este espacio es transformado en un espacio de dimensión superior en donde las clases sean linealmente separables, en este caso el espacio tridimensional donde se encuentra un hiperplano de separación óptimo, parte derecha de la figura. Por último se realiza una trasformación inversa para regresar al espacio de entrada donde se obtiene la frontera de decisión como se observa en la parte inferior izquierda de la figura.

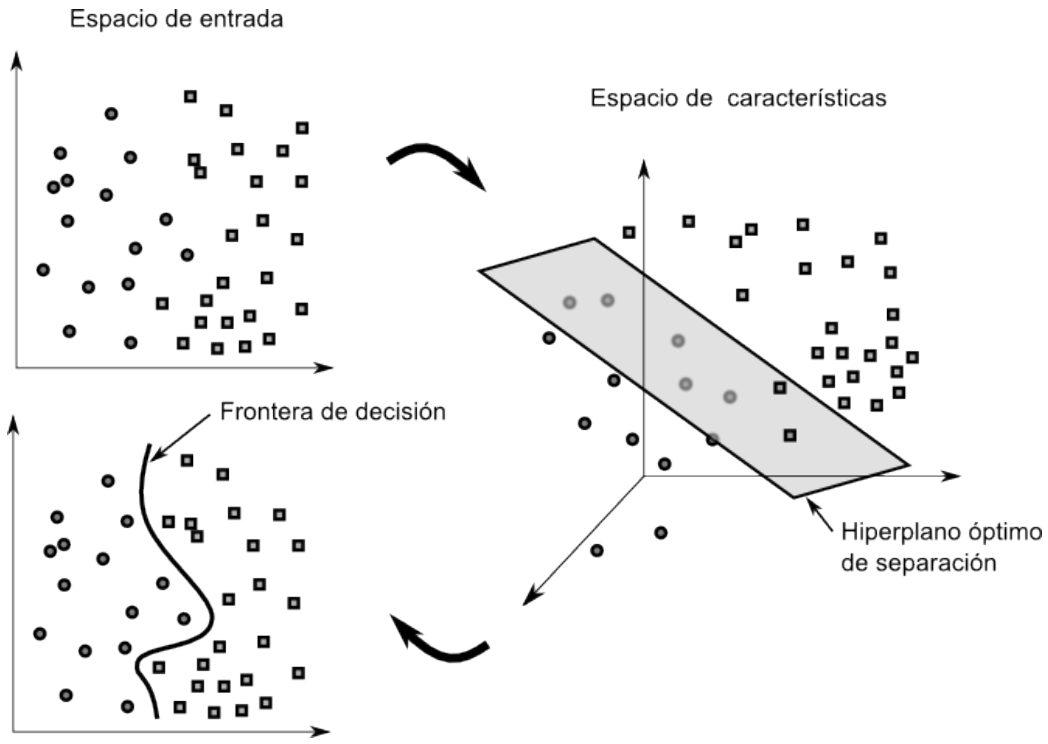


***Figura 2.5**: Transformación del espacio de entrada a un espacio de dimensión superior en donde se busca un hiperplano óptimo de separación para luego realizar una transformación inversa hacia el espacio de entrada.*

2.4.1. Kernels

Encontrar un hiperplano de separación óptimo en el espacio de características es con frecuencia complicado y tiene un alto costo computacional. En el entrenamiento de la máquina de aprendizaje, descrito en la Sección 2.3.1, notar que en la ecuación (2.29) solo se incluye a los datos de entrenamiento en forma de productos internos $\boldsymbol{x_i^T x_j}$. De esta forma, la transformación desde el espacio de entrada al espacio de características puede ser logrado mediante la sustitución del producto interno con:

$$\boldsymbol{x_i^T} \cdot \boldsymbol{x_j} \rightarrow \Phi^T(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j}). \tag{2.42}$$

Se define un funcional llamado kernel que es denotado por $K(\boldsymbol{x}, \boldsymbol{x_i})$ de la siguiente forma:

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = \Phi^T(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x_i}). \tag{2.43}$$

Este funcional es muy importante en SVM porque mediante el uso del kernel se puede construir hiperplanos óptimos en el espacio de características sin la necesidad de considerar el espacio de características por sí mismo de forma explícita. Consecuentemente, si un kernel apropiado es usado en el problema de clasificación, encontrar el hiperplano de separación puede ser realizado sin un sustancial incremento en el costo computacional.

No todas las funciones pueden ser utilizadas como kernel. Un kernel debe satisfacer el teorema de Mercer para su apropiada utilización en un SVM.

Teorema de Mercer

Sea $K(\boldsymbol{x}, \boldsymbol{y})$ un kernel simétrico continuo que es definido en el intervalo cerrado $\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$ y del mismo modo para $\boldsymbol{y}$. El kernel $K(\boldsymbol{x}, \boldsymbol{y})$ puede ser expandido en la serie:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(\boldsymbol{x}) \Phi_i(\boldsymbol{y}), \tag{2.44}$$

con coeficientes positivos, $\lambda_i > 0$ para todo i. Para que esta expansión sea válida y que está converja absoluta y uniformemente, es necesario y suficiente que la condición:

$$\int_b^a \int_b^a K(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{x})\psi(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y} \geq 0, \tag{2.45}$$

se cumpla para todo $\psi(\cdot)$ para el que:

$$\int_b^a \psi^2(\boldsymbol{x})d\boldsymbol{x} < \infty. \tag{2.46}$$

2.4.2. Clasificador con margen suave y kernels

La expansión del kernel $K(\boldsymbol{x}, \boldsymbol{x_i})$ nos permite construir una superficie de decisión que es no lineal en el espacio de entrada, pero su imagen en el espacio de características si es lineal. Ahora podemos plantear la forma dual del problema de optimización con restricciones de un SVM como sigue:

Dado el conjunto de entrenamiento $\{(\boldsymbol{x_i}, t_i)\}_{i=1}^N$ *, encontrar los multiplicadores de Lagrange* $\{\alpha_i\}_{i=1}^N$ *que maximize la función objetivo:*

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j K(\boldsymbol{x_i}, \boldsymbol{x_j}), \tag{2.47}$$

sujeto a las restricciones:

$$\sum_{i=1}^N \alpha_i \, t_i = 0, \tag{2.48}$$

$$0 \leq \alpha_i \leq C, \quad para\ i = 1,2, \dots, N \tag{2.49}$$

donde C *es un parámetro positivo ajustable.*

El problema dual que se acaba de presentar es de la misma forma que para el caso de un hiperplano con margen suave, excepto por el hecho de que el producto interno $\boldsymbol{x_i^T x_j}$ ha sido reemplazado por el kernel $K(\boldsymbol{x_i}, \boldsymbol{x_j})$.

2.4.3. Hipersuperficie de decisión

Sea la transformación no lineal Φ desde el espacio de entrada, de dimensión m_0, al espacio de características, podemos definir el hiperplano de decisión en el espacio de características como sigue:

$$\sum_{j=1}^{m_1} w_j \Phi_j(\boldsymbol{x}) + b = 0, \tag{2.50}$$

donde $\{w_j\}_{j=1}^{m_1}$ denota el conjunto de pesos que conectan el espacio de características con el espacio de salida, el valor de m_1 es la dimensión del espacio de características y b es el bias. La ecuación anterior puede ser simplificada como:

$$\sum_{j=0}^{m_1} w_j \Phi_j(\boldsymbol{x}) = 0, \tag{2.51}$$

donde se asume que $\Phi_0(\boldsymbol{x}) = 1$ para todo $\boldsymbol{x}$, por lo tanto w_0 denota al bias b. En términos de la "imagen" inducida en el espacio de características de los vectores de entrada $\boldsymbol{x}$, podemos definir el hiperplano de decisión en la forma compacta:

$$\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) = 0. \tag{2.52}$$

Adaptando la ecuación (2.24) a la presente situación, involucrando el espacio de características se tiene:

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \, t_i \boldsymbol{\Phi}(\boldsymbol{x_i}). \tag{2.53}$$

Por lo tanto, sustituyendo la ecuación (2.53) en (2.52), se puede definir la hipersuperficie de decisión como:

$$\sum_{i=1}^{N} \alpha_i \, t_i \boldsymbol{\Phi}^T(\boldsymbol{x_i}) \, \boldsymbol{\Phi}(\boldsymbol{x}) = 0, \tag{2.54}$$

esto puede ser expresado en función del kernel, por lo que la hipersuperficie óptima es ahora definida por:

$$\sum_{i=1}^{N} \alpha_i \, t_i K(\boldsymbol{x}, \boldsymbol{x_i}) = 0. \tag{2.55}$$

2.4.4. Funciones kernel frecuentes en SVM

Las funciones kernel que se utilizan con frecuencia en el reconocimiento de patrones son las siguientes:

- Kernel polinomial.
- Kernel para redes con función de base radial.

Para poder presentar mejor la diferencia entre estos dos kernels, se mostrara los resultados obtenidos de utilizar estos kernels en un SVM para clasificar los patrones de dos clases que son generados en forma aleatoria. Los patrones utilizados tendrán solo dos características a fin de poder mostrar gráficamente como son separadas las clases.

Máquina de aprendizaje polinomial

El uso de un kernel polinomial es muy común para resolver distintos problemas de clasificación. La ecuación (2.56) muestra un kernel de tipo polinomial.

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = (\boldsymbol{x}^T \boldsymbol{x_i} + 1)^p, \tag{2.56}$$

donde p es especificado a priori por el usuario.

En la Figura 2.6 se muestra la frontera de decisión que se ha obtenido para la clasificación de los patrones de dos clases. Los parámetros utilizados para la solución han sido $p = 5$ y $C = 7$. En la Figura 2.7 se muestra la superficie obtenida en esta solución usando un kernel polinomial.

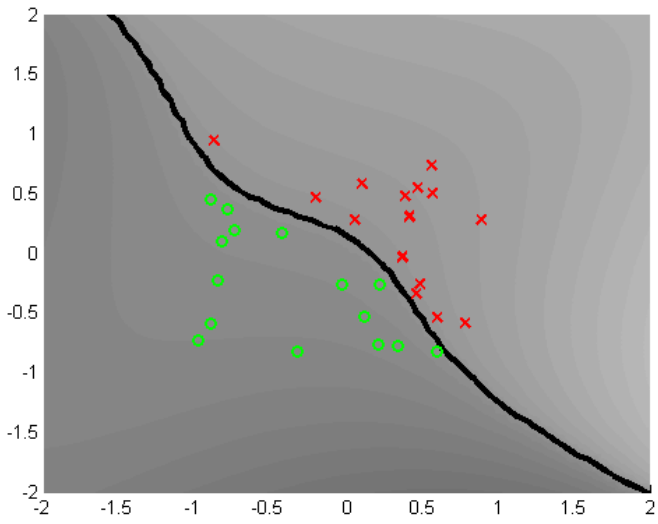


***Figura 2.6**: Frontera de decisión polinómica que clasifica patrones de dos características.*

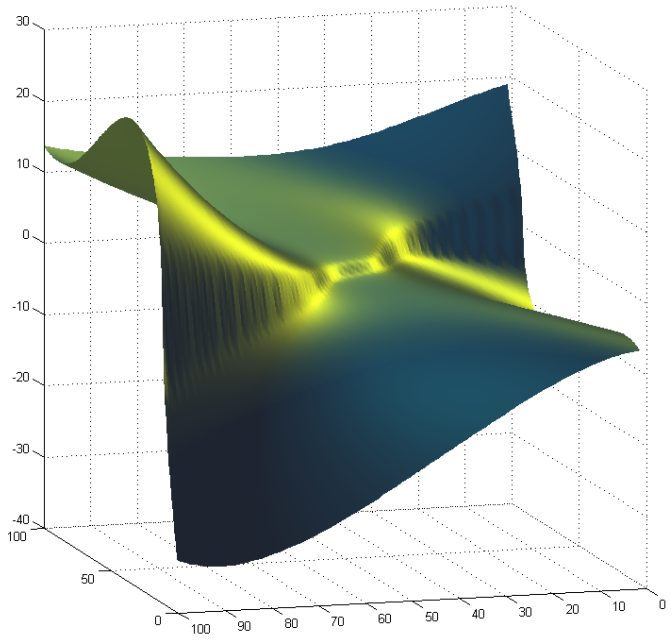


***Figura 2.7**: Superficie de clasificación polinómica.*

Redes con función de base radial

Las funciones de base radial han sido ampliamente estudiadas, en particular las del tipo gaussiano, la ecuación (2.57) muestra el kernel para redes de este tipo:

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{x_i}\|^2\right), \tag{2.57}$$

el ancho σ^2 es especificado a priori por el usuario.

En un SVM que tiene como kernel una función de base radial, el número de funciones de base radial y sus centros son determinados automáticamente en el entrenamiento del SVM por el número de vectores de soporte y sus valores respectivamente. La Figura 2.8 muestra la frontera de decisión para separar los patrones presentados anteriormente esta vez

utilizando un SVM que utiliza un kernel de base radial. Se han tomado como parámetros $\sigma = 0.5$ y $C = 7$. La Figura 2.9 muestra la superficie que separa estos patrones utilizando este kernel.

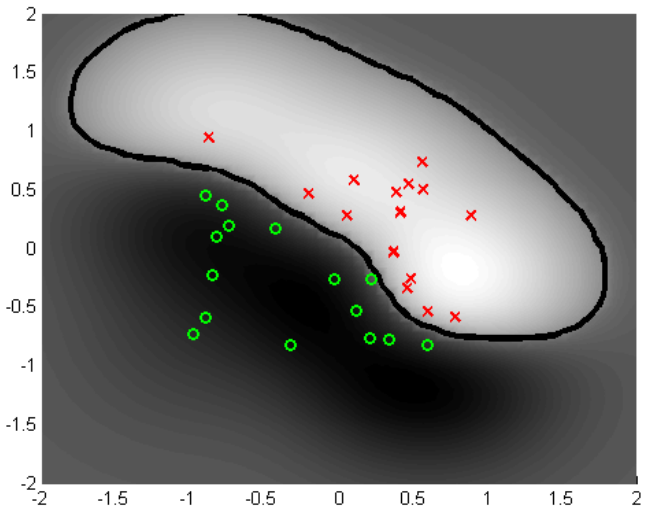


***Figura 2.8**: Frontera de decisión obtenida mediante un kernel de función de base radial.*

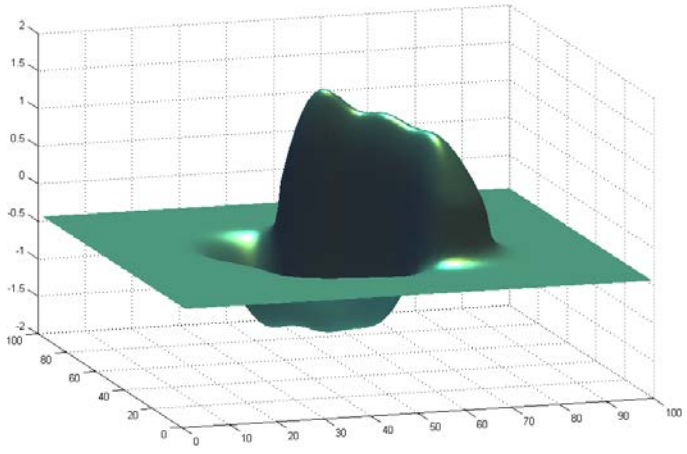


***Figura 2.9**: Superficie de clasificación para un kernel de base radial.*

En forma general, la arquitectura de un SVM es mostrada en la Figura 2.10 [11].

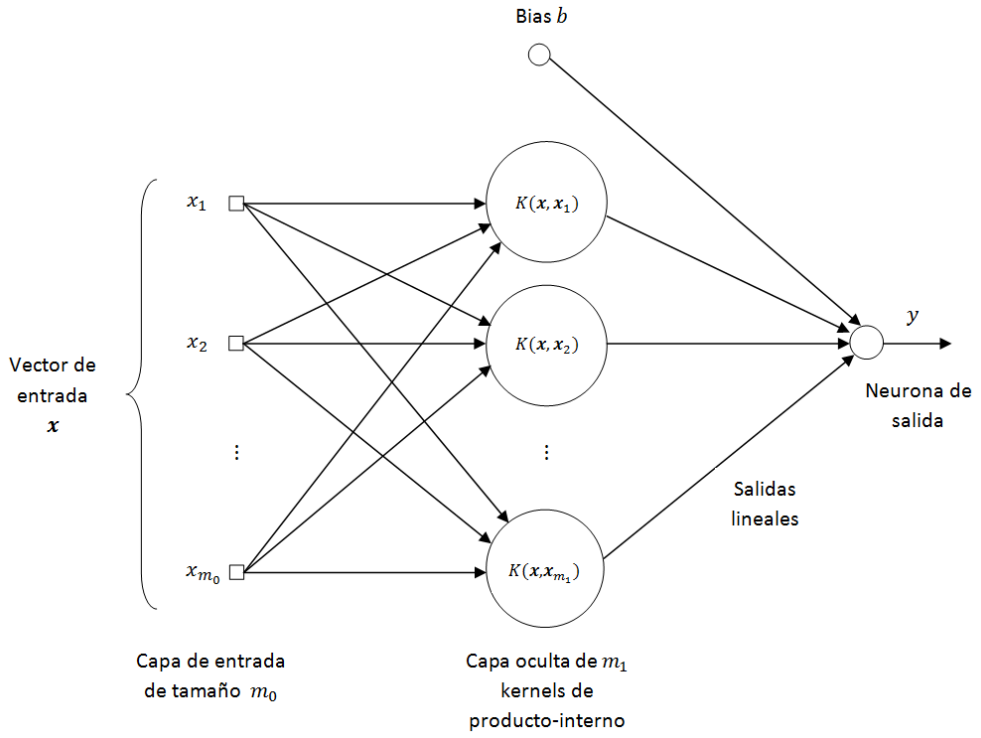


***Figura 2.10**: Arquitectura de un SVM.*

La dimensión del espacio de características (capa oculta) es normalmente muy alta para permitir la construcción del hiperplano de separación de las clases en ese espacio. Para una buena performance en la generalización, la complejidad del modelo es controlado mediante la imposición de ciertas restricciones en la construcción del hiperplano de separación, lo cual resulta en la extracción de una fracción de los datos de entrenamiento como vectores de soporte. El problema computacional involucrado cuando se trabaja con espacios de altas dimensiones es evitado por el uso de kernels y mediante la solución de la forma dual del problema de optimización con restricciones.

Capítulo 3

Visión estereoscópica

3.1. La visión humana frente a la visión por computador

Nosotros, los humanos, obtenemos información de nuestro entorno a través de nuestros sentidos: visión, gusto, olfato, tacto y oído, permitiéndonos estos percibir e interactuar con nuestro medio ambiente. Albert Einstein definió la realidad de la siguiente forma: “Reality is nothing more than the capability of our senses to be wrong”. Sin embargo, los sentidos humanos constituyen la única forma de adquirir información de la “realidad”, [13].

De entre todos los sentidos, la vista destaca como el más importante y complejo de todos. Hay treinta mil fibras nerviosas dedicadas al sentido auditivo mientras que el sistema visual presenta dos millones de fibras. Por tanto, es lógico suponer que dotar a las máquinas del “sentido de la vista” supondría un gran avance en su forma de operar.

La visión por computador intenta emular la forma en que los seres humanos perciben la información visual mediante el uso de cámaras ópticas actuando como

ojos y las computadoras para procesar la información en una “forma inteligente” como lo realiza el cerebro humano. Desafortunadamente, replicar el sentido de la vista es una tarea extremadamente compleja. La visión resulta fácil de realizar para el cerebro pero difícil de entender para nosotros. Ver es un proceso que realizamos cada día sin ningún esfuerzo y automáticamente, lo que es una ventaja para nosotros ocasiona un gran problema a la hora de intentar descubrir cómo funciona e internar replicarlo.

Descripción del modelo

La visión estereoscópica es el conjunto de técnicas que intentan recuperar información de profundidad a partir de dos o más vistas de una escena. Nuestros cerebros reciben imágenes similares de una escena tomadas desde dos puntos cercanos y en el mismo nivel horizontal debido a la forma en que están localizados los ojos. Dos objetos a distinta distancia del observador presentan posiciones relativas diferentes en sus imágenes. El cerebro es capaz de medir esta diferencia, disparidad retinal, y de usarla para estimar la profundidad. Los seres humanos utilizamos dos imágenes para realizar este proceso debido a que tenemos dos ojos, en este trabajo también se utiliza un sistema que adquiere dos imágenes para obtener información relacionada con la profundidad de la escena. Las siguientes secciones describen en detalle los conceptos implicados en el proceso de la visión estereoscópica.

3.2. Captación de escenas

Antes de analizar imágenes estereoscópicas el primer paso que se debe realizar es el de la captación de las escenas. En ocasiones se utilizan sistemas de múltiples sensores en número superior a dos con el objetivo de dar una mejor

solución al problema de correspondencia al tener un mayor número de restricciones que guiarían el proceso. Sin embargo, lo normal es utilizar un sistema de captación formado por dos cámaras que se encuentran separadas por una pequeña distancia fija entre si, a semejanza del sistema visual humano.

La calibración de cámaras es un proceso que normalmente se realiza como parte del conjunto de técnicas en el análisis de imágenes. Es posible abordar, sin embargo, el problema de la correspondencia estereoscópica independientemente de la etapa de calibración; pero si en algún momento debemos establecer algún tipo de relación entre las imágenes (proyecciones de la realidad) y las escenas reales, necesitamos realizar algún tipo de consideración con respecto al sistema de captación, es decir, algún tipo de consideración a cerca de la calibración de cámaras. Este trabajo estima la profundidad relativa cualitativamente de los objetos de una escena, por lo que el problema de correspondencia estereoscópica se realiza independientemente del proceso de calibración del dispositivo de captación.

3.3. Geometría epipolar de un sistema estereoscópico binocular

La geometría involucrada en un sistema compuesto por cámaras para capturar una misma escena tridimensional desde diferentes posiciones es denominada geometría epipolar, este es un concepto importante en un sistema estereoscópico binocular. La geometría epipolar provee las relaciones geométricas entre los puntos 3D de la escena y sus proyecciones en cada una de las imágenes. El conocimiento de estas relaciones geométricas restringe la búsqueda de los puntos correspondientes en las imágenes, de una búsqueda sobre toda la imagen a una búsqueda sobre una línea. La Figura 3.1 muestra la idea básica de la geometría epipolar en donde se presenta la geometría de dos cámaras de proyección central (pinhole), de acuerdo con este modelo, cada cámara está

representada por un lente puntual y un plano imagen situado a una distancia f detrás de la lente. Este modelo tiene el inconveniente de invertir las imágenes, por lo que con frecuencia se sustituye por otro equivalente en el que la lente se sitúa detrás del plano de la imagen. La proyección ortogonal al plano imagen del centro de proyección se denomina eje principal o eje óptico, los centros de proyección son denotados por O_L y O_R, planos de las imágenes I_L e I_R para la cámara izquierda y derecha respectivamente, y las distancias focales de cada cámara son f_L y f_R, que en el caso de la figura son iguales.

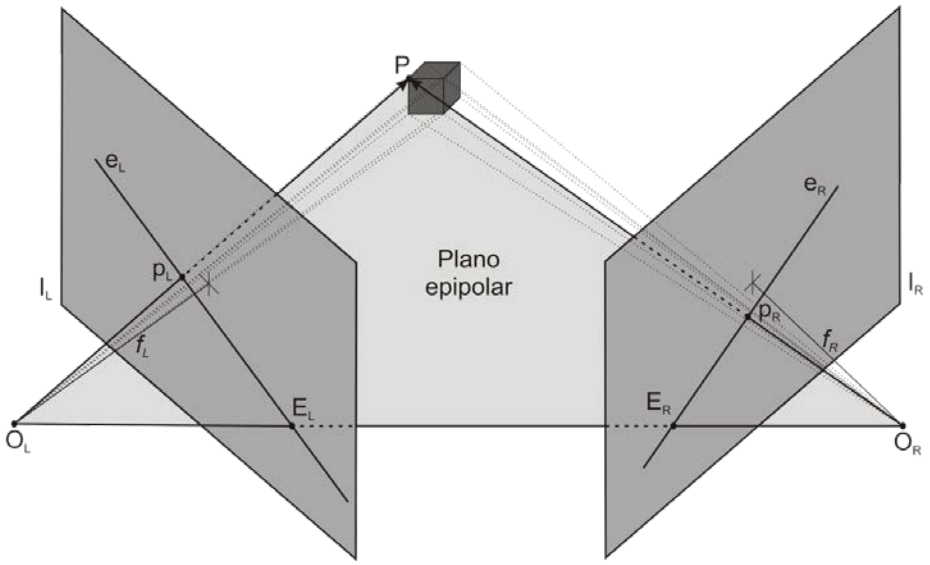


***Figura 3.1**: Geometría epipolar de un sistema estereoscópico binocular.*

La recta que une los centros de las cámaras $O_L O_R$ se denomina línea base (baseline), y corta a los planos de las imágenes I_L e I_R en los puntos E_L y E_R, estos puntos son los epipolos izquierdo y derecho respectivamente.

Dado un punto P de la escena su proyección en la cámara izquierda p_L se encuentra en la intersección de la recta $O_L P$ con el plano de la imagen izquierda I_L. De forma similar se halla p_R para la cámara derecha. Los puntos O_L, O_R y P definen

un plano que se denomina plano epipolar. La línea base pertenece a este plano pues O_L y O_R pertenecen a él por construcción. Más aún, considerando todos los puntos de la escena, cada uno de ellos define junto con O_L y O_R un plano epipolar, de forma que todos los planos epipolares contienen a la línea base. Así, la familia de los planos epipolares forma un haz de planos con eje en la línea base, y todo plano que contenga a la línea base es un plano epipolar.

El plano epipolar interseca a los planos de la imagen I_L e I_R de cada cámara en una recta que se denomina recta epipolar, e_L y e_R respectivamente. Todas las rectas epipolares pasan por el epipolo en la imagen respectiva formando un haz de rectas de vértice en el epipolo.

Para un punto P, las proyecciones p_L y p_R pertenecen al plano epipolar que define. Por lo tanto p_L y p_R pertenecen a las rectas epipolares e_L y e_R respectivamente. Ésta es la propiedad fundamental de la geometría epipolar definida, y se utiliza de la siguiente manera. Suponiendo que se conoce sólo p_L, P puede ser cualquiera de los puntos del rayo $O_L p_L$. Este rayo y la línea base definen el plano epipolar de P. La intersección del plano epipolar de P con I_R es la recta epipolar e_R, a la cual pertenece p_R. De esta forma la búsqueda del correspondiente de p_L no es necesario hacerla en toda la imagen I_R, se restringe a una búsqueda lineal en e_R [14].

Dependiendo de la posición relativa de las cámaras hay tres posibles configuraciones de acuerdo a la dirección de los ejes ópticos (véase Figura 3.2), si estos son paralelos hablaremos de geometría paralela o de ejes paralelos, en caso contrario los ejes podrán o no converger en un punto, si es así, se dirá que existe

un punto de fijación, y si éste no existe nos encontraremos ante el caso general de ejes ópticos no paralelos, que sin embargo deben orientarse de tal manera que las imágenes captadas por ambos sensores tengan una zona común que permita recuperar la información tridimensional.

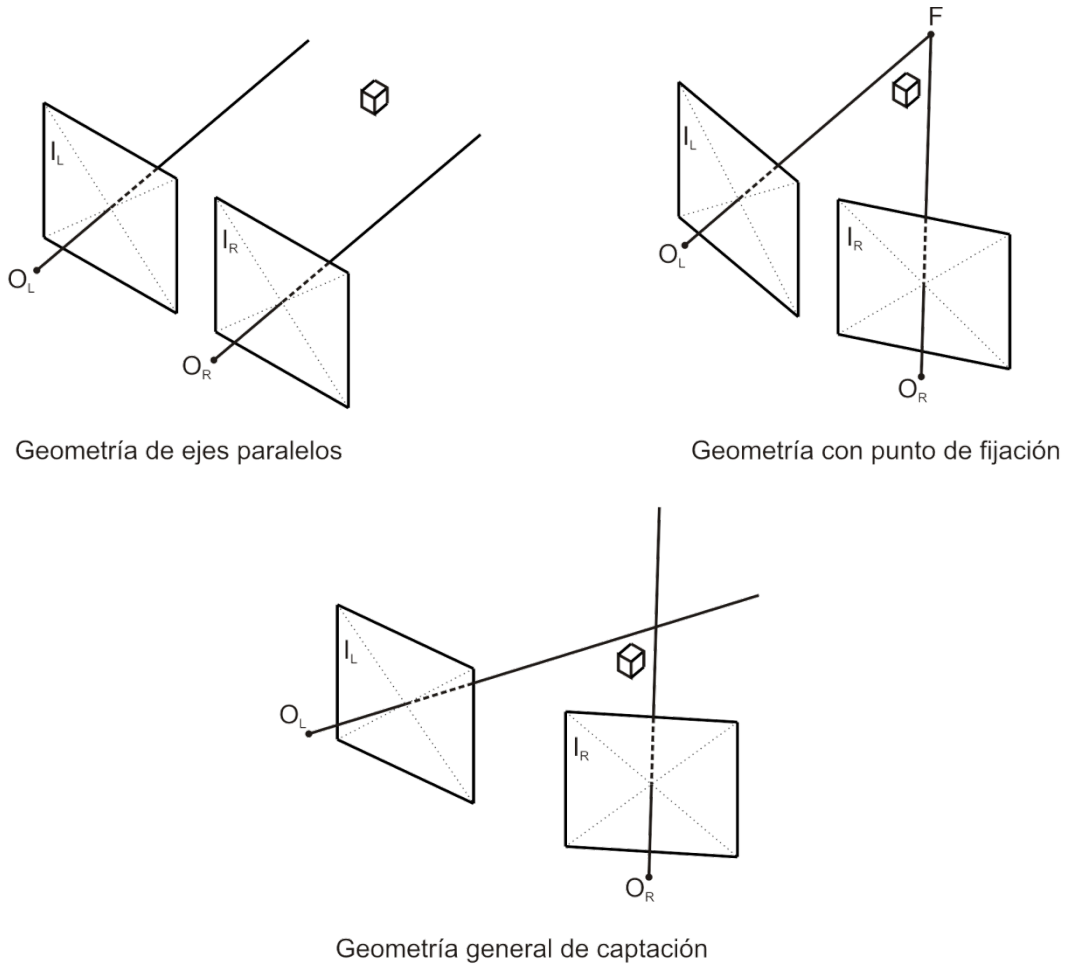


***Figura 3.2**: Posibles configuraciones de un sistema de captación binocular.*

La configuración utilizada en esta tesis es el de geometría de ejes paralelos, en donde ambos epipolos se encuentran en el infinito, pues la línea base es paralela a los planos de las imágenes. De esta forma el haz de rectas epipolares en cada imagen se transforma en un conjunto de rectas paralelas que son las filas de las imágenes izquierda y derecha.

Relación de la profundidad con la disparidad

Considerando una relación geométrica de semejanza de triángulos, la coordenada Z del punto P puede deducirse fácilmente sin más que observar la Figura 3.3.

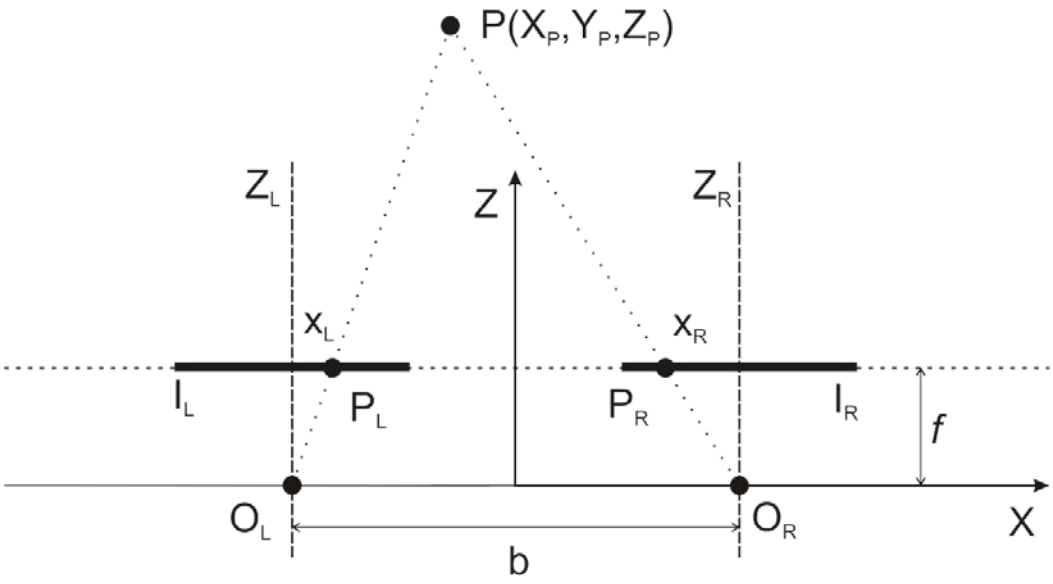


***Figura 3.3**: Representación de la proyección estereoscópica con ejes ópticos paralelos desde una perspectiva perpendicular a los planos de las imágenes.*

$$O_L: \quad \frac{\frac{b}{2}+X}{Z}=\frac{x_L}{f}, \quad O_R: \quad -\frac{\frac{b}{2}-X}{Z}=\frac{x_R}{f}, \tag{3.1}$$

despejando las variables x_L y x_R:

$$x_L=\frac{f}{Z}\left(X+\frac{b}{2}\right), \quad x_R=\frac{f}{Z}\left(X-\frac{b}{2}\right), \tag{3.2}$$

restando ambas ecuaciones:

$$x_L-x_R=d=\frac{fb}{Z} \quad \rightarrow \quad Z=\frac{fb}{d}. \tag{3.3}$$

De los resultados obtenidos en la ecuación (3.3) se deduce que cuando se utiliza esta geometría, la profundidad Z es inversamente proporcional a la disparidad de la imagen. La Figura 3.4 muestra la proyección estereoscópica desde otra perspectiva en donde se aprecia las líneas epipolares en los planos de las imágenes izquierda y derecha.

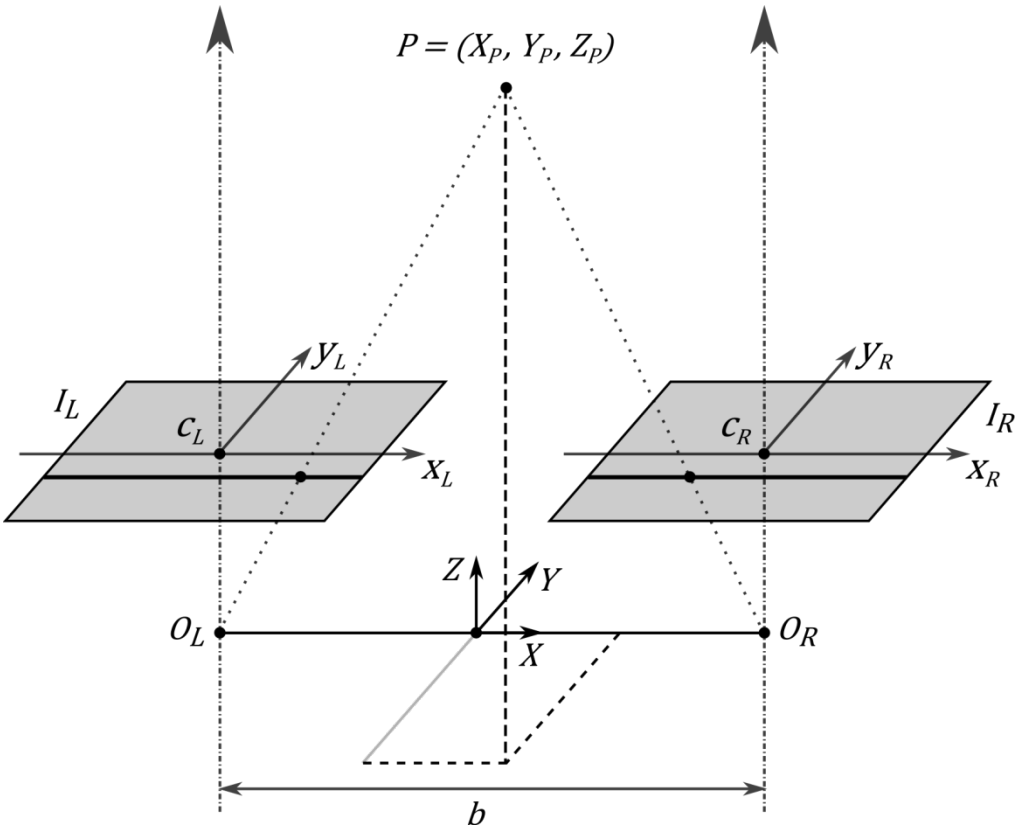


***Figura 3.4**: Proyección estereoscópica de un punto P.*

3.4. Rectificación de imágenes

La configuración de las cámaras con ejes ópticos paralelos se toma con frecuencia debido a su simplicidad, pero siendo físicamente posible, en la práctica resulta difícil alinear los ejes ópticos de una forma muy precisa si se tienen dos cámaras independientes, (existen sistemas de cámaras solidarias en donde los ejes ópticos se encuentran alineados en fábrica). Debido a esta complejidad,

generalmente se aplica el proceso de rectificado a las imágenes, el cual consiste en proyectar las imágenes del par estéreo, de forma que los planos de las imágenes en cada cámara sean paralelos entre sí, y paralelos a la dirección en la cual existe el desplazamiento entre las imágenes. Con esta configuración la búsqueda de puntos correspondientes se simplifica, pues se asegura que el correspondiente de un punto con coordenada vertical y_L en la imagen izquierda, se encuentra en la fila de coordenada $y_R = y_L$ de la imagen derecha. La Figura 3.5 muestra el proceso de rectificación.

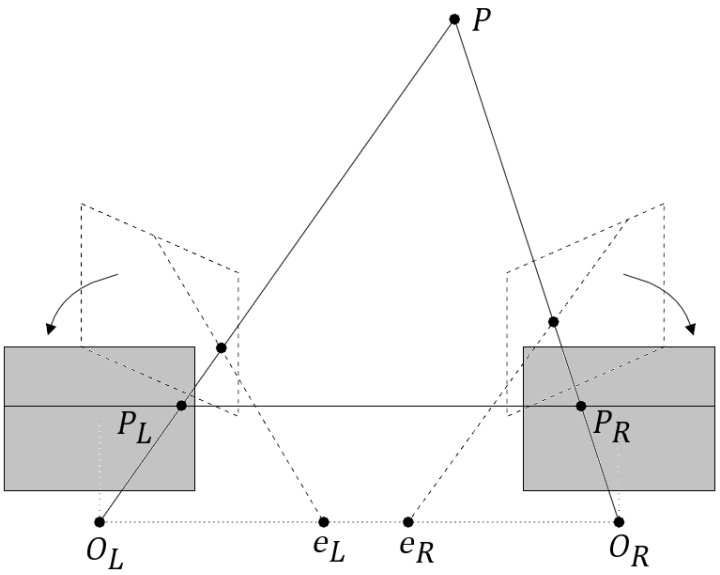


***Figura 3.5**: Rectificación de imágenes. Las imágenes rectificadas quedan paralelas a la recta que une los centros ópticos de las cámaras.*

Para realizar el proceso de rectificación es esencial el cálculo de una matriz denominada matriz fundamental, la cual contiene la geometría epipolar del sistema. La matriz fundamental F tiene dimensión 3×3 y relaciona los puntos correspondientes en las imágenes tal que:

$$\boldsymbol{x_R^T} F \boldsymbol{x_L} = 0. \tag{3.4}$$

En la ecuación (3.4), $\boldsymbol{x_R}$ y $\boldsymbol{x_L}$ son vectores columna de dimensión 3×1 en coordenadas homogéneas de los puntos correspondientes en las imágenes derecha e izquierda, respectivamente. Existen diferentes métodos para estimar la matriz fundamental que pueden ser clasificados en lineales, iterativos y métodos robustos. Uno de los métodos más sencillos para estimar la matriz fundamental es el algoritmo de ocho puntos en donde se debe conocer ocho correspondencias que son definidas manualmente o automáticamente seleccionadas de las imágenes. La rectificación de imágenes no será desarrollada en esta tesis, para mayores referencias sobre cómo realizar la rectificación paso a paso se puede revisar [14].

3.5. Correspondencia estereoscópica

Dada dos proyecciones de una escena, la correspondencia estereoscópica trata de encontrar los puntos homólogos en las imágenes que son la proyección de un único punto en el espacio. El problema de correspondencia es crucial en el proceso de visión estereoscópica. Las técnicas para resolver el problema de correspondencia tienen una alta carga computacional debido a que para cada pixel en la imagen izquierda existen varios candidatos, pixeles en la imagen derecha, que tienen que ser analizados para determinar la mejor correspondencia. De acuerdo a la cantidad de pixeles para los cuales es necesario hallar sus respectivos homólogos en la otra imagen, los métodos que intentan resolver el problema de correspondencia, pueden ser divididas en dos clases: métodos densos y dispersos.

3.5.1. Mapas densos vs dispersos

La visión estereoscópica dispersa se concentra en un grupo selecto de puntos característicos solamente. Un punto característico puede ser parte de un borde, segmentos de líneas o curvas, un punto de interés especial

para la aplicación. Alternativamente un punto característico puede ser un punto con un gran vecindario para una mejor correspondencia. Este enfoque puede potencialmente resultar en un rápido algoritmo de visión estereoscópica debido al reducido número de puntos a analizar y por ende a una reducida carga computacional.

A diferencia de los algoritmos para el cálculo de mapas dispersos, los algoritmos de visión estereoscópica densos encuentran los puntos correspondientes para todos los puntos en la imagen de referencia. En una ejecución completa de un algoritmo de visión estereoscópica denso se calcula el valor de la disparidad para cada pixel en la imagen de referencia. Típicamente este resultado es visualizado como una imagen en escala de grises, también conocido como mapa de disparidad. En esta imagen la intensidad de cada pixel corresponde a un nivel de disparidad, valores de disparidad altos resultan en pixeles más claros. En esta tesis se utiliza un mapa de disparidad denso debido a que información relacionada con la profundidad de los objetos en la escena es necesaria para las siguientes etapas del procesamiento.

3.5.2. Restricciones aplicadas a los métodos de correspondencia

Antes de analizar los métodos utilizados para resolver el problema de la correspondencia, se presenta algunas de las restricciones aplicadas generalmente en los métodos para hallar puntos correspondientes, de esta forma el proceso se vuelve más sencillo y se obtienen resultados más confiables.

- *Restricción epipolar.* Esta dada por la geometría epipolar del par estereoscópico, e implica que el correspondiente de un punto en una imagen debe estar en la recta epipolar del punto en la otra imagen. Esta restricción reduce la búsqueda del correspondiente de toda la imagen a una recta en la misma.

- *Restricción de orden.* Implica que si la proyección del objeto Q está a la izquierda de la proyección del objeto P en la imagen izquierda, entonces la proyección de Q estará a la izquierda de la proyección de P en la imagen derecha. Esta restricción no siempre se cumple en las escenas y conlleva a ambigüedades en el problema de correspondencia, un ejemplo de esto se muestra en la Figura 3.6. Como comentario adicional podemos decir que el sistema visual humano también presenta ambigüedades al recuperar la estructura correcta de la escena cuando no se cumple esta restricción, un ejemplo de esto se tiene cuando se observa un lápiz en posición vertical, sujeto con la mano y por detrás otro objeto “fino”, por ejemplo una columna; dependiendo en cuál de los objetos se enfoque la vista, el otro objeto se presenta como si estuviera presente dos veces en la escena o presentando una transparencia inexistente.

- *Restricción de unicidad.* Implica que cada punto de una imagen puede tener no más de un correspondiente en la otra imagen. Esta restricción contempla que pueda no existir ningún correspondiente, como puede ser en el caso que este oculto en la otra imagen.

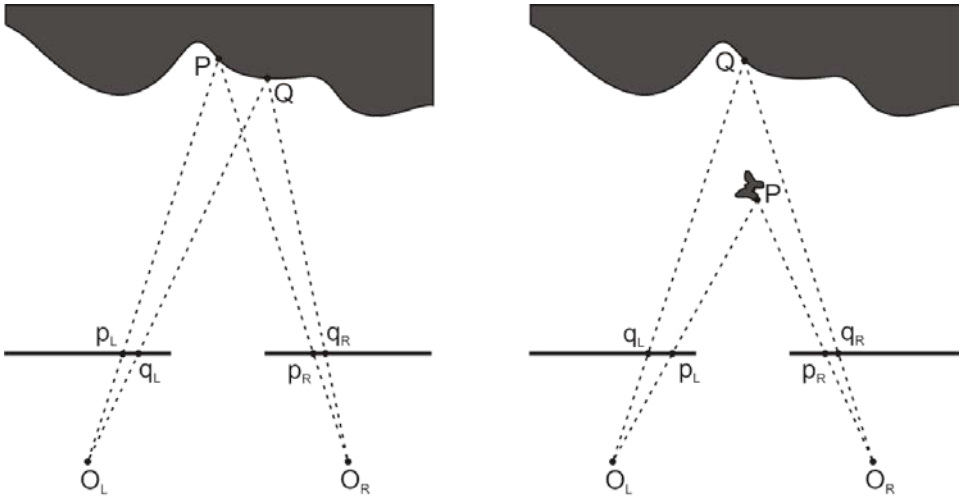


Figura 3.6*: De izquierda a derecha, escena que cumple la restricción de orden y escena que no cumple la restricción de orden.*

- *Restricción de semejanza.* Implica que las características de los puntos en una imagen (intensidad o color, etc.) no deben tener un cambio significativo entre ambas imágenes.

- *Restricción de disparidad.* Esta restricción está basada en la geometría de los objetos e impone un rango de disparidades posibles para los objetos de la escena. Capturando las imágenes de la escena a una distancia media se descarta la posibilidad de que existan objetos muy cercanos, de esta forma se eliminan las disparidades excesivamente grandes. Esta restricción limita la búsqueda en toda la línea epipolar, a una búsqueda restringida a un segmento de ésta línea. Trabajos relacionados con la restricción de disparidad se puede revisar en [15].

3.5.3. Fenómenos involucrados en el problema de la correspondencia

Un fenómeno a tener en cuenta en el planteo de la visión estereoscópica y que influye en el proceso de hallar los puntos correspondientes, son las oclusiones. La geometría de la escena y del sistema de captación provocara la aparición de zonas que no podrán ser vistas al menos por una de las cámaras como se muestra en la Figura 3.7, zonas en las que por lo tanto no se podrá establecer la correspondencia. Los métodos de correspondencia han abordado de varias formas el problema de la oclusión, algunos clasifican los métodos según como hacen este abordaje en: métodos que detectan las oclusiones, métodos que disminuyen la sensibilidad a las oclusiones, y los métodos que modelan las oclusiones.

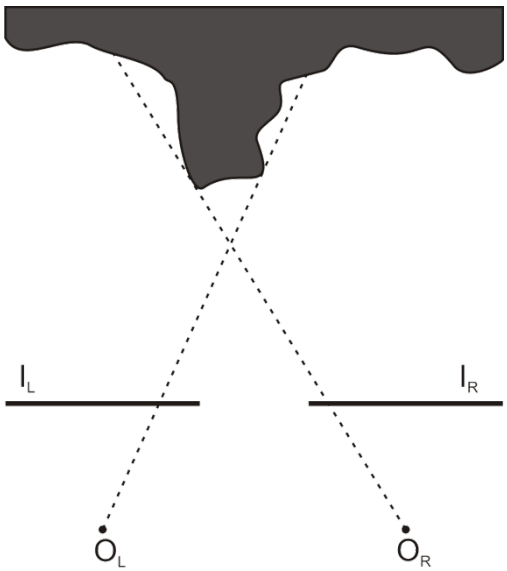


***Figura 3.7**: Escena que presenta oclusiones.*

Existen otros problemas que deben considerarse, por ejemplo la diferente intensidad luminosa con que las cámaras captan la escena; incluso

si las cámaras se comportasen de manera totalmente idéntica en este sentido, la diferente posición de las mismas respecto de la escena y el foco o focos de luz puede provocar diferencias notables de luminosidad en la captación de las imágenes.

Las escenas con poca textura constituyen también un problema para el establecimiento de la correspondencia, puesto que en esta situación, no es posible realizar una correlación con resultados fiables, obteniendo posibles asignaciones ambiguas, las cuales deberán ser resueltas a partir de restricciones de continuidad entre otras.

3.5.4. Métodos de correspondencia

Los métodos implementados en esta tesis se han clasificado de acuerdo a como utilizan la información de las imágenes para encontrar los puntos homólogos. Esta clasificación divide a los métodos en aquellos que utilizan restricciones locales en una ventana alrededor del punto de interés, y aquellos que imponen restricciones globales en las líneas epipolares o en la imagen. Los métodos locales tienen una mayor eficiencia computacional pero presentan dificultades en regiones localmente ambiguas de las imágenes, como son las oclusiones, o regiones de textura uniforme. Los métodos globales son más robustos frente a estos problemas, pero son computacionalmente más costosos. Los métodos locales se pueden sub-clasificar según el método que utilizan para la detección de los correspondientes en:

- *Correlación basada en ventanas*. Se basa en encontrar el punto correspondiente comparando una región (ventana que representa la

muestra) alrededor del punto de interés con un conjunto de regiones de igual estructura extraídas de la otra imagen. La región que presenta mayor semejanza es la elegida y se selecciona el punto correspondiente.

- *Flujo Óptico*. Se basa en plantear una ecuación diferencial que relaciona el desplazamiento d, de un pixel entre las imágenes izquierda y derecha, con el movimiento, asumiendo que su intensidad no varía

$$\nabla_x Id + I_t = 0, \tag{3.5}$$

donde $\nabla_x I$ es el gradiente horizontal de la imagen e I_t es la derivada "temporal". El desplazamiento vertical del pixel se asume que es nulo, $\nabla_y I = 0$, dada la configuración del par estereoscópico. El movimiento en este caso es entre la imagen izquierda y la derecha, no existe una variación temporal entre las imágenes de la escena.

- *Correlación de características.* Se basa en buscar los puntos correspondientes en regiones de la imagen donde existen características relevantes, (vértices, bordes, etc). Estos no son demasiados, y los mapas de disparidad que se pueden calcular no son densos; se obtienen puntos ubicados en el espacio pero no un mapa de disparidad que tiene información acerca de la profundidad para cada punto de la imagen.

La complejidad de los dos primeros métodos es similar y ambos muestran problemas en las discontinuidades de la profundidad presentes en

la escena, y en regiones de textura uniforme. El tercero tiene como principal desventaja no poder calcular un mapa de disparidad denso.

Los métodos globales, imponen restricciones globales en la minimización de alguna expresión de costo o energía que modele el fenómeno estereoscópico, reduciendo los errores en las regiones con ambigüedades en la determinación de puntos homólogos. Dentro de estos métodos se encuentran la programación dinámica y el corte de grafos.

- *Programación dinámica*. Es un método que reduce la complejidad de cálculo en problemas de optimización descomponiendo el problema en sub-problemas menores. Las restricciones que se imponen con este método son, generalmente, la restricción epipolar y la restricción de orden. Para esto se construye una representación de las posibles correspondencias para cada punto construyendo una imagen que se denomina *imagen del espacio de disparidad,* donde se busca un camino que recorra este espacio y minimice un cierto costo. La mayor desventaja de este método es que, generalmente, no agrega una coherencia vertical entre las líneas adyacentes en la búsqueda de las correspondencias a lo largo de líneas epipolares horizontales, dada la configuración del par estereoscópico esto provoca un efecto rayado horizontal en el mapa de disparidad.

- *Corte de grafos.* El corte de grafos se basa en armar un grafo a partir de los datos de las imágenes y buscar un corte mínimo. Dependiendo como se arma el grafo, el resultado obtenido es la minimización de una cierta expresión de energía. Este procedimiento se puede considerar análogo al de hallar el mejor camino en una

imagen bidimensional con programación dinámica, pero extendido a tridimensionalidad con coherencias en las dos dimensiones. El resultado es una superficie que minimiza un costo energético sobre un grafo plano. Estos métodos tienen un costo computacional mayor que la programación dinámica, pero en los últimos años se han desarrollado nuevas implementaciones que reducen sensiblemente este costo computacional [14].

Los métodos de correspondencia globales son un área de investigación activa, y las recientes técnicas de minimización de energía como el corte de grafos resuelven el problema de correspondencia de forma muy precisa. Sin embargo, a una mejor precisión en los resultados, mayor es la complejidad del método y mayor el tiempo de procesamiento del mismo. Dado que para esta tesis no se necesita una alta precisión en el mapa de disparidad para el reconocimiento y clasificación de objetos, y a fin de reducir la carga computacional se ha considerado un método local y un método global que no presentan una alta complejidad. En las siguientes secciones se analiza los métodos de correlación estereoscópica implementados en esta tesis.

Correlación basada en ventanas

Los métodos locales como la correlación basada en ventanas tienen gran popularidad en la correlación estereoscópica por su alta eficiencia. Este método asume que la vecindad de los pixeles homólogos tiene niveles de grises similares. La correspondencia se realiza mediante la comprobación de la similitud de cada dos pixeles. Sin embargo, debido a problemas como diferente intensidad luminosa, ruidos en las imágenes, entre otros, es

probable que los dos pixeles correspondientes apenas tengan el mismo valor de intensidad. Si se realiza la búsqueda de correspondencias utilizando solamente un pixel para el punto de interés, numerosos errores serán detectados. La mejor forma de tomar esto en consideración es incluir la vecindad del punto de interés en la búsqueda de su correspondiente. Esta idea conduce al método de correlación basada en ventanas.

Una ventana centrada en el pixel de interés en la imagen izquierda se compara con otra ventana que es desplazada sobre todos los pixeles candidatos en la imagen derecha (véase Figura 3.8). Si un par de ventanas son idénticas bajo algún criterio en particular, entonces los pixeles en los centros son probablemente pixeles correspondientes de un punto capturado en la escena. Este método usualmente usa ventanas rectangulares o cuadradas centradas en el pixel de interés.

En lugar de buscar el par de ventanas con mayor similitud, el proceso puede ser simplificado encontrando el par de ventanas con menor disimilitud usando los costos de correspondencia.

Costos de correspondencia

Los costos de correspondencia brindan una forma de determinar cuantitativamente las diferencias entre dos pixeles en el par estereoscópico. Para identificar el pixel derecho que es homologo al pixel izquierdo que está siendo considerado, un costo de correspondencia es calculado para cada pixel candidato en la imagen derecha.

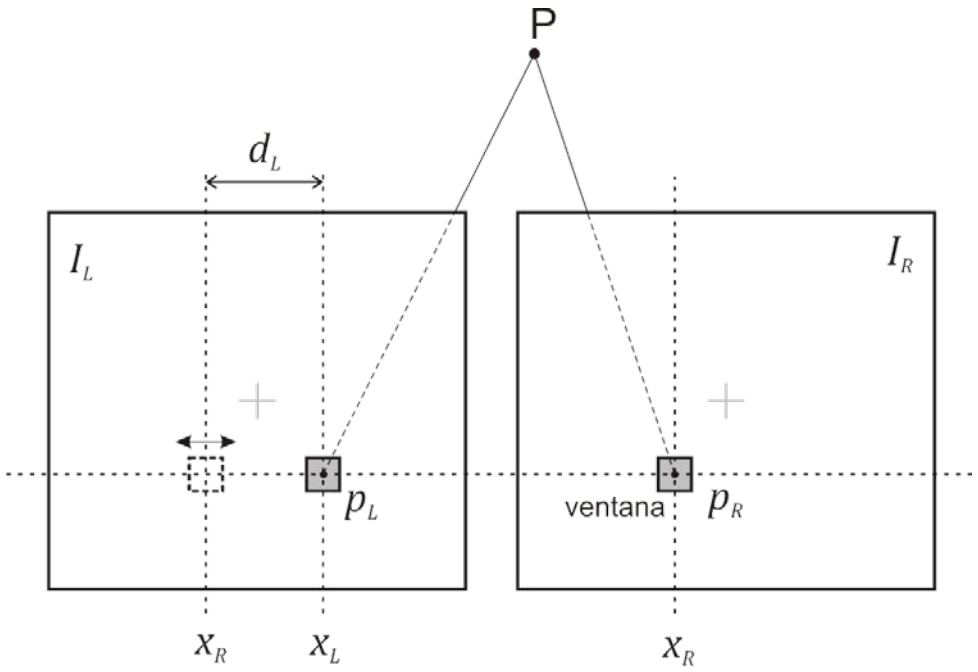


***Figura 3.8**: Disparidad entre pixeles homólogos en la correlación basada en ventanas.*

Los costos de correspondencia más simples asumen que los niveles de gris en los pixeles homólogos son iguales. Costos más robustos consideran explícita o implícitamente ciertas distorsiones radiales y/o ruido en las imágenes. A continuación se presenta las funciones de costo más usadas en los métodos de correlación basada en ventanas en los que se encuentra la suma de diferencias al cuadrado o SSD (del inglés *Sum of Squared Differences*), suma de diferencias absolutas o SAD (del inglés *Sum of Absolute Differences*), y la correlación cruzada normalizada o NCC (del inglés *Normalized Cross Correlation*).

a. Suma de diferencias al cuadrado (SSD)

En este costo, la diferencia al cuadrado o SD (del inglés Squared Difference), es usado para medir la disimilitud entre los niveles de gris de los pixeles. La diferencia al cuadrado entre en nivel de gris $I_L(x_L, y)$ del pixel p_L

con coordenadas (x_L, y) en la imagen izquierda y el nivel de gris $I_R(x_L - d, y)$ del pixel candidato p_R en el desplazamiento d con coordenadas $(x_L - d, y)$ en la imagen derecha puede ser escrito como:

$$SD(x_L, y, d) = [I_L(x_L, y) - I_R(x_L - d, y)]^2. \quad (3.6)$$

La sumatoria de las diferencias al cuadrado en una ventana de tamaño $(2w + 1) \times (2w + 1)$ centrado en p_L y su ventana similar centrada en p_R es llamado suma de diferencias al cuadrado y es definido para imágenes en escala de grises como:

$$SSD(x_L, y, d) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} (I_L(x_L + i, y + j) - I_R(x_L + i - d, y + j))^2, \quad (3.7)$$

donde w es la mitad del ancho de la ventana.

La función de costo SSD, es calculado para todos los posibles candidatos en la imagen derecha, el desplazamiento de la ventana para el cual la función es mínimo es seleccionado (véase Figura 3.9). Así, la disparidad estimada $d_L(x_L, y)$ en el pixel p_L corresponde al desplazamiento d del pixel en la imagen derecha el cual minimiza la función de costo. Esto es expresado como:

$$d_L(x_L, y) = \min_d (SSD(x_L, y, d)). \quad (3.8)$$

Se usa el subíndice L en el símbolo de la disparidad estimada debido a que un pixel en la imagen izquierda es usado como referencia en el cálculo de la función de costo, donde d varía desde d_{min} a d_{max}.

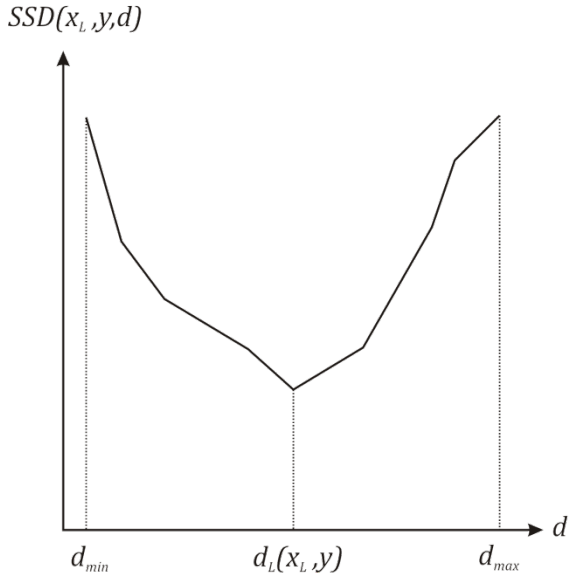


***Figura 3.9**: Minimización de la función de costo SSD, $[d_{min}, d_{max}]$ representa los posibles desplazamientos para el pixel buscado.*

Una vez que la disparidad ha sido estimada para cada pixel de la imagen izquierda, el mapa denso de disparidad para la imagen derecha es formado. El mapa de disparidad es un arreglo de los valores calculados de la disparidad para cada pixel, el cual tiene el mismo tamaño que las imágenes de entrada.

b. Suma de diferencias absolutas (SAD)

La función de costo para la suma de diferencias absolutas SAD, puede ser formulado como:

$$SAD(x_L, y, d) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} |I_L(x_L + i, y + j) - I_R(x_L + i - d, y + j)|. \quad (3.9)$$

Esta función de costo es similar a la suma de diferencias al cuadrado con la ventaja de que se reduce ligeramente la carga computacional en el cálculo del mapa de disparidad.

c. Correlación cruzada normalizada (NCC)

La correlación cruzada normalizada es definida como:

$$NCC(x_L, y, d) = \frac{\sum_{i=-w}^{w} \sum_{j=-w}^{w} |I_L(x_L + i, y + j) \times I_R(x_L + i - d, y + j)|}{NormC_L(x_L, y) \times NormC_R(x_L - d, y)}, \quad (3.10)$$

donde $NormC_L(x_L, y)$ y $NormC_R(x_L - d, y)$ son los coeficientes normalizados correspondientes a los pixeles de la ventana de tamaño $(2w + 1) \times (2w + 1)$, centrado respectivamente en (x_L, y) en la imagen izquierda y en $(x_L - d, y)$ en la imagen derecha. Los coeficientes normalizados son expresados como:

$$NormC_L(x_L, y) = \sqrt{\sum_{i=-w}^{w} \sum_{j=-w}^{w} I_L(x_L + i, y + j)^2}, \quad (3.11)$$

$$NormC_R(x_L - d, y) = \sqrt{\sum_{i=-w}^{w} \sum_{j=-w}^{w} I_R(x_L + i - d, y + j)^2}. \quad (3.12)$$

La correlación cruzada normalizada es una función de costo estadística, su normalización tanto en la media como en la varianza la

hacen insensible a cambios de ganancia y de offset en las cámaras. Sin embargo el cálculo de esta función es computacionalmente más alto a diferencia de los costos SSD y SAD que son medidas más sencillas y a menudo suficientemente buenas.

Programación dinámica para la correspondencia estereoscópica

Como se menciono anteriormente, los métodos de correspondencia global hacen uso de consideraciones integrales para reducir la sensibilidad en las imágenes que tienen problemas de oclusión o que presentan texturas uniformes. No es necesario definir consideraciones globales sobre toda la imagen, en otras palabras, es suficiente definir un costo global entre dos líneas epipolares correspondientes en ambas imágenes. La programación dinámica es una técnica de programación que se usará para resolver el problema de encontrar una secuencia de disparidades óptimo desde un lado de la imagen al lado opuesto, esto puede ser comparado al de encontrar la ruta óptima entre el punto inicial y final en una línea epipolar.

Las imágenes estereoscópicas utilizadas se consideran rectificadas, y las filas de las imágenes izquierda y derecha son correspondientes una a una entre sí. Estas filas son denominadas *scanlines*, y la correspondencia entre ellas estará dada como $S_L \leftrightarrow S_R$ donde S_L es la scanline de la imagen izquierda y S_R la scanline de la imagen derecha.

La idea general detrás de éste método es tratar el problema de correspondencia como un problema de minimización de energía, donde el objetivo es encontrar una solución d que minimice una energía global.

$$E_T(d) = E_d(d) + E_s(d), \tag{3.13}$$

donde:

$$E_d: Energía\ de\ datos,$$

$$E_s: Energía\ de\ suavizado.$$

La función de energía E_T representa el costo total de una secuencia d de pixeles correspondientes en una scanline y consiste en un término relacionado a los datos de entrada y en una energía de suavizado.

La energía de datos, $E_d(d)$, en la ecuación anterior es una medida de la disimilitud entre los pixeles de las imágenes del par estereoscópico. La energía de suavizado, $E_s(d)$, puede ser formulado para manipular las discontinuidades en profundidad y las oclusiones. Para la aplicación de este método primero se debe construir el espacio de disparidad, el cual puede ser representado por una imagen en el que cada punto de la imagen debe ser definido usando alguno de los costos de correspondencia local que se presentaron en las secciones anteriores.

Hay dos formas de construir el espacio de disparidad (véase Figura 3.10), la primera de ellas consiste en definir los ejes vertical y horizontal como las líneas izquierda y derecha de barrido. En este caso, la programación dinámica se utiliza para determinar el camino de mínimo costo que va desde una de las esquinas del espacio de disparidad hasta la esquina opuesta. La segunda forma de construir el espacio de disparidad es definiendo los ejes como la scanline izquierda y el rango de disparidad. Para

este caso, la programación dinámica determina el camino de menor costo para llegar desde un lado del espacio de disparidad al lado opuesto.

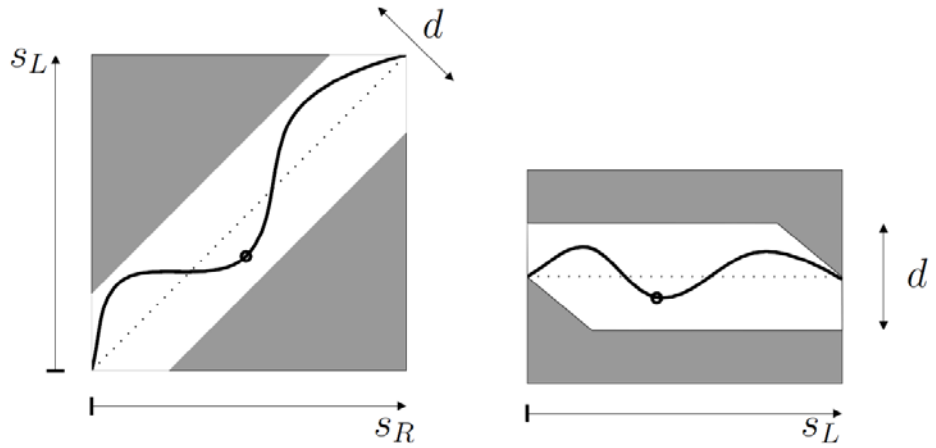


***Figura 3.10**: Formas de construir el espacio de disparidad. De izquierda a derecha, imagen del espacio de disparidad utilizando la primera y segunda propuesta respectivamente [14].*

Bajo este enfoque, la ruta óptima es buscada en un arreglo de correspondencias bidimensional representando todas las posibles correspondencias entre dos puntos de las imágenes. Un ejemplo de la búsqueda de la ruta óptima es presentado en la Figura 3.11. Cada celda en el arreglo bidimensional es una medida de una potencial correspondencia entre dos pixeles, una gran cantidad de correspondencias entre pixeles es excluida gracias a la restricción de orden y al rango de disparidad permitido. Por ejemplo, la restricción de orden prohíbe la correspondencia entre pixeles izquierdos que tienen coordenada $x_L$con pixeles derechos con coordenada x_R mayor a x_L.

Las celdas que cumplen con esta restricción de orden se encuentran sobre la diagonal principal en el arreglo de correspondencias. La restricción de disparidad impone los valores de la disparidad a un rango específico,

permitiendo remover las celdas en la esquina superior derecha del arreglo de correspondencias.

El valor de cada celda está asociado a un costo local, tal como SSD o SAD, la energía de datos o también nombrado costo de datos de una ruta en particular es igual a la suma de los costos locales de todas las celdas por donde cruza la ruta. La energía de suavizado o costo de suavizado puede ser calculado de diferentes formas en donde por lo general se considera el costo local de los vecinos del pixel de interés. El costo de suavizado que se utiliza en esta tesis se describe en el Capítulo 4.

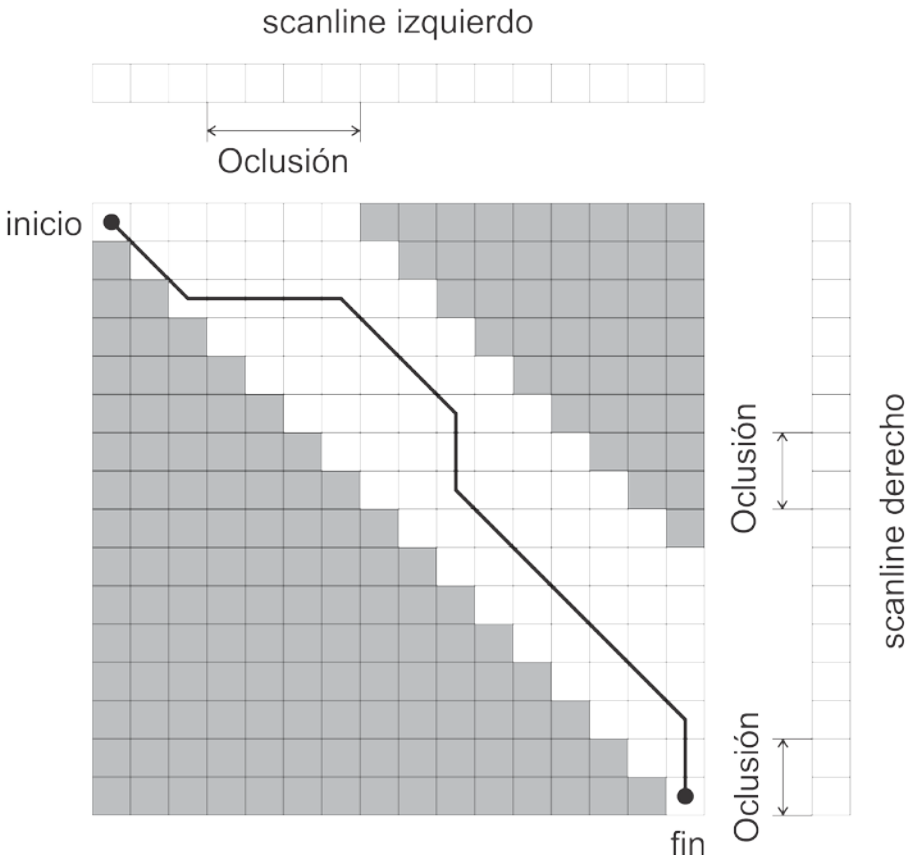


***Figura 3.11**: Correspondencia estereoscópica usando el método de programación dinámica.*

La mayor ventaja de usar este método usando un costo de suavizado en una dimensión es que provee un soporte global para regiones que son pobremente texturadas y que otros métodos de correspondencia basados en medidas locales darían como resultado correspondencias incorrectas. El costo computacional de métodos que usan el costo de suavizado en una dimensión usando programación dinámica es baja comparado con otros enfoques globales que imponen costos de suavizado en las dos dimensiones. Sin embargo, la desventaja de este método es que errores locales pueden generar correspondencias incorrectas a lo largo de toda una scanline.

Capítulo 4

Algoritmos para el reconocimiento de objetos

4.1. Niveles de procesamiento

La compleja tarea de reconocer objetos desde una escena tridimensional es dividida en un conjunto de procesos más simples de tal forma que el sistema tiene una mayor modularidad y se puede realizar un seguimiento de cada una de las etapas para su correcto funcionamiento. En cada una de estas etapas o niveles de procesamiento se va refinando y reduciendo la cantidad de información hasta llegar a la descripción deseada. Los algoritmos implementados en esta tesis se han clasificado en tres niveles de acuerdo al procesamiento que realizan, la Figura 4.1 muestra las etapas del sistema de reconocimiento con los tres niveles de procesamiento. Este capítulo describe los algoritmos implementados en esta tesis, los cuales se mencionan a continuación: algoritmos para el realce de las imágenes de entrada, y la correlación estereoscópica, algoritmos para la segmentación y extracción de características, y finalmente los algoritmos basados en SVM para reconocer los objetos. Todos estos algoritmos son parte de la tarea de

reconocimiento de objetos y son necesarios para que el sistema de reconocimiento cumpla con los objetivos que se han planteado.

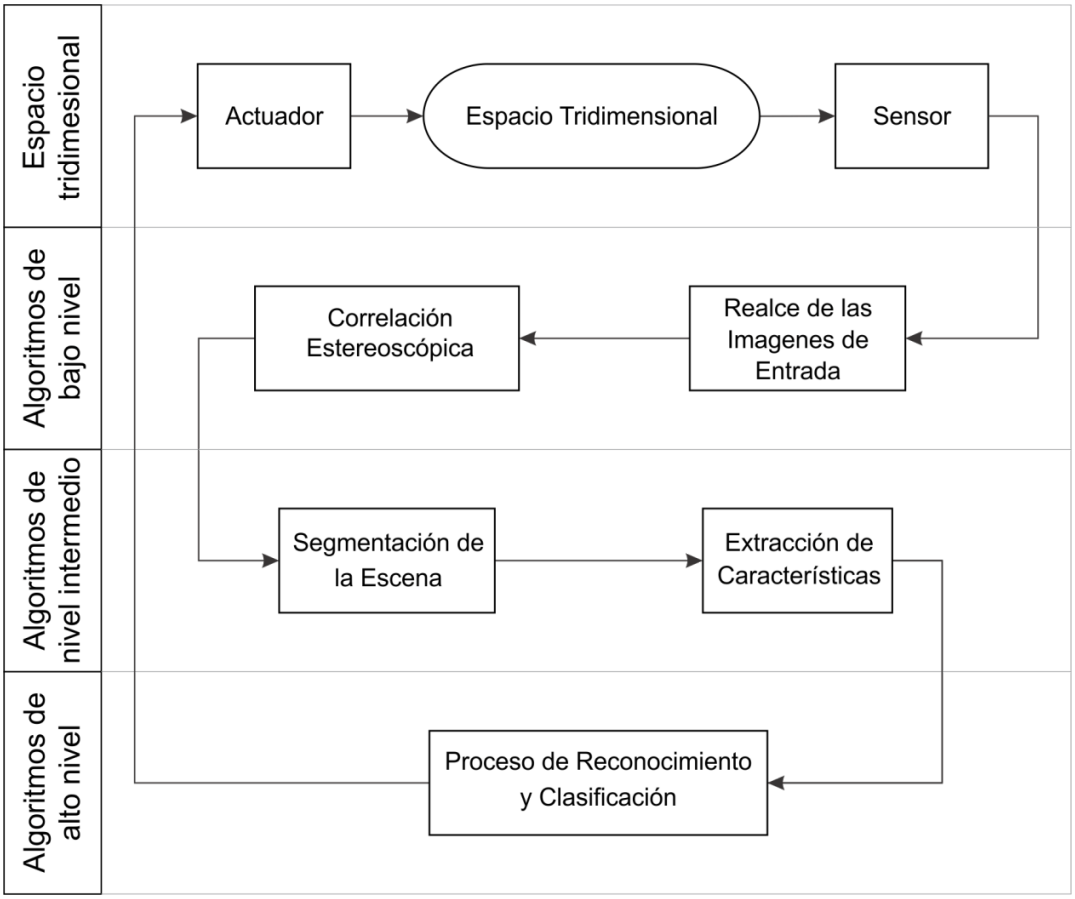


***Figura 4.1**: Esquema de un sistema de reconocimiento de objetos.*

Como se ha mencionado anteriormente, los algoritmos implementados han sido clasificados en tres niveles que se describen de la siguiente forma:

- Algoritmos de bajo nivel. Trabajan directamente con los pixeles de las imágenes de entrada para extraer propiedades como los bordes, gradiente, profundidad de la escena, textura, color, etc.

- Algoritmos de nivel intermedio. Consiste generalmente en utilizar los resultados de los algoritmos de bajo nivel para reducir la cantidad de

información y obtener datos más finos como características de formas, regiones específicas de la escena, etc.

- Algoritmos de alto nivel. Esta generalmente orientada al proceso de interpretación de los datos obtenidos en los niveles inferiores y se utilizan modelos y/o conocimiento a priori del dominio. Esta etapa es normalmente realizada por máquinas de aprendizaje.

Aunque estos niveles son aparentemente secuenciales, esto no es necesario, y se consideran interacciones entre los diferentes niveles incluyendo retroalimentación de los niveles altos a los inferiores [16].

4.2. Configuraciones y procesos iniciales

El software desarrollado para el funcionamiento del sistema de reconocimiento ha sido implementado en el entorno de desarrollo MATLAB®, ya que éste tiene herramientas integradas que pueden ser usadas para un desarrollo modular, flexible y más rápido de diversos sistemas en ingeniería. Para una mayor eficiencia en los tiempos de ejecución de los algoritmos se puede usar otros lenguajes de programación como Java o C++ en la implementación de cada uno de los algoritmos. La unidad central de procesamiento que ejecuta los algoritmos desarrollados en esta tesis es un procesador Intel® Core™ i7-720QM que tiene una frecuencia de reloj de 1.6 GHz.

Los procesos iniciales del sistema realizan la configuración de las cámaras, asignan valores iniciales a los diferentes parámetros de control de los algoritmos que se describen en las secciones siguientes, además de crear la interfaz de usuario del sistema de reconocimiento.

4.2.1. Sistema estereoscópico para la captura de la escena

Para realizar la captura de la escena se ha utilizado un sistema estereoscópico compuesto por dos cámaras LifeCam Studio, (véase Figura 4.2), estas cámaras han sido seleccionadas para esta tesis debido a su interacción exitosa con MATLAB®. El par estereoscópico capturado tiene una resolución de 160 x 120 pixeles para la imagen izquierda y derecha, además de utilizar el espacio de color RGB en ambas imágenes.

El sistema de captación binocular es ubicado en una superficie horizontal con la geometría de ejes paralelos descrita en la Sección 3.3, con los ejes ópticos tan cercanos como sea posible, obteniendo la mínima longitud para la línea base que la geometría de las cámaras permite, lo que constituye una longitud aproximada de 4 cm.

***Figura 4.2**: Sistema binocular compuesto por dos cámaras LifeCam Studio.*

A ésta configuración del sistema estereoscópico se realiza una calibración manual antes de llevar a cabo la captura de la escena, lo cual permite abordar el problema de correspondencia independientemente de la

etapa de rectificación de imágenes. El sistema de reconocimiento no requiere información precisa acerca de la profundidad de los objetos, solo una relación cualitativa de la profundidad de los objetos en la escena, por lo que la configuración de ejes paralelos ajustados manualmente proveen resultados suficientemente correctos para las siguientes etapas de procesamiento del sistema (para una mejor performance en el cálculo del mapa de disparidad, es necesario realizar la rectificación de las imágenes).

4.3. Desarrollo de algoritmos de bajo nivel

Dentro de los algoritmos de bajo nivel, el más complejo de ellos es el algoritmo de correlación que permite encontrar los puntos equivalentes en el par estereoscópico, con lo cual se puede obtener el mapa de disparidad. En esta parte también se describen los algoritmos de realce, los cuales son importantes para una adecuada segmentación.

4.3.1. Algoritmos para el realce de las imágenes de entrada

Las imágenes izquierda y derecha capturadas por el sistema estereoscópico son procesadas individualmente antes de realizar la correlación. El objetivo en esta etapa del procesamiento es realzar características deseadas como los bordes de los objetos, y eliminar las no deseadas tales como el ruido, este procesamiento se ha realizado en el dominio espacial.

Para llevar a cabo esta tarea primero se ha transformado el espacio de color del par estereoscópico de RGB a escala de grises, luego se ha aplicado un filtro de suavizado con máscara de 3x3, en este caso el filtro mediana que es un filtro no lineal, el cual reemplaza el valor de un pixel por

la mediana de los niveles de gris de los pixeles vecinos. Este filtro es aplicado con el fin de eliminar el ruido de las imágenes, pero su desventaja es que tiende a eliminar el borde de los objetos.

Para realzar los bordes en los objetos se ha implementado un filtro de nitidez que hace uso del filtrado Laplaciano de la siguiente forma:

$$f_s(x,y) = f(x,y) - C\nabla^2 f(x,y), \tag{4.1}$$

donde C es una constante de escalado, además:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{4.2}$$

De la ecuación (4.1) se observa que el filtro de nitidez es calculado mediante la diferencia entre la imagen original y el resultado del filtro Laplaciano aplicado a la imagen escalada C veces. El efecto de este filtro en una dimensión es mostrado en la Figura 4.3.

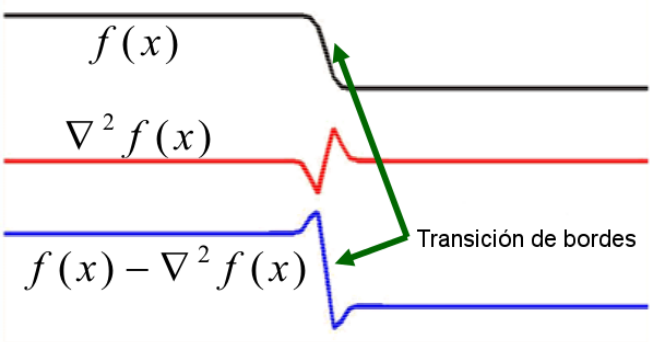


***Figura 4.3**: Efecto del filtro Laplaciano en una dimensión.*

El valor del parámetro C utilizado para el filtro de nitidez es de 0.3, este número fue determinado mediante prueba y error, tomando como referencia el impacto que tiene en el cálculo del mapa de disparidad. La máscara utilizada en este filtro es la que se muestra en la Figura 4.4.

0	0	0
0	1	0
0	0	0

- 0.3

0	1	0
1	-4	1
0	1	0

=

0	-0.3	0
-0.3	2.2	-0.3
0	-0.3	0

***Figura 4.4**: Mascara utilizada para el filtro de nitidez.*

El diagrama de flujo para los algoritmos de realce que se han aplicado a las imágenes de entrada es mostrado en la Figura 4.5.

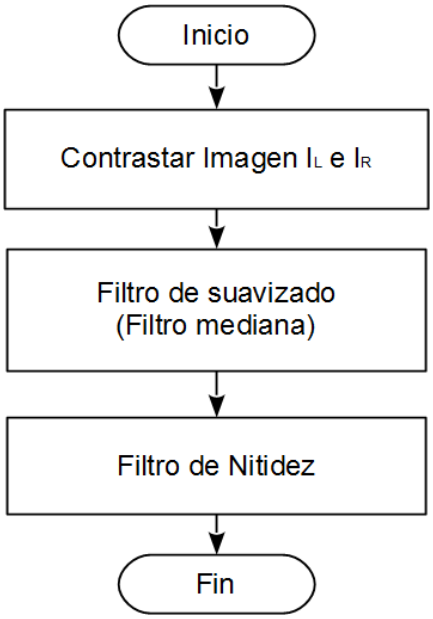


***Figura 4.5**: Diagrama de flujo para realce de imágenes.*

4.3.2. Algoritmos para la correlación estereoscópica

Los algoritmos para la correlación estereoscópica tienen como objetivo generar el mapa de disparidad el cual es inversamente proporcional al mapa de profundidad de la escena. El cálculo del mapa de disparidad es importante debido a que es utilizado para segmentar la escena que se está analizando, en donde los objetos de interés serán aquellos que se encuentran a menor distancia del sistema de reconocimiento. Estos

algoritmos, que constituyen el proceso principal de la visión estereoscópica, brindan al sistema de reconocimiento la posibilidad de abordar una mayor cantidad de problemas de reconocimiento de objetos en diferentes entornos debido a que solo requieren que el fondo de la escena presente textura, como normalmente ocurre en el espacio tridimensional real, a diferencia de otras técnicas que suelen utilizar un fondo de color uniforme en la escena para poder realizar el reconocimiento de los objetos de interés.

Los algoritmos para la correlación tienen como entrada dos imágenes en escala de grises que han sido procesadas con algoritmos de realce y que constituyen el par estereoscópico. La salida de los algoritmos de correlación será el mapa de disparidad denso para la imagen derecha. En esta tesis se ha utilizado un método global, programación dinámica, como el algoritmo de correspondencia principal para el sistema debido a las ventajas que presenta respecto a métodos locales como se describió en el Capítulo 3. Además del algoritmo principal, también ha sido implementado un método local, correlación basada en ventanas, con el propósito de poder comparar la performance de los métodos locales frente a los métodos globales.

Algoritmo de correspondencia principal

En la primera etapa del algoritmo se genera el espacio de disparidad para una scanline, para esto se ha considerado la segunda forma propuesta (descrito en la Sección 3.4.2), en donde uno de los ejes es la scanline izquierda y el otro el rango de disparidad. En la Figura 4.6 está el diagrama de flujo simplificado para el algoritmo de correspondencia principal.

El costo de correspondencia utilizado para el espacio de disparidad es la suma de diferencias absolutas (SAD) con ventanas de 7x7, la elección de este costo es debido a la menor carga computacional que agrega al algoritmo.

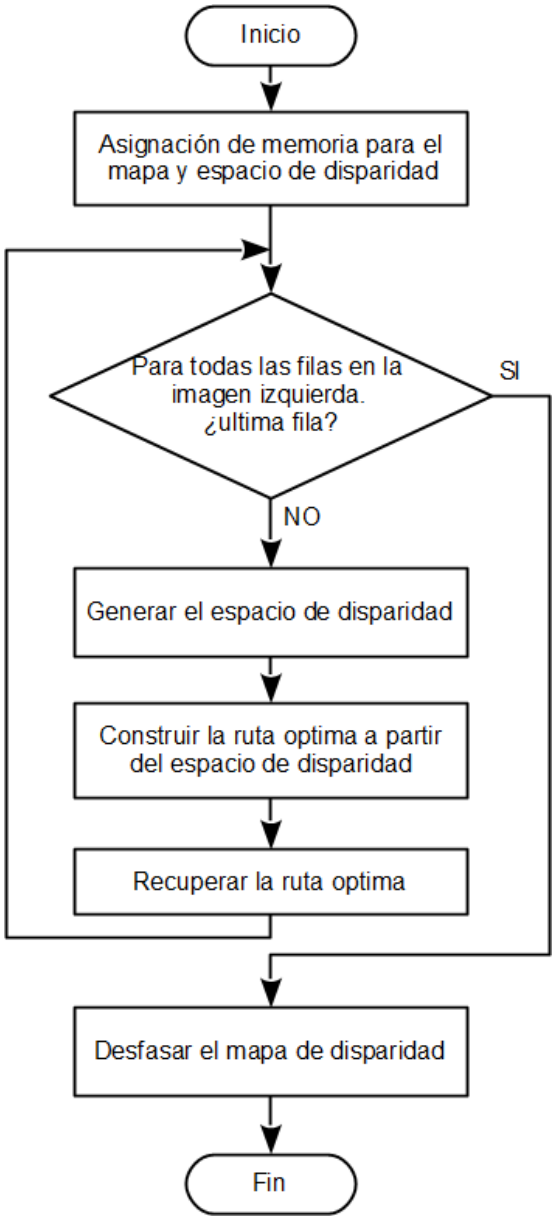


***Figura 4.6**: Diagrama de flujo simplificado para la correlación estereoscópica usando programación dinámica.*

Se ha reducido la búsqueda del punto correspondiente en toda la scanline derecha a un segmento de esta línea mediante la restricción de disparidad, en donde se ha considerado diez niveles para el rango de la disparidad. Estas consideraciones han sido muy importantes para la construcción eficiente del espacio de disparidad que representa la energía de datos en la ecuación (3.13).

En la segunda etapa del algoritmo se realiza el cálculo de la energía de suavizado, el cual es considerado un costo adicional que se calculará a partir de los vecinos del punto analizado a lo largo de la scanline. En este cálculo se ha considerado a tres puntos vecinos por la izquierda y derecha, aplicando una penalidad conforme la distancia va incrementándose respecto al punto de interés. La Figura 4.7 muestra parte de este proceso en forma gráfica.

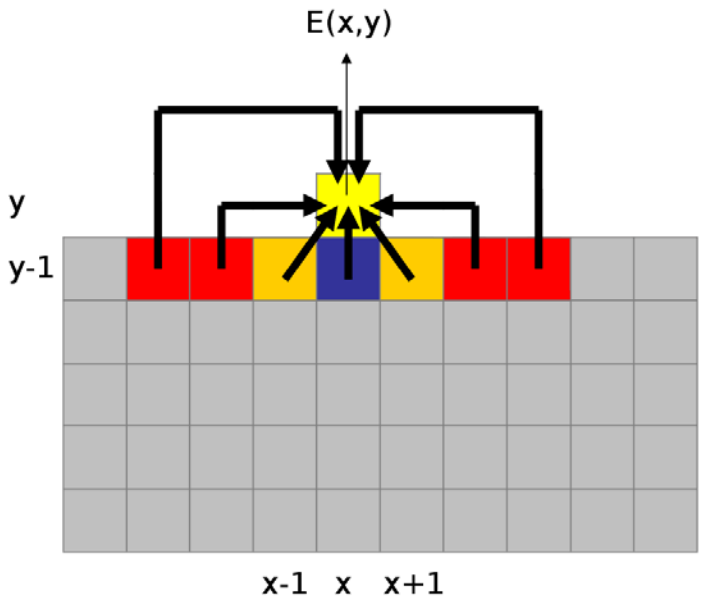


***Figura 4.7**: Calculo de la energía de suavizado considerando los tres puntos más cercanos al punto de interés.*

La energía del punto (x, y) en el espacio de disparidad es calculada mediante la expresión mostrada en la ecuación (4.3), la cual se realiza para todos los puntos en el espacio de disparidad.

$$E(x,y) = C(x,y) + min\begin{Bmatrix} C(x-3,y-1)+3P \\ C(x-2,y-1)+2P \\ C(x-1,y-1)+P \\ C(x,y-1) \\ C(x+1,y-1)+P \\ C(x+2,y-1)+2P \\ C(x+3,y-1)+3P \end{Bmatrix}. \quad (4.3)$$

En la última ecuación, el término $C(x, y)$ representa la energía de datos en el punto de análisis, este valor es calculado mediante suma de diferencias absolutas como se menciono anteriormente. El segundo término del lado derecho en la ecuación representa la energía de suavizado, en cuyo cálculo se introduce una penalidad simbolizada por la letra P y de valor 0.7 en el algoritmo de correspondencia implementado.

La última etapa del algoritmo encuentra una secuencia de disparidades óptimo en la scanline mediante la minimización de la energía total dada por diferentes conjuntos de disparidades. Esto es equivalente al de encontrar un ruta óptima en el espacio de disparidad desde un lado al lado opuesto. Para resolver este problema se ha utilizado programación dinámica por ser una técnica eficiente en la reducción del tiempo de ejecución del algoritmo. La programación dinámica es adecuada en nuestro problema porque podemos dividir el problema original en sub-problemas más pequeños que son resueltos de manera óptima para construir la solución óptima al problema original. Las soluciones óptimas de los sub-problemas son guardadas en una matriz de igual dimensión al espacio de

disparidad que posteriormente son usadas para construir la secuencia óptima de disparidades para la scanline.

Estas tres etapas son realizadas para todas las filas de la imagen izquierda a fin de obtener el mapa de disparidad denso de la escena. El diagrama de flujo completo del algoritmo de correspondencia se encuentra en la Sección A.2 de los anexos.

Correlación basada en ventanas

La solución del problema de correspondencia mediante un algoritmo de correlación basada en ventanas presenta una complejidad menor respecto a la utilización de programación dinámica que es descrito en la sección anterior. El diagrama de flujo para la correlación basada en ventanas está en la Figura 4.8.

En la primera parte del algoritmo se realiza el cálculo de los límites horizontales y verticales para la ventana que será utilizada en el cálculo de los costos de correspondencia, esta ventana se encuentra centrada en el pixel que está siendo analizando y tienen una dimensión de 7x7. Luego de calcular los límites, se extrae la ventana de la imagen derecha para la búsqueda de su correspondiente en la imagen izquierda. En este algoritmo también ha sido utilizada la restricción de disparidad para construir una región de interés en la imagen izquierda, de esta forma se limita la búsqueda del pixel correspondiente a un segmento limitado en una línea epipolar de la imagen izquierda. El costo de correspondencia utilizado es la suma de diferencias al cuadrado (SSD).

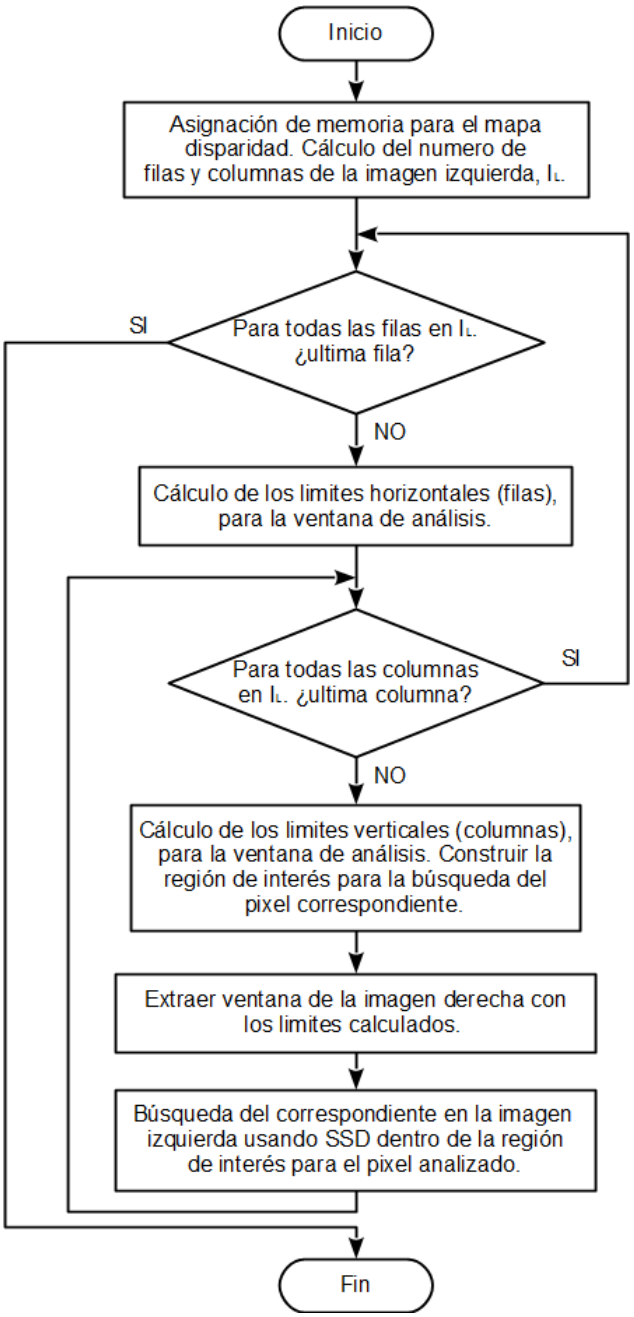


***Figura 4.8**: Diagrama de flujo para la correlación basada en ventanas.*

En la segunda parte del algoritmo se ejecuta la correspondencia propiamente dicha, en donde se inicia la búsqueda de la ventana que minimiza el costo de correspondencia dentro de la región de interés. Este proceso es realizado en todos los pixeles de la imagen para obtener el mapa de disparidad denso requerido.

4.4. Desarrollo de algoritmos de nivel intermedio para la segmentación y extracción de características de objetos

Existen diferentes algoritmos para la segmentación de imágenes dentro de los que se puede mencionar a la segmentación basada en bordes y segmentación basada en regiones. Esta sección del capítulo presenta y desarrolla los algoritmos usados para la segmentación y extracción de características en esta tesis.

4.4.1. Algoritmos para la segmentación del mapa de disparidad

Estos algoritmos tienen como objetivo realizar una segmentación sobre el mapa de disparidad para obtener el objeto de menor distancia al sistema de reconocimiento. Las entradas para este algoritmo son el mapa de disparidad que se ha calculado con los algoritmos de bajo nivel y la imagen derecha del par estereoscópico. La salida está compuesta por una imagen que contiene solo un objeto de la escena y que es el de menor distancia al sistema. La segmentación del mapa de disparidad es muy importante dentro del sistema de reconocimiento debido a que sobre sus resultados se realiza todo el procesado posterior para el reconocimiento del objeto. El diagrama de flujo para este algoritmo está en la Figura 4.9.

El mapa de disparidad entregado por los algoritmos de bajo nivel está compuesto de diez niveles y puede ser representado como una imagen en escala de grises. De acuerdo a la ecuación (3.3), la profundidad de los objetos es inversamente proporcional a la disparidad que tiene estos. Por lo tanto, siendo el objetivo del algoritmo obtener el objeto con menor profundidad, debemos seleccionar el nivel con la máxima disparidad. Estos niveles son representados en los resultados mediante un gradiente de

colores desde el rojo al azul para los niveles de menor y mayor profundidad respectivamente.

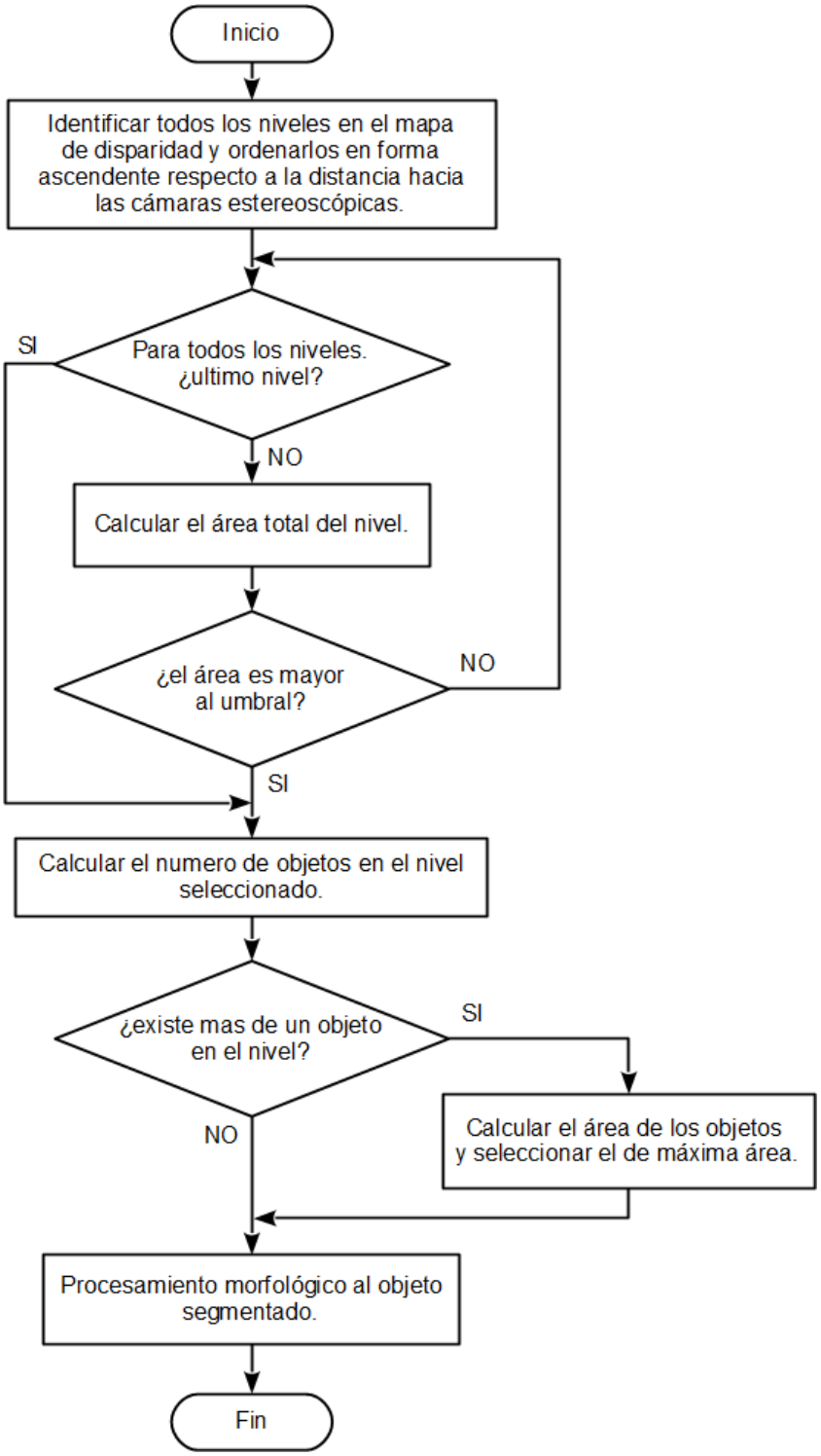


***Figura 4.9**: Diagrama de flujo para la segmentación del mapa de disparidad.*

El mapa de disparidad está sujeto a presentar ruido o distorsiones generadas por oclusiones u otros fenómenos (véase Sección 3.5.3), debido a estos problemas se ha establecido un umbral que establece que el área del nivel de menor distancia debe tener más del 4% de los pixeles de la imagen. De esta forma se eliminan falsos niveles que aparentemente se encuentran a menor distancia de las cámaras. Luego se procede a calcular el número de objetos presentes en el nivel seleccionado, y si existe más de un objeto solo se conserva el que tiene mayor área.

La última parte del algoritmo se dedica a realizar el procesamiento morfológico a la imagen binaria del objeto, en este algoritmo se ha realizado una operación de cierre seguido de una dilatación (véase Sección A.3 de los anexos). La operación de cierre elimina pequeños huecos y une componentes conexos cercanos, esta operación está dada por la ecuación (4.4), en donde la imagen I primero es dilatada por el elemento estructural SE seguido de una erosión del resultado con SE.

$$I \bullet SE = (I \oplus SE) \ominus SE. \tag{4.4}$$

La dilatación realizada al resultado de la operación de cierre tiene como objetivo incrementar la frontera del objeto porque parte de ésta podría no estar incluida en la imagen debido a una incorrecta correlación de los pixeles en la frontera del objeto, de esta forma se asegura que el objeto este completamente contenido en la imagen entregada por el algoritmo. La Figura 4.10 muestra el elemento estructural utilizado en todas las operaciones morfológicas de este algoritmo.

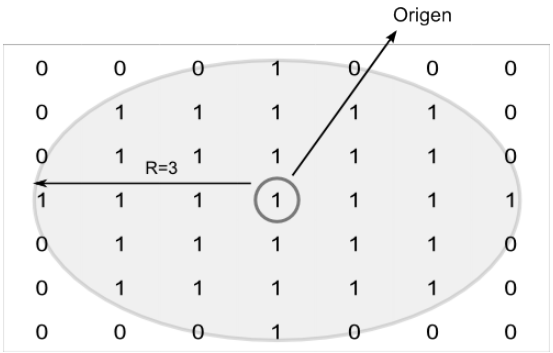


***Figura 4.10**: Elemento estructural para el procesamiento morfológico.*

Segmentación complementaria para objetos de color uniforme

En el análisis de objetos que tienen un color uniforme, se puede realizar una segmentación con mayor precisión teniendo esta información a priori. Para este propósito se ha implementado un algoritmo que realiza la segmentación de acuerdo al color de la imagen, este algoritmo usa el método de división y fusión (Split & Merge), que se basa precisamente en la división y fusión de regiones como su nombre lo indica. El diagrama de flujo para el algoritmo implementado está en las Figuras 4.11 y 4.12. La entrada para el algoritmo es el resultado del algoritmo de segmentación del mapa de disparidad y la salida es una imagen binaria que representa al objeto.

La imagen inicial para el algoritmo de división y fusión debe ser cuadrada, por lo tanto, si la imagen de entrada no tiene estas características, se realiza una previa transformación para obtener una imagen cuadrada sin perder la relación de aspecto de la imagen original. Antes de ejecutar los procesos de división y fusión, se definen los criterios para la similitud entre regiones y el criterio para establecer si una región es uniforme. El criterio establecido en esta tesis para la similitud entre regiones será la diferencia del promedio de los pixeles de cada región, y se considera

que dos regiones tienen un mismo color uniforme si esta diferencia no supera un determinado umbral. El criterio para una región uniforme es que la diferencia entre el máximo y mínimo valor de un pixel en la región no supere un umbral que tendrá el mismo valor del criterio de similitud de regiones.

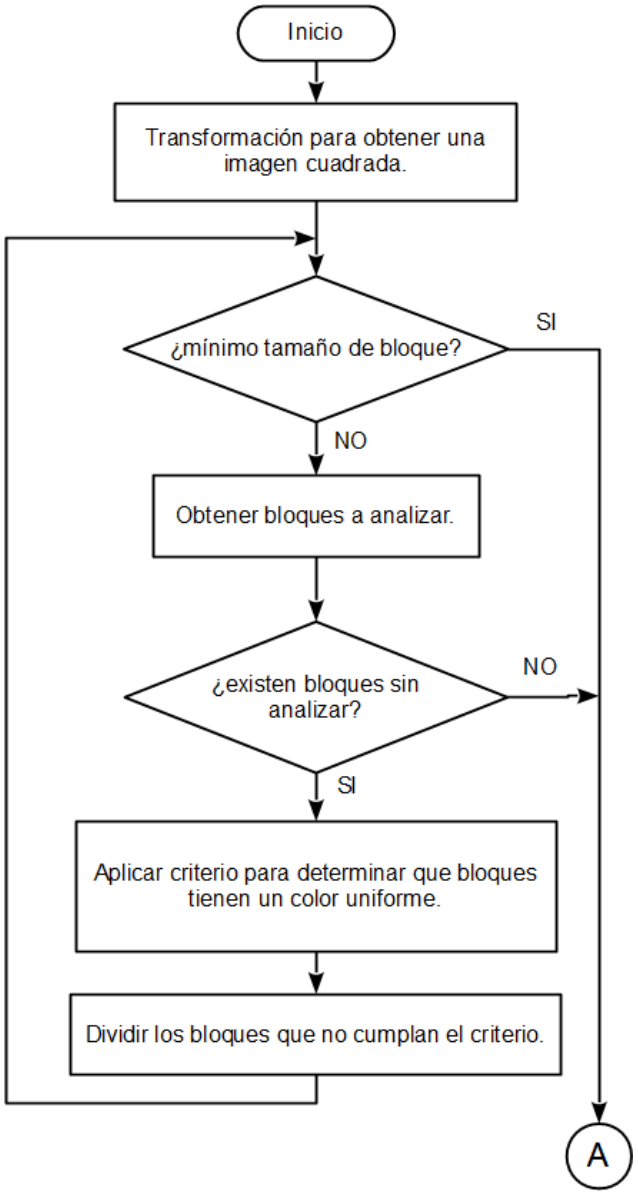


***Figura 4.11**: Diagrama de flujo para el proceso de división.*

En la primera fase del algoritmo se realiza el proceso de división mediante la aplicación del criterio para regiones uniformes en toda la imagen

inicialmente. Si el criterio no se cumple, la imagen inicial se divide en cuatro sub-imágenes de igual tamaño (cuadrantes), luego se vuelve a aplicar el criterio para cada una de las cuatro sub-imágenes. Si para alguna de las sub-imágenes el criterio no se cumple nuevamente, esta se vuelve a dividir en cuatro partes iguales, y así sucesivamente hasta que se alcance un mínimo tamaño para la región.

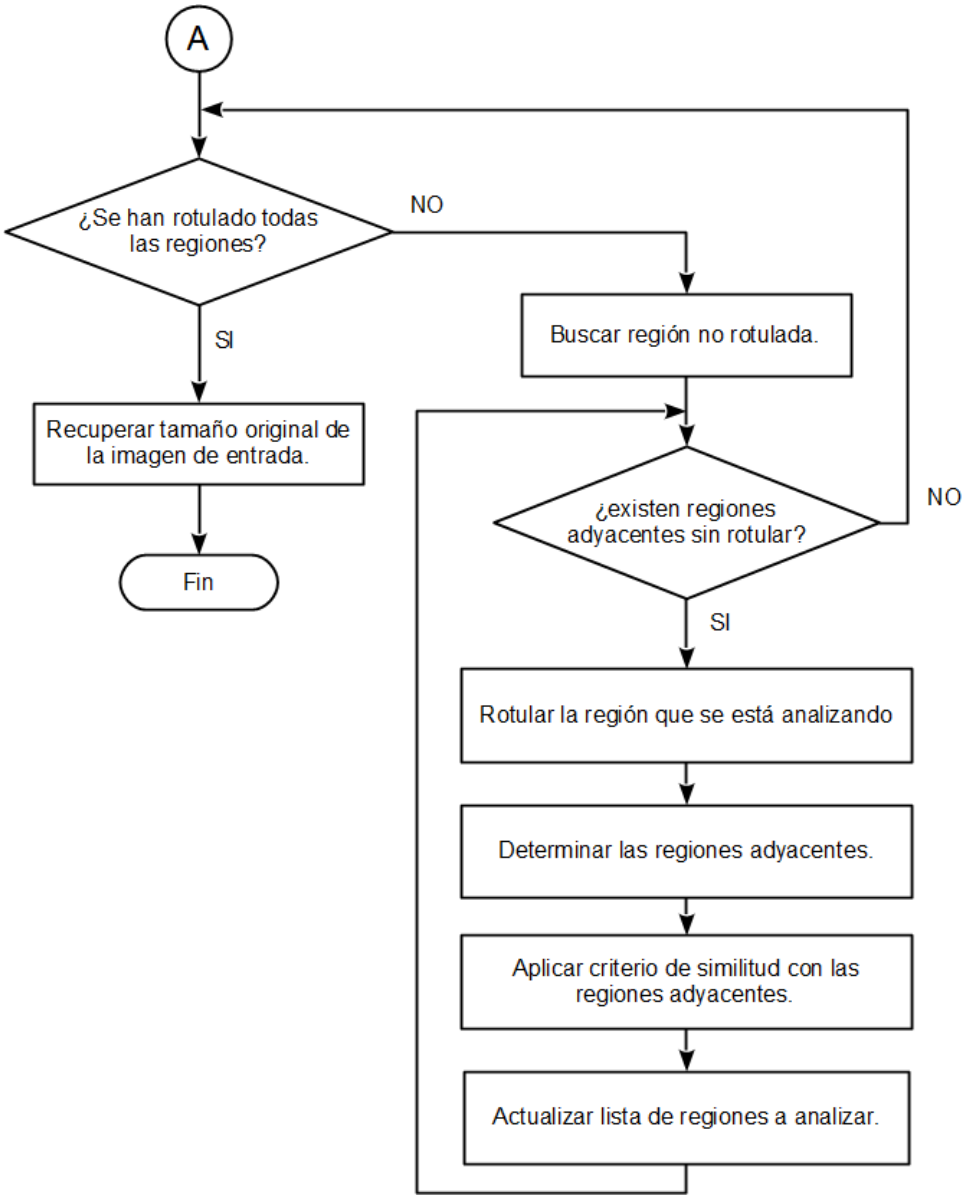


***Figura 4.12**: Diagrama de flujo para el proceso de fusión.*

La segunda fase del algoritmo consiste en fusionar las regiones adyacentes que cumplan el criterio de similitud, esto se realiza mediante la asignación de un rotulo a cada región adyacente que cumpla el criterio y que tiene el mismo valor del rotulo de la región que se está analizando. El procedimiento se itera hasta que no queden zonas adyacentes con similares características que sigan separadas. La Figura 4.13 muestra regiones en una imagen que dan una idea del proceso de división y fusión

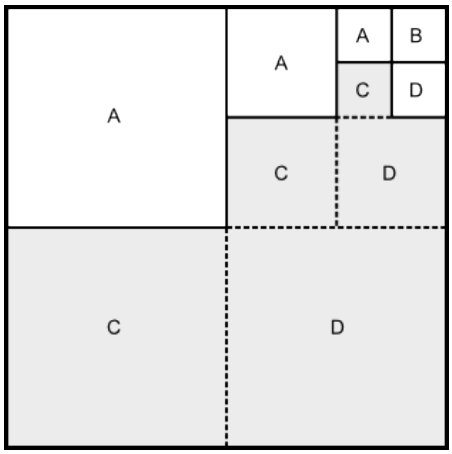


***Figura 4.13**: Representación de las regiones (bloques), en la imagen.*

Finalmente, con el fin de obtener una imagen binaria se realiza el etiquetado de todas las regiones uniformes resultantes del algoritmo de división y fusión con el objetivo de calcular el área de cada una de ellas y seleccionar el de máxima área para el análisis posterior, para ello se considera que el objeto de color uniforme tiene el mayor porcentaje de pixeles en la imagen de entrada al algoritmo. En el caso de tener información a priori acerca del objeto que se está analizando, se puede realizar algún proceso complementario para obtener resultados más precisos.

4.4.2. Algoritmos para la extracción de características

El objetivo de la extracción de características es la reducción de la dimensión del espacio de entrada de los algoritmos de alto nivel, que se realiza mediante el cálculo de algunas medidas numéricas e información relevante de los datos de entrada. Las imágenes están normalmente compuestas de una gran cantidad de pixeles y el costo computacional asociado para realizar la tarea de reconocimiento es muy elevado, por lo tanto, se necesita un vector de características que describe al objeto contenido en la imagen. Estas características deben ser independientes del escalado, rotación o traslación de los objetos considerados y normalmente se basan en la forma geométrica, color, textura, entre otros [17]. El conjunto de características utilizadas en esta tesis que son calculadas en forma secuencial por los algoritmos de extracción de características, se han dividido en tres grupos que se describen a continuación.

Descriptores básicos

Los descriptores básicos miden un conjunto de propiedades a partir de la imagen binaria del objeto y se basan en la geometría de éste. Se han considerado cinco características básicas que se calculan de la siguiente forma.

- Excentricidad, escalar que especifica la excentricidad de la elipse que tiene los mismos segundos momentos como la región del objeto. La excentricidad es la relación de la distancia entre los focos de la elipse y la longitud de su eje principal.

- Solidez, escalar que especifica la proporción entre el área del objeto y el área del menor polígono convexo que contiene el objeto.

- Extensión, esta característica es similar a la solidez y se calcula como el ratio entre los pixeles del objeto a los pixeles del menor rectángulo que contiene al objeto.

- Número de Euler, esta característica es un descriptor topológico que se calcula de forma general como el número de objetos menos el número de agujeros. En caso de que solo exista un objeto, el número de Euler se calcula como: $E = 1 - H$, donde H representa el número de agujeros.

- Compacidad, es un número adimensional que nos permite medir que tan compacto es el objeto, su valor es mínimo para objetos con geometría circular. La definición de la compacidad se encuentra en la ecuación (4.5).

$$c = p^2/A, \tag{4.5}$$

  donde p es el perímetro y A es el área del objeto.

Momentos invariantes

Los momentos invariantes deben su nombre a que estos no cambian ante el escalado, la traslación o rotación del objeto a diferencia de los momentos generales. Los momentos bidimensionales para una imagen digital, muestreada *M x M* que tiene una función $f(x, y)$, se definen como:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} x^p y^q f(x,y), \quad p,q = 0,1,2,3 \ldots \tag{4.6}$$

Si existe una traslación del objeto por una cantidad (a,b), los momentos de $f(x,y)$ son definidos como:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x+a)^p (y+b)^q f(x,y). \tag{4.7}$$

Por lo tanto, los momentos centrales μ_{pq} pueden ser calculados utilizando la ecuación (4.7) mediante la sustitución $a = -\overline{x}$ y $b = -\overline{y}$ como lo muestran las ecuaciones (4.8):

$$\overline{x} = \frac{m_{10}}{m_{00}}, \quad y \quad \overline{y} = \frac{m_{01}}{m_{00}},$$
$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x+\overline{x})^p (y+\overline{y})^q f(x,y). \tag{4.8}$$

Cuando se aplica una normalización al escalado, los momentos centrales cambian de la siguiente forma:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}{}^{\gamma}}, \ \ donde: \ \ \gamma = \frac{p+q}{2} + 1. \tag{4.9}$$

De los momentos de segundo y tercer orden se define siete momentos invariantes a cambios de escala, posición y rotación del objeto [18], estos siete momentos forman parte del conjunto de características calculadas por el algoritmo. En términos de los momentos centrales, los siete momentos están dados por:

$$
\begin{aligned}
M_1 &= (\eta_{20} + \eta_{02}),\\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2,\\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2,\\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,\\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]\\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],\\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),\\
M_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]\\
&\quad - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}
\tag{4.10}
$$

Descriptores HOG

Los descriptores HOG (del inglés *Histogram of Oriented Gradients*), han sido utilizados cuando se analizan objetos compuestos de varios colores, estos descriptores proporcionan información de los cambios debido a los bordes de los objetos contenidos en la imagen y difiere de otros métodos en que el cálculo se realiza sobre una malla densa uniformemente espaciada, esta razón es por la que se ha decidido utilizar estos descriptores en esta tesis.

El algoritmo que obtiene los descriptores HOG tiene como entrada la imagen segmentada obtenida a partir del mapa de disparidad. Sobre esta imagen se calculan los componentes horizontal G_x y vertical G_y del gradiente, usando estos componentes se calcula la magnitud y el ángulo del gradiente sobre cada pixel (m, n) de la imagen de acuerdo a las ecuaciones (4.11).

$$|G(m,n)| = \sqrt{G_x(m,n)^2 + G_y(m,n)^2},$$
$$\measuredangle G(m,n) = \tan^{-1}\left(\frac{G_y(m,n)}{G_x(m,n)}\right). \quad (4.11)$$

Para obtener los descriptores se procede a dividir la imagen en bloques uniformes no traslapados, para esta tesis se han utilizado nueve bloques de idéntica dimensión. Luego se selecciona cada bloque añadiendo cierta área a su contorno para que exista un solape con los bloques adyacentes, sobre cada uno de los bloques se realiza una sumatoria de las magnitudes de acuerdo a su ángulo, para tal fin el intervalo de ángulos de -180 a 180 grados es dividido en 9 segmentos distribuidos uniformemente. Por lo tanto se obtendrán un total de 9 descriptores por cada bloque en la imagen. Finalmente se realiza una normalización de los resultados para obtener unos descriptores más robustos [19]. El vector final que contiene los descriptores se formará por la concatenación de los descriptores pertenecientes a cada bloque de la imagen, este vector tendrá 81 descriptores porque se utilizan nueve bloques para la imagen de entrada. Para mayor referencia sobre los descriptores HOG, se puede revisar [20].

4.5. Desarrollo de algoritmos de alto nivel para el reconocimiento y clasificación de objetos

El desarrollo de algoritmos de alto nivel que comprenden técnicas de inteligencia artificial son descritos en esta sección, esto incluye la arquitectura de la máquina de aprendizaje, la etapa de entrenamiento y el reconocimiento de nuevos objetos. A lo largo del desarrollo de estos algoritmos se ha tomado como base la librería de objetos spider [21] en MATLAB®, la cual es software libre y puede ser modificado bajo los términos de la GNU General Public License publicado por la

Free Software Foundation. Esta librería contiene una gran cantidad de algoritmos que tienen como fundamento a los SVM.

4.5.1. Arquitectura de SVM

La arquitectura de la máquina de aprendizaje tiene una estructura flexible ya que la cantidad de kernels en la capa oculta varía de acuerdo a la cantidad de vectores de soporte que tiene el SVM, estos vectores de soporte dependen de la etapa de entrenamiento, que en este trabajo se ha realizado para diferentes objetos.

El clasificador utilizado para el reconocimiento de los objetos será un clasificador no lineal con margen suave debido a que esta estructura es la más general y robusta frente a una mayor variedad de problemas y en particular frente a una mayor variedad en el reconocimiento de diferentes objetos.

La arquitectura presentada en esta sección corresponde al entrenamiento del SVM para un conjunto de objetos de forma esférica y que tienen un color uniforme. La arquitectura usada para el reconocimiento de estos objetos tiene como entradas a los descriptores básicos y a los momentos invariantes que constituyen 12 características que son parte de las salidas de los algoritmos de segundo nivel, y tiene como salida un valor booleano que indica si el objeto presentado al sistema corresponde a la misma clase de los objetos ya entrenados (véase Figura 4.14). El kernel de la capa oculta ha sido seleccionado mediante prueba y error utilizando los dos kernels descritos en la Sección 2.4.4, la ecuación (4.12) muestra el kernel seleccionado para el reconocimiento de los objetos esféricos.

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = (\boldsymbol{x}^T \boldsymbol{x_i} + 1)^7. \quad (4.12)$$

Este kernel es calculado seis veces para cada patrón de entrada debido a que en el reconocimiento de los objetos esféricos se han utilizado seis vectores de soporte.

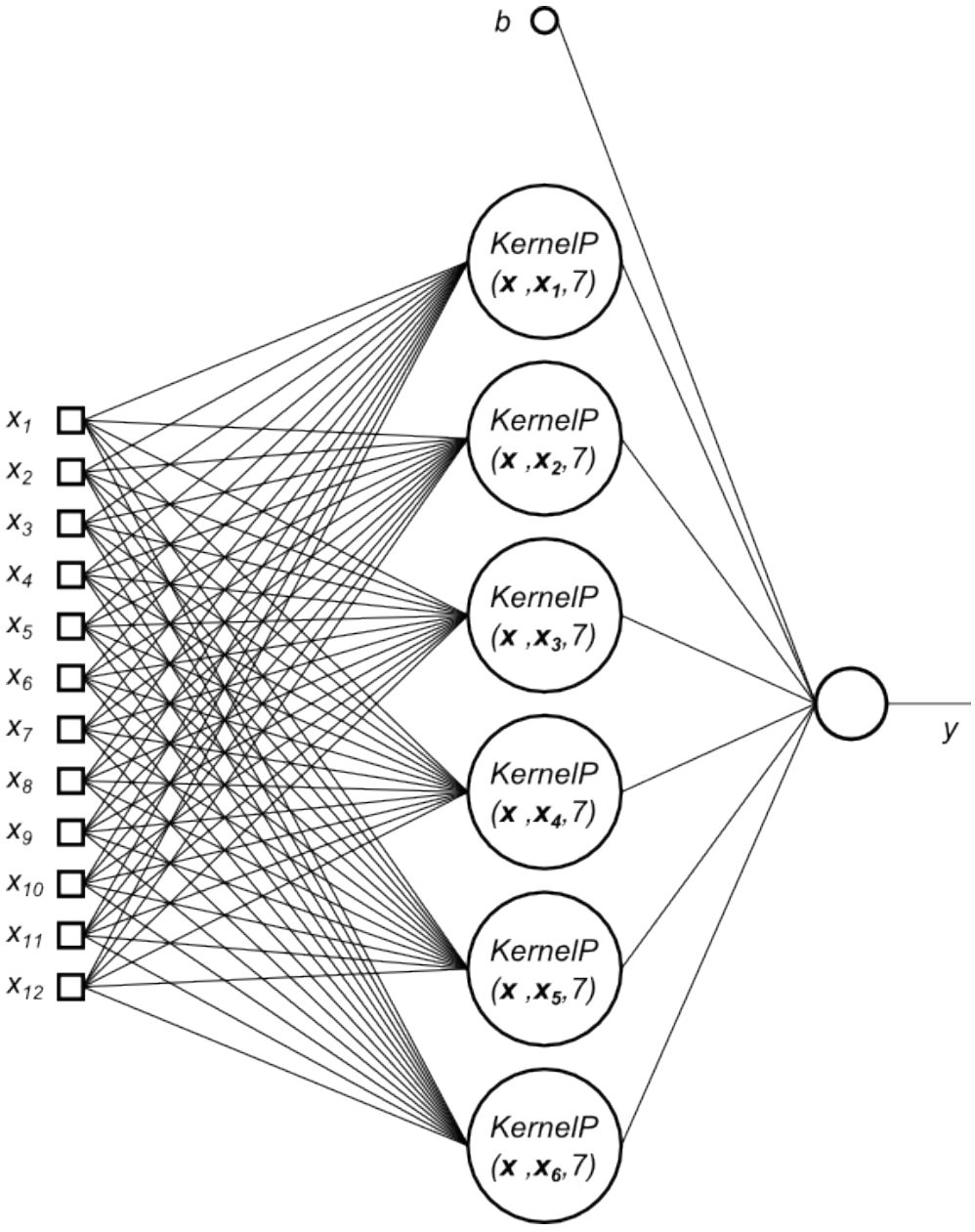


***Figura 4.14**: SVM para el reconocimiento de objetos esféricos de color uniforme.*

La implementación de los kernels así como la propia máquina de aprendizaje ha sido realizada mediante programación orientada a objetos, por lo que se han definido clases para el SVM y los kernels. Los atributos de la clase de SVM representan los diferentes parámetros de la máquina de aprendizaje, por lo que su configuración es sencilla y la implementación de toda la estructura es modular gracias a las ventajas brindadas por la programación orientada a objetos.

4.5.2. Etapa de entrenamiento

La etapa de entrenamiento consiste en resolver un problema de optimización cuadrático con restricciones lineales (véase Sección 2.4.2), el cual puede ser resuelto mediante diferentes algoritmos, dentro de los que destaca la Optimización Mínima Secuencial (SMO), este algoritmo es conveniente cuando se tienen grandes conjuntos de entrenamiento ya que reduce de forma importante los recursos consumidos por el computador. Para el entrenamiento de los objetos esféricos, mencionados en la sección anterior, se ha hecho uso de la programación cuadrática, ya que el conjunto de entrenamiento no es muy grande y el número de características tampoco es muy elevado. Se plantea el problema de optimización de la siguiente forma:

Dado el conjunto de entrenamiento, maximizar la siguiente función objetivo:

$$L_D = \sum_{i=1}^{50} \alpha_i - \frac{1}{2} \sum_{i=1}^{50} \sum_{j=1}^{50} \alpha_i \alpha_j t_i t_j (\boldsymbol{x}_i^T \boldsymbol{x_j} + 1)^7, \tag{4.13}$$

sujeto a las restricciones:

$$\sum_{i=1}^{50} \alpha_i \, t_i = 0, \tag{4.14}$$

$$0 \leq \alpha_i \leq 10, \quad para \ \ i = 1,2,\ldots,50.$$

El conjunto de entrenamiento es compuesto de 50 imágenes de 100 pixeles de ancho y alto, de los cuales 20 de ellas corresponden a ejemplos positivos, y el resto que es compuesto de objetos de diferente clase son los ejemplos negativos. Parte del conjunto de entrenamiento es mostrado en las Figuras 4.15 y 4.16.

***Figura 4.15**: Ejemplos positivos para el entrenamiento de SVM.*

***Figura 4.16**: Ejemplos negativos para el entrenamiento de SVM.*

El conjunto de entrenamiento tiene que ser procesado por el algoritmo de extracción de características para reducir la dimensión del

espacio de entrada, de lo contrario, el costo computacional del algoritmo de entrenamiento sería muy elevado. El valor del parámetro C que es la cota superior para los multiplicadores de Lagrange en el problema de optimización tiene el valor de 10 (véase ecuación (4.14)). Este valor ha sido escogido de tal forma que la máquina de aprendizaje no pierda capacidad de generalización

4.5.3. Reconocimiento de nuevos objetos

La forma en la que se realiza el reconocimiento de nuevos objetos que no han sido presentados en la etapa de entrenamiento se describe en esta sección. Esta etapa permite evaluar la performance de la máquina de aprendizaje y en el caso de los objetos esféricos se ha logrado un buen desempeño del sistema. Como se describió en la Sección 2.4.3, la hipersuperficie de decisión está dada por:

$$\sum_{i=1}^{6} \alpha_i \, t_i (\boldsymbol{x}^T \boldsymbol{x_i} + 1)^7 = 0. \tag{4.15}$$

Considerando que estamos trabajando con un problema de clasificación binario, la función de decisión puede escribirse como:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{6} \alpha_i \, t_i (\boldsymbol{x}^T \boldsymbol{x_i} + 1)^7 \right). \tag{4.16}$$

El diagrama de flujo para la identificación de nuevos objetos está en la Figura 4.17. La implementación del código para este diagrama de flujo ha sido realizado mediante un método que pertenece a la clase SVM, por lo

que todas las instancias de la clase SVM pueden hacer uso de este método para la clasificación de nuevas entradas.

La manipulación de los datos de entrada y salida al sistema de reconocimiento también es realizada mediante instancias de una clase que como parte de sus atributos tienen una variable X que denotan las entradas y una variable Y para las salidas. Toda esta estructura permite una manipulación de una forma más segura de los datos gracias a la propiedad de encapsulamiento de la programación orientada a objetos.

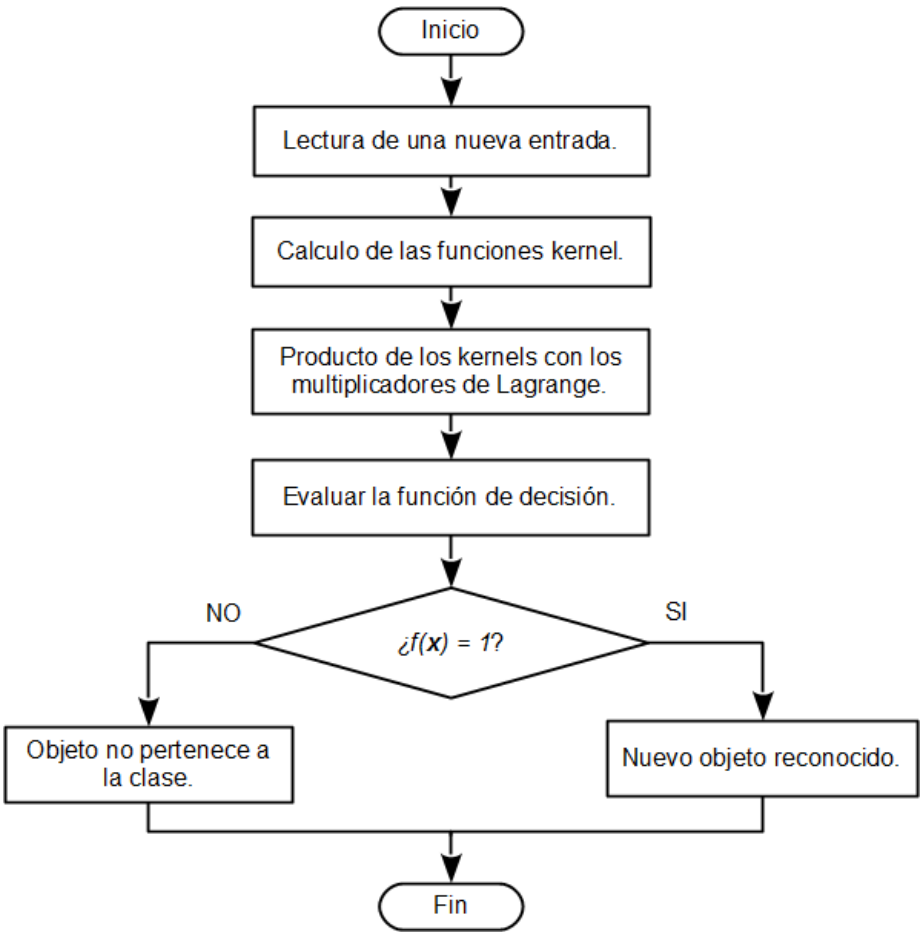


***Figura 4.17**: Diagrama de flujo para la identificación de nuevos objetos.*

4.6. Transformación de datos a través de los procesos del sistema

Cada uno de los algoritmos en los diferentes niveles del sistema de reconocimiento que se han descrito en las secciones anteriores, transforman los datos de entrada en información útil que es utilizada por la siguiente etapa, en la Figura 4.18 se observa cómo va cambiando la estructura de los datos a través de las diferentes etapas del sistema.

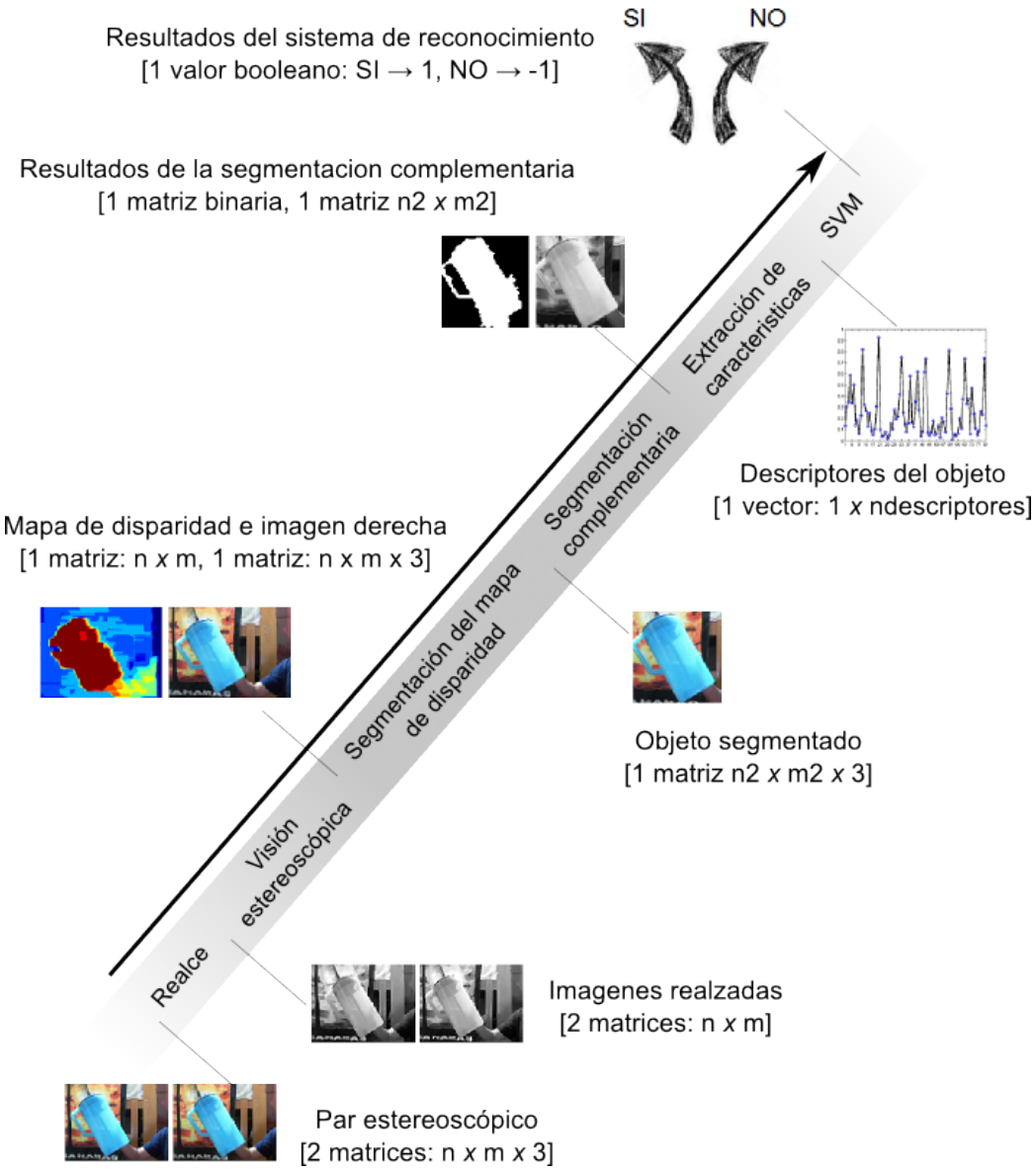


***Figura 4.18**: Transformación de los datos en las diferentes etapas del sistema de reconocimiento.*

Los datos iniciales están compuestos por un par estereoscópico que son el resultado de la captura de una escena en particular por el sistema estereoscópico, este par está compuesto de dos imágenes en color que son almacenadas en dos matrices de dimensiones $n\ x\ m\ x\ 3$, donde n representa el número de filas y m representa el número de columnas, estas imágenes están representadas en el modelo de color RGB, por lo que cada una tiene tres capas que contienen las intensidades de los tres colores de luz primarios. Mediante la aplicación de los algoritmos de realce los datos son transformados en dos imágenes en escala de grises de igual dimensión que las imágenes de entrada. Estas imágenes realzadas son las entradas para los algoritmos de correspondencia mediante los cuales los datos son transformados en el mapa de disparidad de la escena que se almacena en una matriz de dimensión $n\ x\ m$, en la figura anterior, el mapa de disparidad es mostrado en una escala de colores para mayor claridad. El algoritmo para la segmentación del mapa de disparidad que tiene como entradas al mapa de disparidad junto a la imagen derecha del par estereoscópico transforma estos datos en un objeto segmentado que está compuesto de una imagen en color almacenada en una matriz de dimensiones $n2\ x\ m2\ x\ 3$, donde $n2$ y $m2$ representan el número de filas y columnas de la imagen respectivamente. La segmentación complementaria convierte estos datos en una imagen binaria y una imagen en escala de grises para luego ser transformados en un vector de características mediante los algoritmos de extracción de características.

Finalmente, el vector de características de dimensión $1\ x\ ndescriptores$, donde $ndescriptores$ representa al número de descriptores utilizados, es convertido a un valor booleano mediante la ejecución de SVM, este valor booleano determina si el objeto presentado al sistema se ha reconocido o es un objeto desconocido para el sistema.

Capítulo 5

Simulaciones y resultados experimentales

En este capítulo se presentan las simulaciones y resultados experimentales que se obtuvieron luego de entrenar la máquina de aprendizaje para dos objetos en particular. El primer conjunto de objetos entrenados corresponde a esferas que tienen un color uniforme, para estos objetos los detalles de la máquina de aprendizaje fueron descritos en la Sección 4.5.1. Los resultados para el segundo objeto entrenado corresponden al reconocimiento de un conjunto de botellas que está compuesto de diferentes colores y formas. Las Secciones 5.1 y 5.2 presentan los resultados parciales del sistema, mientras que en la Sección 5.3 están los resultados finales que corresponden a los algoritmos de alto nivel.

5.1. Resultados de los algoritmos de bajo nivel

Los resultados presentados en esta sección corresponden a las técnicas empleadas para realzar las características de los objetos de interés y a los algoritmos utilizados para obtener el mapa de disparidad. Los resultados presentados en esta y la siguiente sección han sido obtenidos en base a dos pares

estereoscópicos (véase Figura 5.1), un par estereoscópico muestra un objeto esférico y el otro corresponde a la presentación de una botella al sistema.

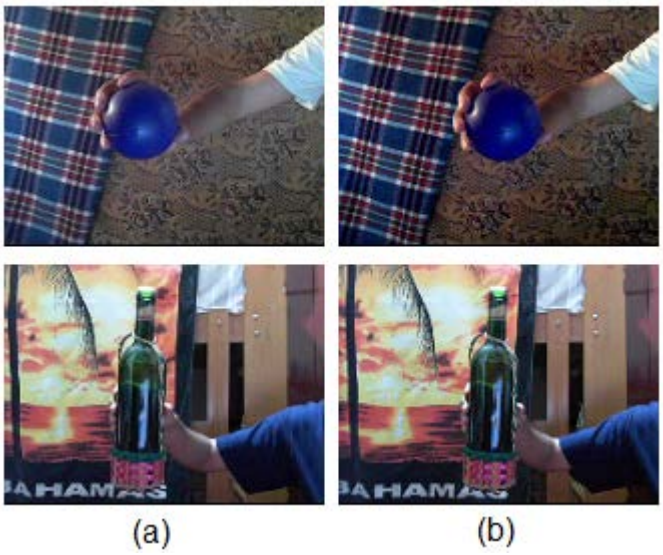


***Figura 5.1**: Par estereoscópico. (a) Imagen derecha; (b) Imagen izquierda.*

La escena capturada por el sistema en donde se encuentra el objeto a analizar debe tener de preferencia un fondo que presente textura, de esta forma se obtendrán mejores resultados en la etapa de correspondencia estereoscópica y que además tendrá un impacto en los resultados finales. Otro aspecto importante a considerar es la iluminación, las cámaras utilizadas en la tesis permiten cierto grado de variación en la intensidad de la iluminación de la escena. Sin embargo, para un buen desempeño de los algoritmos desarrollados en esta tesis se requiere que los objetos a reconocer, contenidos en el par estereoscópico, presenten contraste con el ambiente que los circunda.

5.1.1. Algoritmo de realce

Los resultados del algoritmo de realce que se pueden apreciar en la Figura 5.2 corresponden a la imagen derecha de cada uno de los pares

estereoscópicos mostrados anteriormente, estos resultados son obtenidos de acuerdo a lo descrito en la Sección 4.3.1.

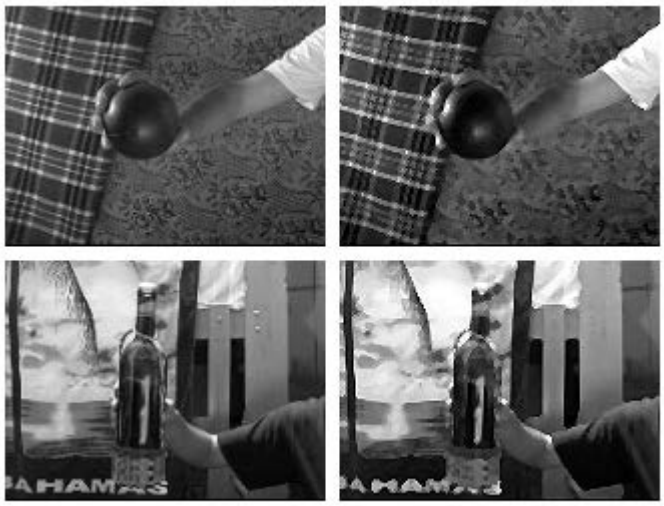


***Figura 5.2**: Resultado del proceso de realce para la imagen derecha. (a) Imagen de entrada; (b) Imagen realzada.*

5.1.2. Correlación estereoscópica

Como se menciono en la Sección 4.3.2, la correlación estereoscópica ha sido realizada por un método global y un método local. Los resultados que se encuentran en la Figura 5.3 verifican que los métodos globales son más efectivos que los métodos locales, y es por ello que han sido elegidos como método principal para el proceso de correlación estereoscópica.

Los mapas de disparidad mostrados están representados en una escala cromática de colores en un rango que va desde el azul al rojo, y que tiene como colores intermedios al cian, amarillo y anaranjado entre otros. En esta escala cromática los objetos más alejados del sistema tienen colores azulados mientras que los objetos que se encuentran a menor distancia están representados con colores rojizos.

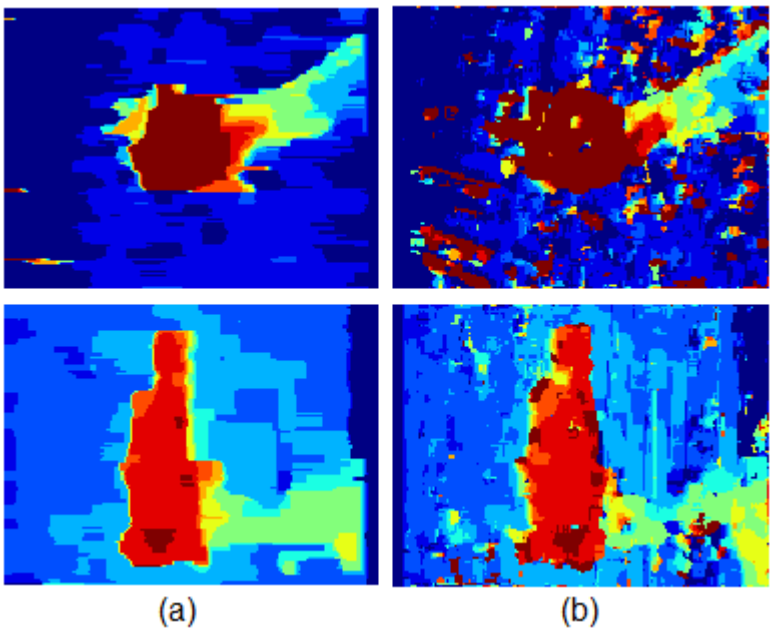


***Figura 5.3**: Resultados de la correlación estereoscópica. (a) Programación dinámica; (b) Correlación basada en ventanas.*

5.2. Resultados de los algoritmos de nivel intermedio

Los resultados de la implementación de los algoritmos de nivel intermedio como lo son la segmentación del mapa de disparidad, la segmentación mediante el método de división y fusión para objetos de color uniforme y la extracción de características son presentados en esta parte del capítulo.

5.2.1. Segmentación del mapa de disparidad

La segmentación del mapa de disparidad se realiza en varias etapas, en esta sección se presentan los resultados obtenidos de estas etapas para un objeto esférico.

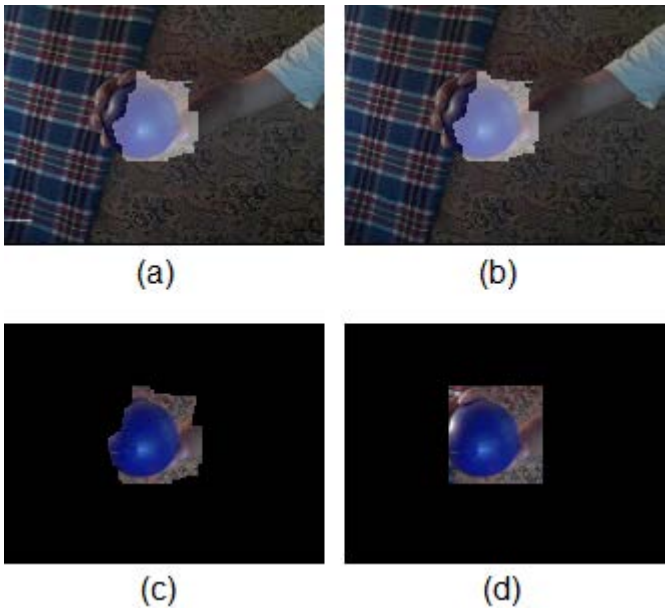


***Figura 5.4**: Etapas para la segmentación del mapa de disparidad.*

Como se menciono en la Sección 4.4.1, el mapa de disparidad está compuesto de varios niveles que corresponden a diferentes profundidades de la escena. La primera etapa del algoritmo determina el nivel más cercano al sistema que supere un cierto umbral para el área, los resultados de esta etapa pueden observarse en la Figura 5.4a. La segunda etapa verifica la cantidad de elementos conectados en el nivel seleccionado en la etapa anterior, si existe más de un elemento conectado en el nivel, se selecciona el de máxima área, en este caso en particular se han eliminado las dos líneas que se encontraban en la esquina inferior izquierda (véase Figura 5.4b). La etapa siguiente aísla y aplica procesamiento morfológico al resultado obtenido anteriormente, esto se observa en la Figura 5.4c, finalmente se halla el menor rectángulo circunscrito para la región resultante (véase Figura 5.4d).

El resultado final de la segmentación del mapa de disparidad para la escena que contiene a la botella está en la Figura 5.5.

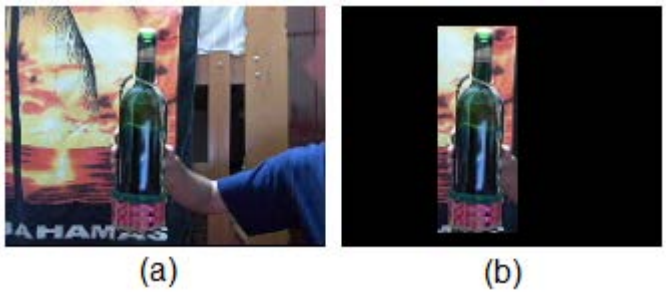


***Figura 5.5**: Segmentación del mapa de disparidad para la escena que contiene la botella. (a) Imagen derecha del par estereoscópico, (b) Objeto segmentado.*

Si los objetos que se están analizando son de color uniforme, se puede aplicar la segmentación complementaria que utiliza el método de división y fusión, descrito en la Sección 4.4.1. Esta segmentación puede aplicarse al objeto esférico que se ha estado analizando, ya que este tiene un color uniforme. El resultado de la aplicación del método a este objeto es mostrado en la Figura 5.6.

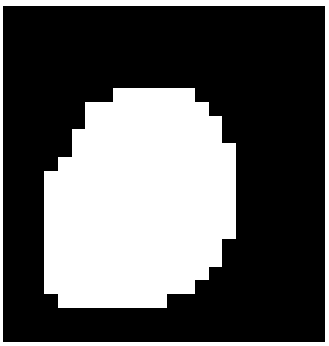


***Figura 5.6**: Imagen binaria del objeto esférico segmentado mediante algoritmos de división y fusión.*

5.2.2. Extracción de características

Los descriptores utilizados para el objeto esférico están conformados por los descriptores básicos y los momentos invariantes, estos descriptores son utilizados para todos los objetos que tienen un color uniforme. La Tabla 5.1 muestra el valor de los primeros ocho descriptores calculados.

ID	Descriptor	Valor
1	Excentricidad	0.6167
2	Solidez	0.9616
3	Extensión	0.8393
4	Número de Euler	1.0000
5	Compacidad	0.7958
6	Momento Invariante 1	0.1652
7	Momento Invariante 2	0.0015
8	Momento Invariante 3	0.0001

***Tabla 5.1**: Descriptores para un objeto esférico en particular.*

El gráfico en la Figura 5.7 muestra una curva construida con el valor de los descriptores del objeto esférico.

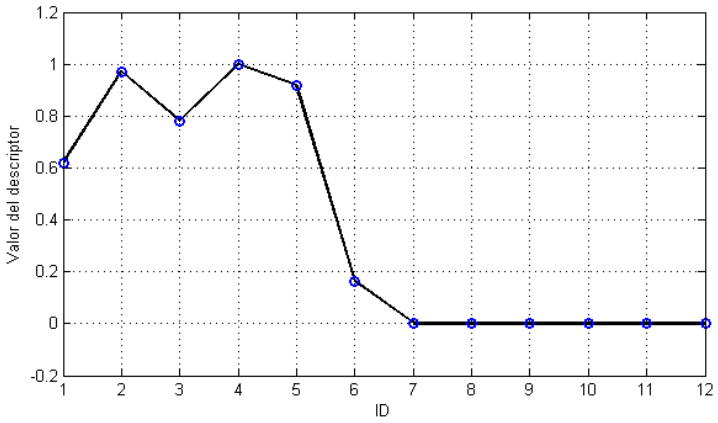


***Figura 5.7**: Curva de tendencia de los descriptores para el objeto esférico.*

Para objetos que no están compuestos de un color uniforme, como el caso de la botella, se utilizan los descriptores HOG. Los descriptores HOG están compuestos de 81 características como se menciono en la Sección 4.4.2, el gráfico en la Figura 5.8 muestra la curva generada por estos descriptores para la botella.

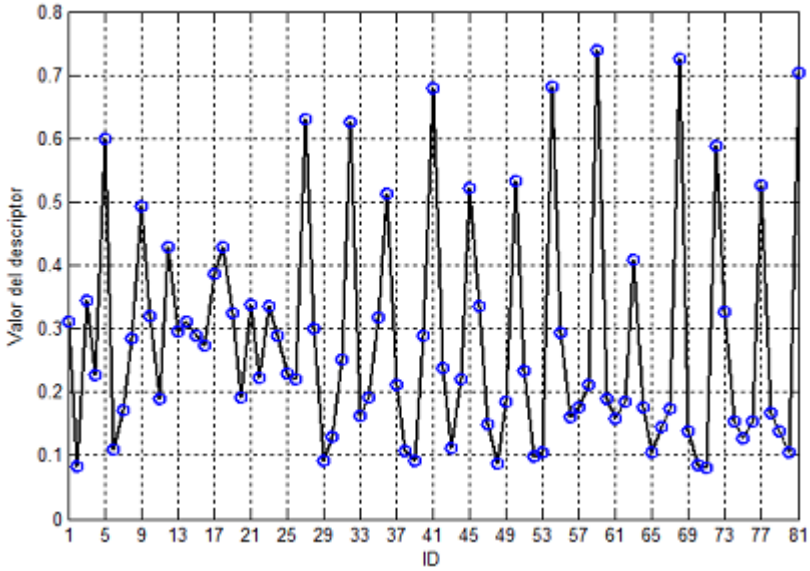


***Figura 5.8**: Curva generada a partir de los descriptores obtenidos para la botella.*

El valor de los primeros diez descriptores se muestra en la Tabla 5.2.

ID	Valor
1	0.3104
2	0.0830
3	0.3455
4	0.2264
5	0.6004
6	0.1101
7	0.1708
8	0.2843
9	0.4932
10	0.3211

***Tabla 5.2**: Primeros diez descriptores para la botella.*

5.3. Resultados de los algoritmos de alto nivel

La inteligencia artificial del sistema que se basa en SVM se encuentra implementada en los algoritmos de alto nivel y sus resultados son los más importantes ya que constituyen la respuesta final del sistema de reconocimiento. En la sección siguiente se presentan los resultados obtenidos en el reconocimiento de los objetos esféricos en diferentes situaciones, y en la última parte se muestran algunas de las imágenes utilizadas en el entrenamiento del sistema para el reconocimiento de botellas y los resultados obtenidos ante la presentación de nuevos objetos para la clasificación de las botellas.

5.3.1. Clasificación de nuevos objetos presentados al sistema de reconocimiento

El desempeño del sistema es evaluado mediante los resultados obtenidos ante la presentación de nuevos objetos que no se encuentran en el conjunto de entrenamiento. Parte de los nuevos objetos mostrados al sistema para el reconocimiento de los objetos esféricos se encuentra en la Figura 5.9.

Para realizar el proceso de reconocimiento y clasificación de un objeto de interés, primero se debe entrenar la máquina de aprendizaje. El conjunto de entrenamiento utilizado para el reconocimiento de botellas ha estado compuesto de 110 imágenes de diferentes tamaños, ochenta y cinco de estas imágenes son ejemplos positivos y el resto está compuesto de objetos varios. Algunas de las imágenes utilizadas son mostradas en las Figuras 5.10 y 5.11.

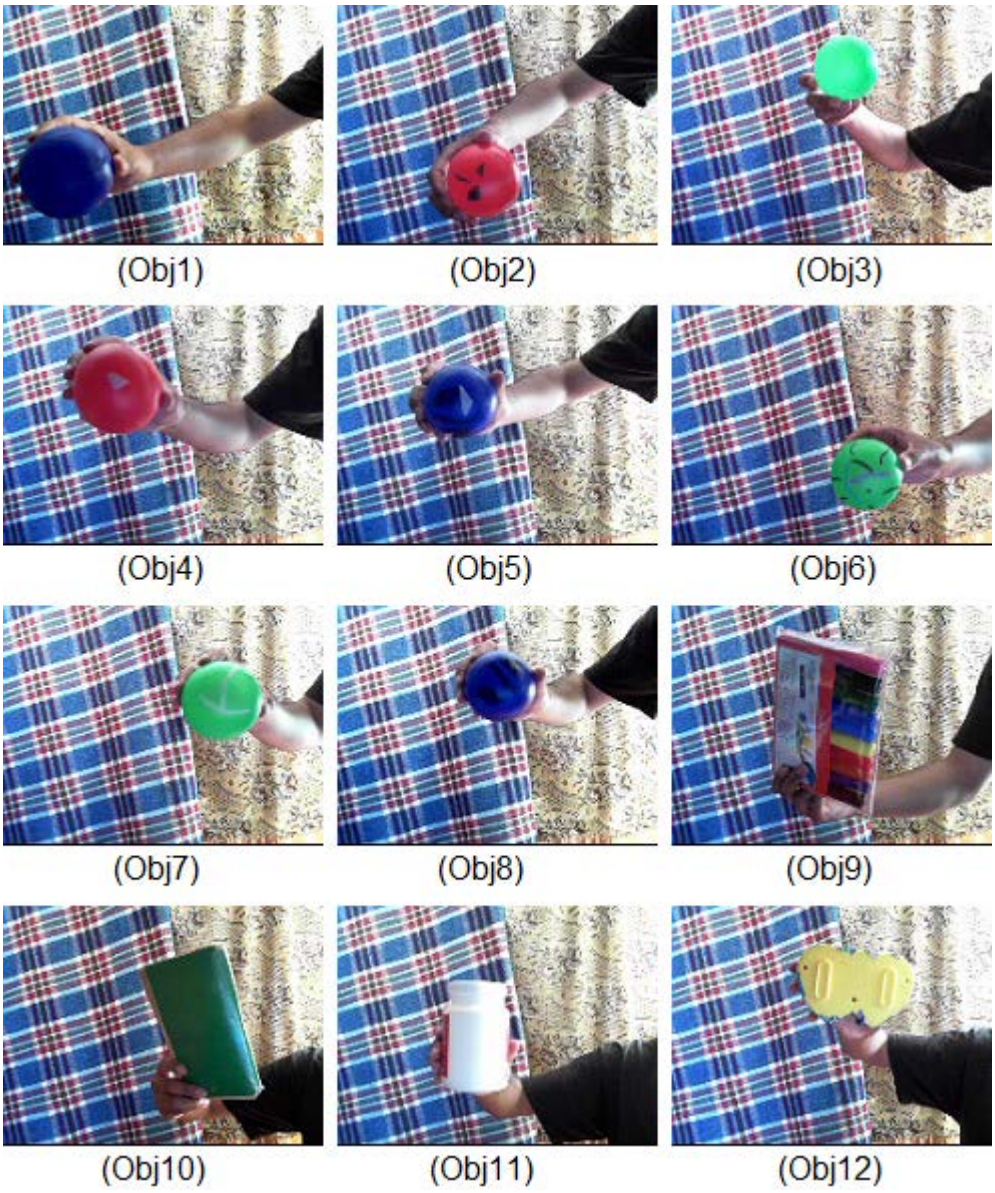


***Figura 5.9**: Objetos nuevos presentados al sistema para el reconocimiento de esferas. Todas las imágenes corresponden a la imagen derecha del par estereoscópico.*

En la Tabla 5.3 se muestran los resultados obtenidos en el reconocimiento de los objetos mostrados en la figura anterior.

ID	Salida esperada	Salida obtenida	Descripción
Obj1	1	1	Se ha detectado una esfera.
Obj2	1	1	Se ha detectado una esfera.

Obj3	1	1	Se ha detectado una esfera.
Obj4	1	1	Se ha detectado una esfera.
Obj5	1	0	El objeto no es una esfera.
Obj6	1	1	Se ha detectado una esfera.
Obj7	1	1	Se ha detectado una esfera.
Obj8	1	1	Se ha detectado una esfera.
Obj9	0	0	El objeto no es una esfera.
Obj10	0	0	El objeto no es una esfera.
Obj11	0	0	El objeto no es una esfera.
Obj12	0	0	El objeto no es una esfera.

***Tabla 5.3**: Resultados del sistema de reconocimiento frente a objetos esféricos.*

***Figura 5.10**: Parte de los ejemplos positivos utilizados en el entrenamiento de la máquina de aprendizaje para el reconocimiento de botellas.*

***Figura 5.11**: Parte de los ejemplos negativos utilizados en el entrenamiento de la máquina de aprendizaje para el reconocimiento de botellas.*

Luego del entrenamiento de la máquina de aprendizaje se procedió a presentar nuevos objetos al sistema de reconocimiento, la Figura 5.12 y 5.13 muestra algunos de los objetos presentados para evaluar si el sistema ha aprendido a clasificar las botellas de otros objetos.

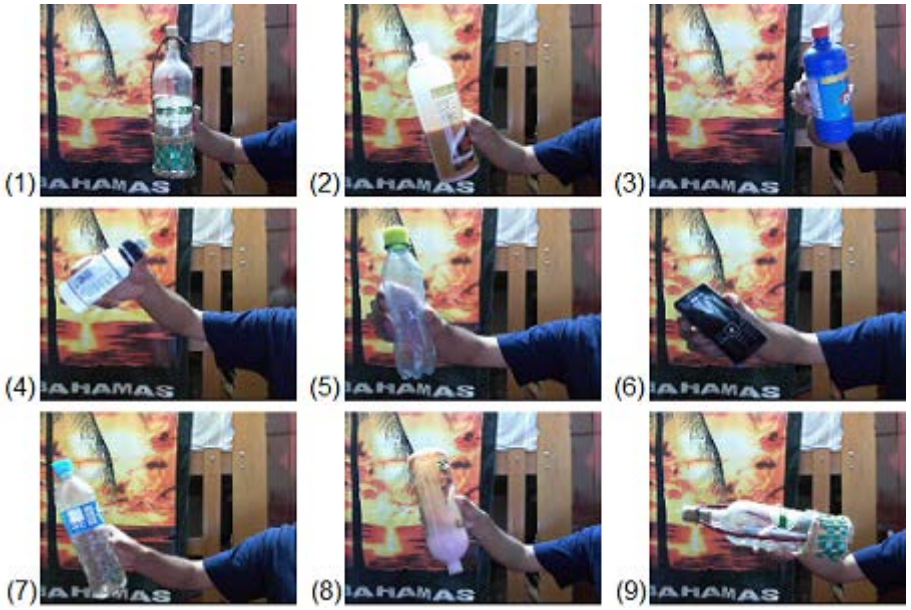


***Figura 5.12**: Objetos nuevos presentados al sistema para el reconocimiento de botellas, [1-9].*

***Figura 5.13**: Objetos nuevos presentados al sistema para el reconocimiento de botellas, [10-21].*

En la Tabla 5.4 se muestran los resultados obtenidos en la identificación de nuevos objetos para el entrenamiento de la máquina de aprendizaje en el reconocimiento de botellas.

ID	Salida esperada	Salida obtenida	Descripción
1	1	1	Se ha detectado un objeto de la misma clase
2	1	1	Se ha detectado un objeto de la misma clase
3	1	1	Se ha detectado un objeto de la misma clase
4	1	1	Se ha detectado un objeto de la misma clase
5	1	1	Se ha detectado un objeto de la misma clase
6	0	0	No se ha reconocido el objeto mostrado
7	1	1	Se ha detectado un objeto de la misma clase
8	1	1	Se ha detectado un objeto de la misma clase
9	1	0	No se ha reconocido el objeto mostrado
10	1	1	Se ha detectado un objeto de la misma clase

11	0	0	No se ha reconocido el objeto mostrado
12	1	1	Se ha detectado un objeto de la misma clase
13	0	0	No se ha reconocido el objeto mostrado
14	1	1	Se ha detectado un objeto de la misma clase
15	1	1	Se ha detectado un objeto de la misma clase
16	1	1	Se ha detectado un objeto de la misma clase
17	0	0	No se ha reconocido el objeto mostrado
18	1	0	No se ha reconocido el objeto mostrado
19	1	1	Se ha detectado un objeto de la misma clase
20	0	0	No se ha reconocido el objeto mostrado
21	1	1	Se ha detectado un objeto de la misma clase

***Tabla 5.4**: Resultados del sistema para el reconocimiento de botellas.*

Los resultados del sistema de reconocimiento para los dos objetos presentados en este capítulo muestran un buen desempeño del sistema, de las tablas anteriores podemos puntualizar que once de los doce objetos presentados al sistema para el reconocimiento de objetos esféricos fueron clasificados correctamente, esto equivale al 91.67% de aciertos. Por otra parte, el porcentaje de aciertos del sistema cuando se realizó el reconocimiento de las botellas fue de 90.48%, en este caso se presentaron 21 objetos, de los cuales 19 fueron clasificados correctamente.

Respecto al tiempo consumido por los diferentes procesos que se llevan a cabo en el procesador para la ejecución de los algoritmos, éste es en promedio de 1100 ms para el análisis completo de un par estereoscópico que involucra la ejecución de los algoritmos de bajo nivel, intermedio y alto nivel.

Capítulo 6

Discusión, conclusiones y recomendaciones

En este capítulo se presentan las discusiones finales, conclusiones y se proponen algunas recomendaciones para trabajos futuros.

6.1. Discusión de resultados

En este trabajo se busco diseñar un sistema de reconocimiento capaz de aprender a clasificar objetos reales usando visión estereoscópica e inteligencia artificial basada en SVM. En los siguientes párrafos se analizan los objetivos planteados frente a los resultados obtenidos.

El desarrollo de algoritmos que realizan el procesamiento individual de cada una de las imágenes es llevado a cabo por los algoritmos de realce, cuyos resultados cumplen los requisitos necesarios para poder ser procesados por los algoritmos de correspondencia. En la Figura 5.2 se puede observar que las imágenes en la derecha presentar mayor contraste y poseen una mayor nitidez. Estos algoritmos son estables y sencillos, además no dependen de variables aleatorias, por lo que podrían ser utilizados por otros sistemas de reconocimiento.

Los algoritmos implementados con el objetivo de solucionar el problema de correspondencia para la visión estereoscópica responden correctamente a los procedimientos descritos en el Capítulo 3. Los resultados presentados en la Figura 5.3 son suficientemente correctos para el objetivo general planteado. La programación dinámica es un método general que ha sido usada en esta tesis para resolver el problema de correspondencia, se ha demostrado que este método es superior a la correlación basada en ventanas y puede ser ejecutado en cualquier computador de propósito general.

La segmentación de la escena fue lograda mediante la ejecución de dos etapas independientes, en la primera etapa que es aplicada a todos los objetos, se obtienen resultados satisfactorios siempre que el objeto tenga un área mínima dentro de la imagen, esto puede ser modificado mediante los umbrales internos que posee el algoritmo. La segunda etapa que corresponde a la segmentación mediante el método Split & Merge (se aplica cuando los objetos de interés tienen un color uniforme), muestra buenos resultados en imágenes que tienen un fondo bien contrastado, éste método no es recomendable cuando se analicen objetos que no resalten con su fondo.

Los descriptores utilizados en la tesis han cumplido el objetivo planteado para esta etapa del sistema, los descriptores básicos funcionan adecuadamente para objetos de color uniforme, sin embargo, estos están limitados a analizar imágenes binarias a diferencia de los descriptores HOG cuyos resultados han servido como datos de entrada para la máquina de aprendizaje cuando los objetos están compuestos de varios colores. Los descriptores HOG son usados en aquellos casos donde se requiere características sobre todo el objeto en un modo de ventana deslizante, a diferencia de otros métodos para extraer características como

los descriptores SIFT [22], que son usados para relacionar regiones locales que han sido elegidos por un detector de puntos de interés.

Los resultados mostrados en la Sección 5.3.1 brindan el desempeño del sistema de reconocimiento, éstos son dados por las salidas de los algoritmos de alto nivel que dependen de los niveles inferiores para su buen desempeño. Estos resultados son validos siempre que los objetos presenten contraste con su fondo, ya que sin este requisito los algoritmos de bajo nivel tendrían resultados engañosos para los siguientes niveles de procesamiento. El sistema de reconocimiento presentado en esta tesis podrá ser aplicado siempre que se realice el entrenamiento adecuado de la máquina de aprendizaje, además la escena debe tener la iluminación suficiente para que los objetos presentados tengan contraste con su fondo. La calibración manual que debe realizarse al sistema estereoscópico antes de iniciar el proceso de reconocimiento es una desventaja frente a sistemas que realizan este procedimiento de forma automática como por ejemplo mediante la aplicación de la rectificación del par estereoscópico. Otro punto en contra del sistema de reconocimiento propuesto es que éste depende del entorno Matlab® para su ejecución, por lo que en su estado actual esto es una dificultad para que pueda ser integrado a sistemas de mayor complejidad que requieran reconocer objetos como parte de sus funciones para su operación.

El porcentaje de aciertos obtenido por el sistema de reconocimiento desarrollado en este trabajo es ligeramente superior al 90%, otros trabajos similares que realizan reconocimiento de objetos como en la clasificación de vehículos [23], el porcentaje de aciertos es de 63%. En el trabajo realizado en [23], se usa visión estereoscópica para segmentar la escena mediante la detección de los limites de un cuadro usando proyecciones de puntos a un área de interés definido por el

usuario, además se utiliza funciones kernel de base radial para los SVM que tienen coeficientes wavelet como vector de entrada. El uso de visión estereoscópica para segmentar la escena considerando el objeto de menor distancia al sistema y support vector machine con funciones kernel polinomiales han brindado una ventaja respecto a otros sistemas que utilizan técnicas diferentes u otras consideraciones en los niveles de procesamiento. Sin embargo, en el trabajo realizado en [23] los tiempos de ejecución de los algoritmos es de 300 ms a diferencia del tiempo tomado por los algoritmos desarrollados en esta tesis que es de 1100 ms. Esta diferencia en los tiempos de ejecución se debe principalmente al lenguaje de programación utilizado en la implementación de los algoritmos, en [23] se utilizo librerías desarrollas en C++ lo que explica esta ventaja respecto a los algoritmos propuestos ejecutados en el entorno Matlab®. El reconocimiento de objetos es realizado usando redes neuronales artificiales en otros trabajos, como en [24] donde se logro un porcentaje de aciertos de 75% que es menor a sistemas que utilizan support vector machines como técnica de inteligencia artificial, lo que reafirma la superioridad de este método en la resolución de ciertos problemas.

6.2. Conclusiones

En los siguientes párrafos se describen las conclusiones a las que se han llegado como resultado del trabajo realizado.

- La inteligencia artificial basada en SVM y visión estereoscópica del sistema de reconocimiento es capaz de aprender de ejemplos previos en la etapa de entrenamiento y posteriormente realizar una clasificación binaria de nuevos ejemplos presentados al sistema.

- La visión estereoscópica del sistema que utiliza programación dinámica en el algoritmo de correlación estereoscópica como método principal, tiene resultados efectivos en el cálculo del mapa de disparidad frente a métodos locales como la correlación basada en ventanas. La ventaja que tiene la programación dinámica es que ésta es más robusta frente a regiones localmente ambiguas, pero tiene como desventaja un mayor costo computacional y falta de coherencia vertical lo que provoca un efecto rayado horizontal en los resultados.

- El algoritmo para segmentar el mapa de disparidad tiene las ventajas, por un lado, extraer el objeto de interés correctamente para su posterior análisis utilizando un procesamiento modular en donde se puede realizar un seguimiento de los resultados de cada una de las etapas; y por otro lado, ser robusto frente a distorsiones presentes en el mapa de disparidad causadas por errores en la correlación estereoscópica. En contraposición, este algoritmo tiene por desventaja la determinación manual de umbrales de acuerdo al porcentaje de pixeles que ocupa el objeto en la imagen.

- La implementación de los algoritmos de extracción de características que calculan tres tipos de descriptores no tiene una alta complejidad, por lo que el tiempo de ejecución es pequeño y esto es una ventaja para el sistema ya que estos deben ser ejecutados para cada nuevo objeto que quiera ser reconocido.

- Mediante el uso de SVM y visión estereoscópica se obtiene un elevado porcentaje de aciertos en la tarea de reconocimiento de objetos que realiza una clasificación binaria, por lo que la visión estereoscópica y los principios

en los que se basan los SVM constituyen un paso importante en la forma de construir máquinas capaces de aprender de la experiencia.

- La adaptabilidad que tiene el sistema de reconocimiento es una ventaja frente a sistemas que se implementan para trabajar bajo ciertas condiciones en particular, el sistema presentado en esta tesis puede realizar el reconocimiento y la clasificación binaria de una variedad amplia de objetos si estos son entrenados adecuadamente mediante ejemplos previos para el reconocimiento del objeto de interés.

6.3. Recomendaciones para trabajos futuros

Las recomendaciones están dirigidas al empresariado innovador de tecnología en los procesos de producción, así mismo al sector académico que esté interesado en procesos automatizados para el reconocimiento de objetos o temas afines.

- Aplicar el sistema de reconocimiento y clasificación en la solución de un problema concreto aprovechando información a priori del objeto y de la escena en particular, como puede ser en el caso de la clasificación de frutos maduros de los que aun no deben ser cosechados, o en una tarea de exploración realizada por un robot que debe evitar obstáculos en donde la visión estereoscópica brinda información importante acerca de la profundidad.

- Utilizar la visión estereoscópica en aplicaciones que requieran conocimiento de profundidad y además puede ser complicado aplicar otras técnicas como sucede en la medicina para realizar endoscopias o cirugía cardiovascular entre otros, en donde se utilizan endoscopios con dos mini cámaras que

permiten obtener dos imágenes diferentes que posteriormente pueden ser utilizados para realizar diagnósticos mediante el reconocimiento de irregularidades en las imágenes.

- Considerar el uso del sistema de reconocimiento y clasificación de objetos que ha sido desarrollado en esta tesis, como base de nuevas investigaciones y temas afines, ya que se obtuvieron resultados satisfactorios durante su operación.

- Fortalecer el uso de support vector machine como técnica principal en sistemas que hagan uso de inteligencia artificial en su operación, puesto que su performance es normalmente superior frente a otras técnicas como las redes neuronales.

- Implementar el código para los algoritmos en otros lenguajes de programación como Java, Python o C++, para que de esta forma el sistema de reconocimiento sea independiente de la plataforma Matlab®, considerando además que la eficiencia de los tiempos de ejecución mejorará considerablemente cuando los algoritmos sean implementados en estos lenguajes de programación.

- Desarrollar el sistema embebido para los procesos de reconocimiento y clasificación implementados en esta tesis, el cual permitiría su utilización sobre un robot explorador o en una línea de producción.

- Integrar al sistema de reconocimiento un conjunto de actuadores para que de acuerdo a la tarea a realizarse éste pueda interactuar con el espacio tridimensional, como por ejemplo en la clasificación física de los objetos.

Bibliografía

[1] GOLDBERG, Steven B.; MAIMOME, Mark W.; MATTHIES, Larry; *Stereo Vision and Rover Navigation Software for Planetary Exploration*, IEEE Aerospace Conference Proceedings, Montana, USA, Marzo 2002. <http://www-robotics.jpl.nasa.gov/publications/Mark_Maimone/aero.pdf>

[2] MAKI, J. N.; BELL III, J. F.; HERKENHOFF, K. E.; SQUYRES, S. W.; KIELY, A.; KLIMESH, M.; SCHWOCHERT, M.; LITWIN, T.; WILLSON, R.; JOHNSON, A.; MAIMONE, M.; BAUMGARTNER, E.; COLLINS, A.; WADSWORTH, M.; ELLIOT, S. T.; DINGIZIAN, A.; BROWN, D.; HAGEROTT, E. C.; SCHERR, L.; DEEN, R.; ALEXANDER, D.; LORRE, J.; *Mars Exploration Rover Engineering Cameras*, Journal of Geophysical Research, Vol. 108, USA, 11 de Diciembre del 2003, NO. E12, 8071.

[3] Jet Propulsion Laboratory, Mars Science Laboratory – Curiosity Rover, Eyes and Other Senses, [consulta: 2013-01-27], Disponible en: <http://mars.jpl.nasa.gov/msl/mission/rover/eyesandother/>

[4] Google Inc., Search by Image – How it works, [consulta: 2012-01-22], Disponible en: <http://www.google.com/insidesearch/searchbyimage.html >

[5] SHOTTON, Jamie; FITZGIBBON, Andrew; COOK, Mat; SHARP, Toby; FINOCCHIO, Mark; MOORE, Richard; KIPMAN, Alex; BLAKE, Andrew; *Real-Time Human Pose Recognition in Parts from Single Depth Images*, Microsoft Research Cambridge & Xbox Incubation, 2011.

[6] DIFTLER, M. A.; AMBROSE, R. O.; TYREE, K. S.; GOZA, S.M.; HUBER, E.L.; *A Mobile Autonomous Humanoid Assistant*, Div. of Autom., Robotics, & Simulation, NASA Johnson Space Center, Houston, Texas, USA, Junio 2005, ISBN: 0-7803-8863-1.

[7] SAKAGAMI, Yoshiaki; WATANABE, Ryujin; AOYAMA, Chiaki; MATSUNAGA, Shinichi; HIGAKI, Nobuo; FUJIMURA, Kikuo; *The intelligent ASIMO: System overview and integration*, IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, Octubre 2002.

[8] Honda Motor Co., Ltd., Public Relations Division, *The Honda HUMANOID ROBOT*, Información técnica, Septiembre 2007, pp. 22.

[9] CRISTIANINI, Nello; TAYLOR, John Shawe; *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2000, ISBN:0521780195.

[10] VAPNIK, Vladimir N., *The Nature Of Statistical Learning Theory*, Springer, New York, 1995.

[11] HAYKIN, Simon, *Neural Networks: A Comprehensive Foundation*, 2nd, Prentice Hall, 1999, pp. 318-350, ISBN: 0-13-273350-1.

[12] SALOMON, Jesper, "*Support Vector Machines for Phoneme Classification*" [tesis para optar el grado de magister], Edinburgh, Scotland, UK, University of Edinburgh, 2001, pp. 9-32.

[13] ARMANGUÉ QUINTANA, Xavier, "*Modelling Stereoscopic Vision Systems for Robotic Applications*" [tesis doctoral], Cataluña, España, Universidad de Gerona, Julio 2003, pp. 1, 2, 49-71.

[14] LECUMBERRY RUVERTONI, Federico, "*Cálculo de Disparidad y Segmentación de Objetos en Secuencias de Video*" [tesis para optar el grado de magister], Montevideo, Uruguay, Universidad de la República, 2005.

[15] KOENDERINK, J.J.; VAN DOORN, A.J.; "*Geometry of binocular visión and a model for stereopsis*", Biological Cybernetics 21, no. 1, Springer-Verlag, 1976, pp. 29-35.

[16] SUCAR, Luis Enrique; GOMEZ, Giovanni; "*Visión Computacional*", Departamento de Computación ITESM, Cuernavaca, Mexico, 2011.

[17] SOMOLINOS SÁNCHEZ, José Andrés, *Avances en Robótica y Visión por Computador*, Ediciones de la Universidad de Castilla-La Mancha, España, 2002, pp. 56,57, ISBN: 84-8427-199-4.

[18] MERCIMEK, Muharrem; GULEZ, Kayhan; VELI MUMCU, Tarik; "*Real object recognition using moment invariants*", Sadhana Vol 30 Part 6, Turkey, 2005, pp. 765-775.

[19] LUDWIG, O.; DELGADO, D.; GONCALVES, V.; NUNES, U.; "*Trainable classifier-fusion schemes: an application to pedestrian detection*", Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on. IEEE, 2009.

[20] DALAL, Navneet; TRIGGS, Bill; "*Histograms of oriented gradients for human detection*", Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[21] WESTON, Jason; ELISSEEFF, Andre; BAKIR, Gökhan; SINZ, Fabian, The Spider – Librería de objetos en Matlab, [consulta: 2012-05-07], Disponible en: <http://people.kyb.tuebingen.mpg.de/spider/main.html>

[22] LOWE, David, "*Distinctive image features from scale-invariant keypoints*", International Journal of Computer Vision, vol. 60, pp. 91-110, 2004.

[23] PAYSAN, Pascal, "*Stereovision based vehicle classification using support vector machines*" [tesis para obtener el diploma de ingeniero], Alemania, University of Applied Sciences Fachhochschule Esslingen, 2004.

[24] DE VRIES, Jelmer. "*Object Recognition: A Shape-Based Approach using Artificial Neural Networks.*" Department of Computer Science, University of Utrecht, 2006.

Anexos

A.1. Interfaz gráfica del sistema de reconocimiento

En la Figura A.1 se muestra la interfaz del sistema de reconocimiento que permite la selección del objeto a reconocer desde una lista, el método de correspondencia estereoscópica es seleccionado mediante botones de opción, en donde la programación dinámica es usada como método principal respecto a la correlación basada en ventanas. Finalmente, el sistema puede iniciar su ejecución en un modo manual o automático que debe indicarse desde la interfaz gráfica.

***Figura A.1**: Interfaz del sistema de reconocimiento.*

A.2. Diagrama de flujo (programación dinámica)

En las Figuras A.2 y A.3 se puede apreciar el diagrama de flujo para la correspondencia estereoscópica utilizando programación dinámica.

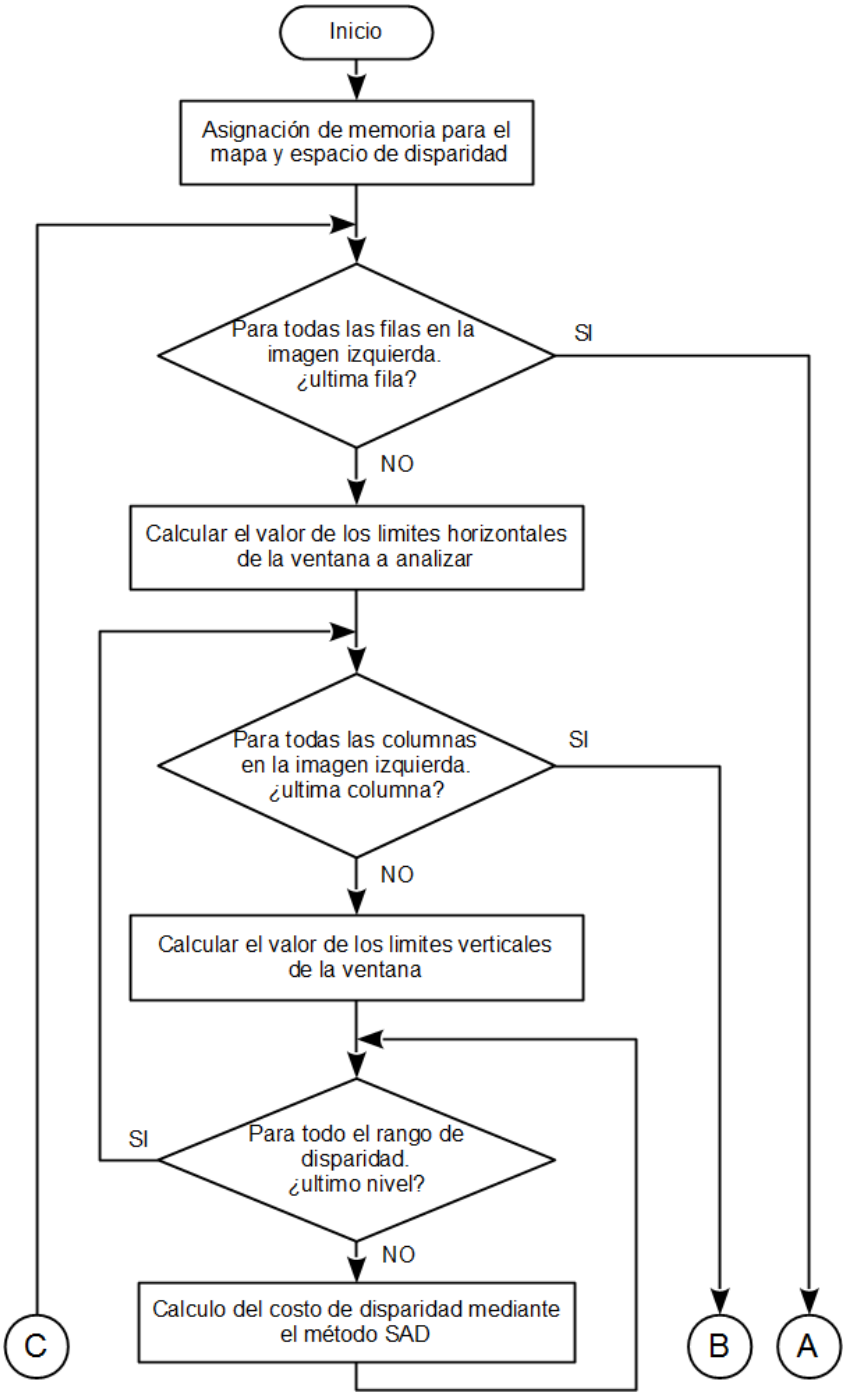


***Figura A.2**: Diagrama de flujo del algoritmo para la correspondencia estereoscópica utilizando programación dinámica. Parte 01.*

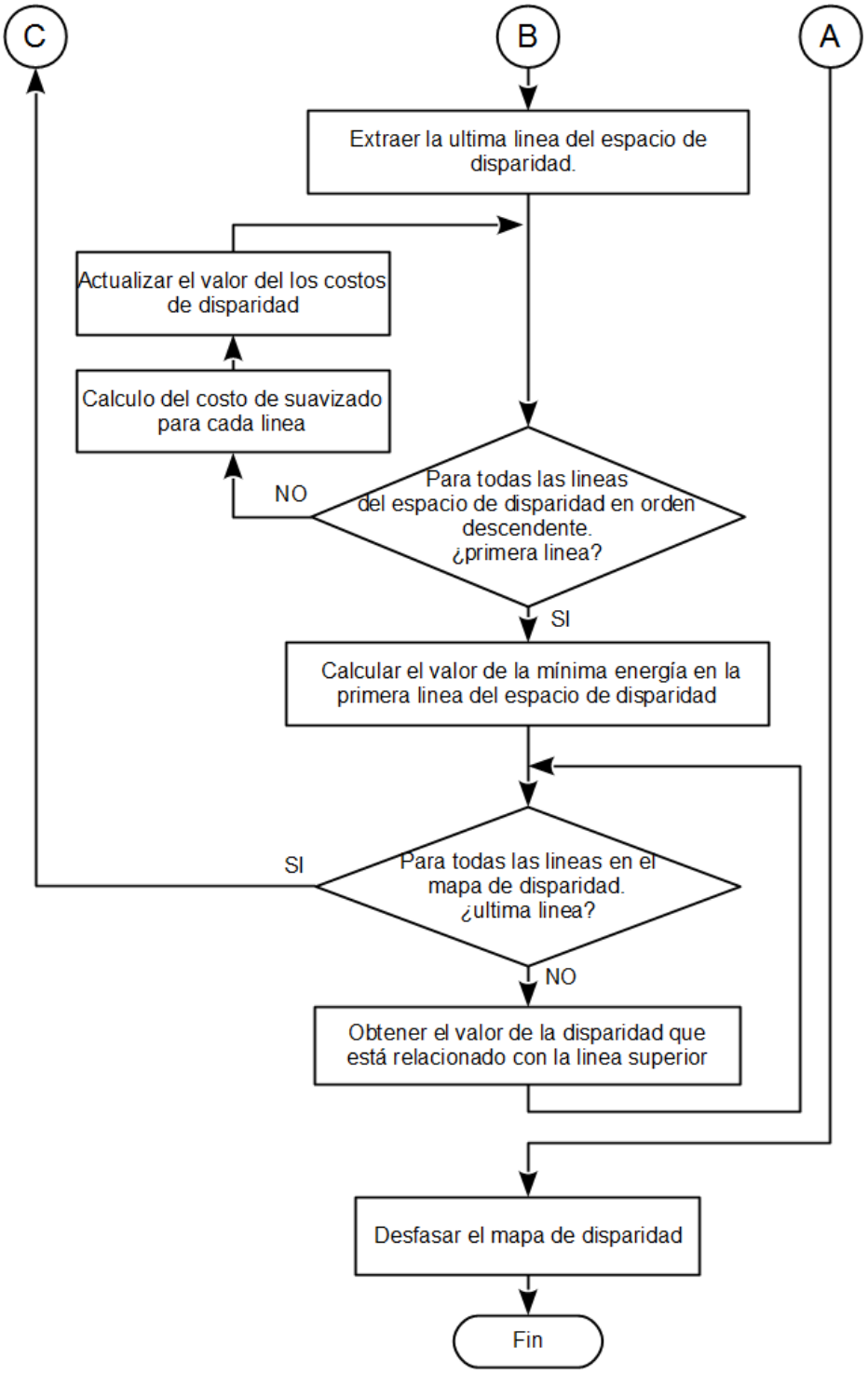


***Figura A.3**: Diagrama de flujo del algoritmo para la correspondencia estereoscópica utilizando programación dinámica. Parte 02.*

A.3. Operaciones morfológicas básicas

Para la definición de las operaciones básicas del procesamiento morfológico en imágenes binarias, se precisa mencionar algunas operaciones sobre conjuntos.

Reflexión: La reflexión del conjunto B, denotado como $\hat{B}$, es definida como:

$$\hat{B} = \{w | w = -b, \ para \ \ b \in B\}. \tag{A.1}$$

Traslación: La traslación de un conjunto A por el punto $z = (z_1, z_2)$, denotado $(A)_z$, es definido como:

$$(A)_z = \{c | c = a + z, \ para \ \ a \in A\}. \tag{A.2}$$

De esta forma, los operadores morfológicos fundamentales se definen en las siguientes líneas.

Dilatación: Sean A y B conjuntos de Z^2 y $\emptyset$ el conjunto vacio, se define la dilatación de A por B, denotada por $A \oplus B$, como:

$$A \oplus B = \{x | \left(\hat{B}\right)_x \cap A \neq \emptyset\}, \tag{A.3}$$

el conjunto B es normalmente llamado elemento estructural.

Erosión: Sean A y B conjuntos de Z^2, se define la erosión de A por B, denotada por $\mathrm{A} \ominus B$, como:

$$\mathrm{A} \ominus B = \{x | (B)_x \subseteq \mathrm{A}\}. \tag{A.4}$$

De la ecuación (A.4), la erosión de un conjunto A por un elemento estructural B, es el conjunto de todos los elementos x para los cuales B trasladado por x está contenido en A.

A.4. Pseudocódigo de los algoritmos principales

En esta sección se presenta el pseudocódigo de los algoritmos que realizan los procesos más importantes.

Algoritmo 1. Programación dinámica para la correspondencia estereoscópica.

```
lectura del par estereoscópico y parámetros iniciales
dPD ← matriz nula de igual dimensión que la imagen izquierda
[nRowsLeft, nColsLeft] ← numero de filas y columnas en la imagen
                         izquierda
finf ← 1000 // falso infinito
disparityCost ← asignación de memoria para la matriz que almacena el
                espacio de disparidad
for todas las filas en la imagen izquierda do
    disparityCost ← todos los elementos de la matriz igual a finf
    [minr, maxr] ← límite superior e inferior del bloque a analizar
    for todas las columnas en la imagen izquierda do
        [minc, maxc] ← límite izquierdo y derecho del bloque a
                       analizar
        [mind, maxd] ← asignar el rango de disparidad
        for todo el rango de disparidad do
            disparityCost ← costo de disparidad para el punto
                            actual en la imagen mediante SAD
        end for
    end for
    optimalIndices ← matriz nula de igual dimensión que disparityCost
    cp ← ultima fila de la matriz disparityCost
    for todas las filas del espacio de disparidad (orden descendente) do
        v ← costo de suavizado para la fila actual
        cp ← fila actual del espacio de disparidad + v
        optimalIndices ← almacenar ubicación de la ruta optima
                         para la fila actual
    end for
    ix ← ubicación del mínimo valor en el vector cp
```

```
        dPD(fila actual, 1)  ← ix
        for columnas desde la segunda hasta la última en el mapa de
            disparidad do
                dPD  ← disparidad para la columna actual de acuerdo a la
                        matriz optimalIndices
        end for
    end for
    dPD  ← dPD -1
```

Algoritmo 2. Procesamiento del mapa de disparidad.

```
    lectura del mapa de disparidad e imagen derecha
    objects  ← determinar la cantidad de niveles en el mapa de disparidad y
                ordenarlos en forma descendente numéricamente
    for  todos los valores en objects  do
        bw  ← imagen binaria para el nivel actual en objects
        S  ← cálculo del área en la imagen binaria bw
        if  área en S > Umbral de área  then
            break;
        end if
    end for
    auxNR  ← número de objetos en el nivel seleccionado
    if  auxNR > 1 then
        S.area  ← cálculo del área para cada región
        bw  ← región de máxima área
    else if  auxNR == 0
        bw  ← negación de bw
    end if
    // procesamiento morfológico //
    se  ← crear el elemento estructural
    bw  ← operación de cerrado entre bw y se
    bw  ← dilatar bw con el elemento estructural se
    // extraer el objeto seleccionado
    object  ← imagen derecha donde bw es igual a la unidad
```

Algoritmo 3. Split and Merge.

```
lectura de la imagen, criterio de similitud y mínimo tamaño de bloque
[m, n] ← dimensiones de la imagen de entrada
if m ó n no es potencia de 2 then
        nl ← nueva dimensión de la imagen que es potencia de 2
        I ← rellenar con ceros para obtener una imagen cuadrada
end if
S ← matriz nula de dimensión igual a la imagen I que almacena los
      bloques divididos
dim ← nl
while dim > mínimo tamaño del bloque do
        blockValues ← obtener bloques a analizar
        Sind ← obtener los índices de los bloques
        if Sind es vacio then
                break;
        end if
        dim ← dim/2
        doSplit ← evaluar que bloques no cumplen el criterio de similitud
        S ← Actualizar la matriz dividiendo los bloques respectivos
end while
L ← matriz nula de dimensión igual a la imagen I para almacenar etiquetas
nseg ← 1
while exista algún elemento igual a cero en L do
        ind2Analyze ← buscar bloque no etiquetado
        while no esté vacio ind2Analyze do
                L ← actualizar matriz de etiquetas asignando la etiqueta
                      nseg al bloque actual
                indN ← índices de los bloques adyacentes
                dN ← diferencia entre el valor del bloque actual y los
                       bloques adyacentes
                indNeighborsSimilar ← índices de bloques adyacentes que
                       cumplen el criterio de similitud
                ind2Analyze ← actualizar lista de índices para analizar
        end while
```

```
        nseg  ← nseg + 1
    end while
    L  ← L recortado al tamaño original de la imagen
```

Algoritmo 4. Extracción de características.

```
    lectura de la imagen binaria y en escala de grises del objeto a analizar
    lectura de fdu que determina los descriptores a ser devueltos
    // cálculo de los descriptores básicos sobre la imagen binaria //
    db  ← matriz nula de dimensión 5x1  // descriptores basicos
    db(1)  ← cálculo de la excentricidad
    db(2)  ← cálculo de la solidez
    db(3)  ← cálculo de la extensión
    db(4)  ← cálculo del numero de Euler
    db(5)  ← cálculo de la compacidad
    // cálculo de los momentos invariantes sobre la imagen binaria //
    dmi  ← momentos invariantes para la imagen binaria
    if  fdu es igual a 1  then
        // cálculo de los descriptores HOG sobre la imagen es escala de
        grises //
        H  ← matriz nula de dimensión 81x1
        grad_x  ← cálculo del gradiente en x
        grad_y  ← cálculo del gradiente en y
        ang  ← cálculo de la orientación en cada punto de la imagen
        mag  ← cálculo de la magnitud en cada punto de la imagen
        cont  ← 0
        for  número de bloques eje horizontal do
            for  número de bloques eje vertical do
                cont  ← cont + 1
                ang2  ← extraer las orientaciones desde ang para el
                        bloque actual considerando un solape
                mag2  ← extraer las magnitudes desde mag para el
                        bloque actual considerando un solape
                bin  ← 0
```

```
                    H2 ← matriz nula de dimensión 9x1
                    for todos los segmentos angulares do
                        bin ← bin +1
                        for todos los elementos en mag2 do
                            if orientación pertenece al segmento
                                angular actual then
                                H2 (bin) ← H2(bin) + magnitud
                                    del elemento actual
                            end if
                        end for
                    end for
                    H2 ← normalizar H2
                    H((cont-1)*9+1:cont*9,1) ← H2
                end for
            end for
            dHOG ← H
            vdescriptores ← dHOG
        else
            vdescriptores ← concatenar db y dmi
        end if
```

Algoritmo 5. Entrenamiento de SVM (Matlab®).

```
// el algoritmo de entrenamiento de la máquina de aprendizaje consiste en
resolver un problema de optimización cuadrática //
// en el desarrollo de este algoritmo ha sido utilizado el toolbox spider //
mai ← crear una instancia de la clase SVM
k ← crear una instancia de la clase kernel
mai.child ← k
mai.C ← asignación del parámetro C
datos ← lectura del conjunto de entrenamiento
t ← obtener las respuestas deseadas desde la variable datos
KerMa ← cálculo de la matriz de kernels (matriz cuadrada)
len ← número de filas de la matriz KerMa
```

H ← -1∗KerMa ∗ (t ∗ t^T)
f ← matriz con elementos igual a la unidad de dimensión len x 1
Aeq ← t^T
beq ← 0
[lb, ub] ← [0, límite superior para los valores de alpha (mai.C)]
opts ← estructura de opciones adicionales
alpha ← resolver el problema de optimización cuadrática con restricciones
fin ← determinar los valores de alpha diferentes de cero
mai ← actualizar mai con los valores de alpha mediante la variable fin

Algoritmo 6. Reconocimiento de nuevos objetos.

mai ← lectura de la máquina de aprendizaje
dat ← lectura del nuevo objeto a identificar
Xsv ← determinar los vectores de soporte desde el objeto mai
alpha ← multiplicadores de Lagrange asociados a Xsv
t ← target asociado a los vectores de soporte
KerMaTest ← obtener la matriz de kernels
g ← alpha ∗ t ∗ KerMaTest
f ← signo de la variable g
if *f es igual a la unidad* **then**
    out ← objeto identificado
else
    out ← objeto no pertenece a la clase
end if

A.5. Funciones utilizadas en Matlab®

Matlab® es un excelente entorno para simular algoritmos propuestos debido a la disponibilidad de funciones matemáticas ya implementadas y la facilidad de añadir nuevas funciones. Las funciones utilizadas e implementadas para los diferentes algoritmos son descritas en la Tabla A.1.

Nombre del archivo	Descripción
mainPT.m	*Programa principal para el sistema de reconocimiento desde donde se invoca a todas las demás funciones implementadas.*
ehcm.m	*Realza los detalles de interés en una imagen.*
dispPD.m	*Realiza la correspondencia estereoscópica utilizando programación dinámica.*
dispBM.m	*Realiza la correspondencia estereoscópica utilizando correlación basada en ventanas.*
procesarMapaDisp.m	*Procesa el mapa de disparidad para obtener una primera segmentación.*
mainSeg.m	*Función principal para la segmentación de la imagen para posteriormente obtener los descriptores.*
sefcam.m	*Segmentación de una imagen utilizando el método split and merge.*
qtdecomp2.m	*Función que realiza el proceso de división de una imagen calculando un valor representativo para cada bloque.*
mergesefcam.m	*Realiza el proceso de fusión para el método split and merge.*
indNeighborsv3.m	*Función que encuentra los índices de los bloques vecinos en el proceso de fusión.*
mainDescriptores.m	*Función principal para obtener los descriptores de los objetos.*
invMoments.m	*Realiza el cálculo de los momentos invariantes.*
hog.m	*Realiza el cálculo de los descriptores HOG.*
svm.m	*Genera un objeto svm con los parámetros brindados.*
kernel.m	*Crea un objeto kernel para el cálculo de productos internos en el espacio de características.*
get_kernel.m	*Calcula la matriz de kernels con datos específicos.*
data.m	*Almacena los datos en dos componentes X (input) e Y (output).*
get_x.m	*Retorna la matriz X de un objeto de datos.*
get_y.m	*Retorna la matriz Y de un objeto de datos.*
train.m	*Función principal para el entrenamiento de la*

	máquina de aprendizaje.
training.m	*Realiza el proceso de entrenamiento propiamente dicho.*
quadprogPT.m	*Resuelve el problema de optimización cuadrático incluyendo una barra de progreso.*
test.m	*Función principal para la identificación de nuevos objetos.*
testing.m	*Realiza el proceso de identificación propiamente dicho.*

***Tabla A.1**: Funciones utilizadas en Matlab®.*