

SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES

LUIS HERNANDO RÍOS GONZÁLEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA ELÉCTRICA
PEREIRA
2015

SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES

LUIS HERNANDO RÍOS GONZÁLEZ

Proyecto de Grado

Magister Alfonso Álzate Gómez
Director de Proyecto de Grado

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA ELÉCTRICA
PEREIRA
2015

DEDICATORIA

A DIOS, por cada instante de vida que me ha dado.

A mi PADRE, que desde el cielo ha sido mi Guía y Faro.

A mi MADRE por su entrega, dedicación y AMOR desmesurado.

A LAURITA, Mi hija, que es la inspiración de cada instante de vida.

A todos mis SERES QUERIDOS, por hacer parte de mi vida y darme la fortaleza necesaria en los momentos más difíciles de mi Existencia.

AGRADECIMIENTOS

En primer lugar quiero dar las gracias a la UNIVERSIDAD TECNOLÓGICA DE PEREIRA, mi sitio de trabajo, por permitir que cada día pueda ser un mejor profesional.

Gracias al Director del Proyecto de Grado MsC Alfonso Alzate Gómez, por su dedicación y apoyo durante la realización de este proyecto.

Del mismo modo, también quiero agradecer a los Doctores Domenec Puig Valls y Miguel Angel Garcia por su inmenso conocimiento, su apoyo y su guía en el desarrollo de este trabajo.

A los Doctores Maximiliano Bueno López y Paulo Andrés Muñoz por el tiempo dedicado a la revisión y evaluación de este proyecto.

A la MsC Marcela Botero Arbelaez, “Mi correctora de estilo” por su apoyo incondicional, las largas horas de trabajo, su amistad, su amor y compañía, la cual me dio la fuerza necesaria en los momentos más difíciles.

Además quiero agradecer a todos y cada una de las personas que hacen parte de mi vida y que son la razón de ser para que cada día sea una fiesta por vivir.

El Autor de este proyecto agradece al grupo de visión por computador del doctor ANDREA VEDALDI por la maravillosa página, cuya ayuda fue invaluable.

The author of this project thanks to computer vision group Dr. ANDREA VEDALDI for the wonderful page, whose help was invaluable.

GRACIAS! GRACIAS! GRACIAS!!!!

CONTENIDO

	Pág.
RESUMEN	9
INTRODUCCIÓN	10
1. FORMULACIÓN DEL PROBLEMA	13
2. JUSTIFICACIÓN	15
3. OBJETIVOS	17
3.1 OBJETIVO GENERAL	17
3.2 OBJETIVOS ESPECÍFICOS	17
4. MARCO TEÓRICO	18
4.1 RECONOCIMIENTO DE OBJETOS	18
4.1.1 Reconocimiento de Instancias	18
4.1.2 Reconocimiento de Categorías	19
4.2 EXTRACCIÓN DE CARACTERÍSTICAS	19
4.3 DESCRIPTORES GLOBALES DE IMÁGENES	20
4.4 DESCRIPTORES LOCALES DE IMAGEN	21
4.4.1 Puntos de Interés	21
4.4.1.1 Detector de Esquinas de HARRIS	22
4.4.2 Detector SIFT (Scale Invariant Feature Transform)	30
4.4.2.1 Cálculo de Correspondencias (matching)	45
4.4.3 Detector SURF (Speed up Robust Features)	46
4.4.3.1 Detección de Puntos de Interés	46
4.4.3.2 Asignación de la Orientación	49
4.4.3.3 Descriptores SURF	50
4.5 DESCRIPTORES DE IMÁGENES SEMI-LOCALES	51

4.6	MÉTODO DE BOLSA DE PALABRAS VISUALES (BAG OF VISUAL WORDS – BOVW)	52
4.6.1	Diccionarios Visuales	52
4.6.2	Cuantificación de Características	54
4.6.3	Selección de la Representación	54
4.6.4	Comparación de Imágenes basada en su Representación	55
4.7	MÉTODO DE GRAFOS VISUALES	57
4.7.1	Bolsa de Grafos Visuales (BoVG)	58
4.7.1.1	Libro de Códigos de Palabras Visuales	59
4.7.1.2	Libro de Código basado en Grafos	60
4.7.2	Creación de un Libro de Códigos basado en Grafos	61
4.7.3	Creación de una Bolsa de Grafos Visuales	63
4.8	MÁQUINAS DE VECTORES DE SOPORTE	63
4.8.1	Definición de Máquina de Vector de Soporte	64
4.8.1.1	Errores de Entrenamiento	66
4.8.1.2	Función Kernel	66
4.8.1.3	SUR Regresión	68
4.8.1.4	Comparación entre Redes Neuronales Artificiales y las Máquinas de Vector de Soporte	69
5.	METODOLOGÍA	70
5.1	ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES	70
5.2	ESQUEMA DE DETECCIÓN DE OBJETOS	72
5.3	ESQUEMA DE BÚSQUEDA	73
5.4	SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES UTILIZANDO LA TÉCNICA DE BOLSA DE PALABRAS VISUALES (BoVW)	74
6.	RESULTADOS OBTENIDOS DEL SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES	79
6.1	ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES	80

6.1.1	Ejemplo de Localización de Puntos de Interés	80
6.1.2	Ejemplo de Correspondencia de Puntos Característicos	84
6.1.3	Ejemplo de Detección de Esquinas	86
6.2	ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES	87
6.2.1	Ejemplos de comparación de imágenes a partir de la coincidencia de sus puntos de Interés	87
6.2.1.1	Caso 1: Comparación de la misma Imagen	89
6.2.1.2	Caso 2: Comparación de dos imágenes con cambio de perspectiva	90
6.2.1.3	Caso 3: Comparación de dos imágenes con cambio de escala	92
6.2.1.4	Caso 4: Comparación de dos imágenes con cambio de escala y perspectiva	94
6.2.1.5	Caso 5: Comparación de dos imágenes con alto cambio de perspectiva	95
6.2.1.6	Caso 6: Localización de un objeto que hace parte de una imagen	97
6.2.2	Conclusiones del esquema de detección de puntos de interés y búsqueda de coincidencias entre imágenes	118
6.3	ESQUEMA DE BÚSQUEDA	119
6.3.1	Esquema de búsqueda de objetos específicos en un conjunto de imágenes a partir de su grado de coincidencia	119
6.3.1.1	Ejemplo 1	119
6.3.1.2	Ejemplo 2	125
6.3.1.3	Ejemplo 3	130
6.3.1.4	Ejemplo 4	135
6.3.2	Conclusiones del esquema de búsqueda	140
6.4	SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES	140
6.4.1	Sistema de clasificación de imágenes utilizando bolsa de características personalizada	140
6.4.1.1	Ejemplo 1	141
6.4.1.2	Ejemplo 2	147

6.4.1.3	Conclusiones del sistema de clasificación y reconocimiento de imágenes	152
6.4.2	Sistema de clasificación de imágenes utilizando bolsa de palabras visuales (BoVW)	152
6.4.2.1	Procedimiento del sistema de recuperación de imágenes utilizando bolsa de palabras visuales	153
6.4.2.2	Sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales, máquinas de vector de soporte (SVM) y descriptores SURF y SIFT	154
7.	CONCLUSIONES	174
8.	RECOMENDACIONES	
9.	BIBLIOGRAFÍA	

RESUMEN

Mediante descriptores, se pueden definir los puntos clave que caracterizan una imagen cualquiera, los cuales luego podrán ser localizados en otras escenas en las que existen rotaciones, cambios de escala e iluminación y occlusiones parciales. De esta forma se podrá realizar la búsqueda automática de objetos en distintas imágenes. Durante el desarrollo de este proyecto de grado, se realizará un estudio de diferentes métodos de extracción de características de las imágenes y la utilización de dichos métodos en la implementación de un sistema de clasificación y reconocimiento de imágenes.

Para la implementación del sistema de clasificación y reconocimiento de imágenes utilizando la técnica de Bolsa de Palabras Visuales (BofVW), máquinas de vector de soporte y descriptores, inicialmente se parte de la implementación de diversas técnicas para hallar puntos de interés y descriptores sobre algunas imágenes de la base de datos de imágenes levantada por el autor. En segunda instancia se implementaron varios esquemas de clasificación y reconocimiento aplicando los descriptores SIFT y SURF, y realizando la comparación de puntos de interés entre las imágenes para hallar las coincidencias(**Esquema de detección de puntos de interés y búsqueda de coincidencias entre imágenes**); Primero se hizo la comparación entre dos imágenes y luego una imagen contra un conjunto de imágenes de una base de datos(**Esquema de búsqueda de objetos específicos, en un conjunto de imágenes a partir de su grado de coincidencia**). Luego se implementó el sistema de clasificación y reconocimiento de imágenes utilizando la bolsa de palabras visuales (BoVW).

Para esta fase del proyecto se implementó el sistema de clasificación de imágenes utilizando bolsa de características personalizada. Sistemas CBIR (CBIR- Sistemas basados en contenido para la recuperación de imágenes) y el sistema de clasificación de imágenes utilizando bolsa de palabras visuales, máquinas de vector de soporte y descriptores (SIFT y SURF).

Palabras clave: Reconocimiento de imágenes, SIFT, SURF, VL_FEAT, Codebook, *Bag-of-Visual Words*, k-means, Support Vector Machine (SVM).

INTRODUCCIÓN

En muchas aplicaciones reales relacionadas con imágenes captadas de un entorno de trabajo, se hace necesario aplicar métodos que permitan clasificar y reconocer imágenes a partir de la clase de objetos encontrados en éstas.

A pesar de que los resultados obtenidos en reconocimiento y clasificación de imágenes son abundantes y muy significativos; debido a las grandes variaciones de la apariencia de las imágenes, como la escala, la iluminación, la pose y las características de fondo, aún existen problemas por resolver relacionados con la forma de mostrar e interpretar de manera razonable los objetos en una imagen.

Diferentes métodos de clasificación y reconocimiento varían mucho en detalles como: la forma de detección, representación, variación en el aspecto (si es parcial o total) y si la posición se representa de manera explícita o implícita. Los algoritmos de coincidencia y los procedimientos de aprendizaje están poco normalizados; la mayoría de los autores se basan en pasos manuales para eliminar el fondo y el desorden, y normalizar la posición de los conjuntos de entrenamiento; además, el reconocimiento pasa a menudo por una exhaustiva búsqueda sobre la posición de la imagen y la escala.

La clasificación de patrones por asignación de una o varias etiquetas a una imagen con base en su contenido semántico, también se puede definir como un modelo de reconocimiento de imágenes.

Un planteamiento muy frecuente es la modelación de la distribución de características de bajo nivel contenidas en las imágenes, sin tener en cuenta las posiciones absolutas o relativas de dichas características. La recuperación de imágenes basada en su contenido (no semántico) mejora significativamente la calidad de las búsquedas. Para ello, es necesario concebir modelos que clasifiquen imágenes a partir de las características extraídas de éstas.

Dado que el reconocimiento de imágenes es un problema complejo de solucionar, existen múltiples algoritmos y grupos de investigación, dedicados a estudiar este problema y a dar posibles soluciones.

Las diferentes soluciones se dan desde diversas estrategias abordadas para la solución del problema que tienen en cuenta algún tópico específico, por ejemplo:

algunas soluciones se dan abordando la descripción de las imágenes en base a su contenido; esta descripción se realiza normalmente mediante características de bajo nivel como color, textura, bordes, etc., pero dada la alta dimensionalidad de estas características, se hace necesaria la reducción de la dimensión de las características extraídas.

Otro método de reconocimiento se basa en la clasificación de las imágenes mediante el uso de determinadas técnicas. Una de las técnicas más utilizadas para el reconocimiento y clasificación de imágenes a partir de objetos es la llamada *Bag-of-Visual Words (BoVW)*. Esta técnica consiste en la generación de diccionarios visuales o “codebook” con las “palabras visuales” que aparecen en las imágenes y la respectiva caracterización de éstas, mediante un histograma en el que se represente el número de ocurrencias de los términos visuales iguales dentro de la imagen.

La caracterización de la imagen se puede realizar de diferentes maneras. Entre los extractores de características más utilizados se pueden mencionar: el descriptor SIFT [1], el descriptor SURF, aunque también se utilizan detectores de bordes y esquinas [2], el detector de SUSAN [3], Geometric-Blur (GB) [4], etc.

Debido a que el número de características diferentes extraídas con estos métodos es grande, se suelen cuantificar para generar así “codebooks” más pequeños con los que se puede obtener representaciones más compactas de las imágenes. El problema de la cuantificación es la pérdida de información que puede ser relevante para la solución. Existen métodos de aprendizaje que intentan compensar este problema. El método más utilizado es el algoritmo k-means [5] aunque también se utilizan otros algoritmos como el Modelo de Mezcla de Gaussianas (GMM) [6], *mean-shift* [7], o árboles de decisión [8].

Una vez generados estos “codebooks”, se procede a la clasificación de las imágenes con métodos de aprendizaje automático como SVM (*Support Vector Machine*) [9], K-NN (*K-Nearest Neighbor*) [10] o el clasificador pLSA (*probabilistic Latent Semantic Analysis*) [10, 11]. Otro método para la clasificación de imágenes es el llamado NBNN [9], que a diferencia del modelo BoVW, no requiere de una fase previa de entrenamiento/aprendizaje mediante algoritmos como k-means o GMM. Este método de clasificación detecta el objeto mediante un cálculo de distancias entre objetos la clase más cercana (la más similar) en la base de datos. Otro clasificador que proporciona buenos resultados en la detección de objetos es el llamado *Stacked R. Forest* [12].

Una vez clasificadas las imágenes, se procede finalmente a la evaluación del clasificador, normalmente mediante el uso de gráficas de Precision/Recall o curvas ROC, calculando la Precisión Media del clasificador y el Área bajo la Curva, en inglés “*Area Under Curve*” (AUC) de la curva ROC.

Definiremos a continuación algunos conceptos necesarios en el reconocimiento de imágenes, como son los descriptores de características y las técnicas de clasificación de las imágenes.

1. FORMULACIÓN DEL PROBLEMA

¿Cómo implementar un sistema que permita clasificar y reconocer imágenes a partir de la interpretación de los objetos contenidos en éstas?

En muchas aplicaciones reales relacionadas con imágenes captadas de un entorno de trabajo, se hace necesario aplicar métodos que permitan clasificar y reconocer imágenes a partir de la clase de objetos encontrados en éstas.

A pesar de que los resultados obtenidos en reconocimiento y clasificación de imágenes son abundantes y muy significativos; debido a las grandes variaciones de la apariencia de las imágenes, como la escala, la iluminación, la pose y las características de fondo, aún existen problemas por resolver relacionados con la forma de mostrar e interpretar de manera razonable los objetos en una imagen.

Diferentes métodos de clasificación y reconocimiento varían mucho en detalles cómo la forma de detección, representación, variación en el aspecto (si es parcial o total) y si la posición se representa de manera explícita o implícita. Los algoritmos de coincidencia y los procedimientos de aprendizaje están poco normalizados; la mayoría de los autores se basan en pasos manuales para eliminar el fondo y el desorden, y normalizar la posición de los conjuntos de entrenamiento; además, el reconocimiento pasa a menudo por una exhaustiva búsqueda sobre la posición de la imagen y la escala.

Una metodología eficiente de reconocimiento y clasificación de imágenes debe abordar aspectos, como:

- Creación de representaciones realmente discriminantes y precisas, es decir muy pequeñas diferencias entre las imágenes u objetos deben ser codificadas, sin dejar de ser válidas para especificar transformaciones geométricas relacionadas con el dominio.
- Construcción de topologías e implementación de algoritmos que sean adecuados para representar los modelos de los objetos de las imágenes de manera eficiente, que sean lo suficientemente flexibles para adaptarse a la variabilidad del objeto (cambio de aspecto debido a condiciones de iluminación, cambio de perspectiva, etc).

- Implementación de técnicas eficientes de agrupamiento, correspondencia y clasificación, que mejoren los procesos de aprendizaje, reconocimiento y clasificación de las imágenes. De aquí que los conjuntos de entrenamiento deben ser pequeños y los algoritmos de aprendizaje óptimos, para obtener sistemas de clasificación eficientes y con reducido consumo de tiempo de computo.

2. JUSTIFICACIÓN

Visión por computador es el estudio y la aplicación de métodos que permiten a los ordenadores entender el contenido de las imágenes. El término “entender” significa extraer la información específica a partir de los datos que proporciona la imagen para un fin específico. Dicha información sirve para que sea interpretada por un operador humano (por ejemplo, la detección de células cancerígenas en una imagen microscópica) o para controlar algún proceso (por ejemplo, un robot en la industria realizando una tarea de clasificación y selección de partes en una cadena de producción o un vehículo autónomo desplazándose por un entorno de trabajo a partir del seguimiento de marquillas visuales distintivas).

Aunque existen numerosos sistemas dedicados a clasificar y reconocer imágenes, todavía presentan falencias a la hora de identificar aquellas imágenes que no se encuentran en un formato preestablecido o bajo ambientes con características controladas tales como la escala, la iluminación, la pose y el fondo. En la actualidad, la mayoría de los sistemas requieren de pasos manuales para eliminar el fondo, el desorden y normalizar la posición de los conjuntos de imágenes de entrenamiento previo a su clasificación.

Por medio del desarrollo de este trabajo de grado, se pretende hacer un sistema de clasificación y reconocimiento de imágenes que contenga los siguientes aspectos:

- La creación de representaciones realmente discriminantes y precisas, es decir, muy pequeñas diferencias entre las imágenes u objetos deben ser codificadas, sin dejar de ser válidas para especificar transformaciones geométricas relacionadas con el dominio.
- La construcción de topologías e implementación de algoritmos que sean adecuados para representar los modelos de los objetos de las imágenes de manera eficiente, que sean lo suficientemente flexibles para adaptarse a la variabilidad del objeto (cambio de aspecto debido a condiciones de iluminación, cambio de perspectiva, etc.).
- La implementación de técnicas eficientes de agrupamiento, correspondencia y clasificación, que mejoren los procesos de aprendizaje, reconocimiento y clasificación de las imágenes. De aquí que los conjuntos de entrenamiento deben

ser pequeños y los algoritmos de aprendizaje óptimos, para obtener sistemas de clasificación eficientes y con reducido consumo de tiempo de computo.

Además, este trabajo de grado también servirá como apoyo a los proyectos de investigación formales que se desarrollan por el grupo de investigación en Robótica y Percepción Sensorial- GIROPS, en su línea de “*Percepción Sensorial*”, buscando integrar dicho proyecto a otros relacionados con robótica móvil, como navegación de robots móviles y SLAM. Si bien este tipo de trabajos, viene siendo desarrollando por diferentes grupos de investigación y la comunidad de visión por computador va en aumento, el desarrollo de aplicaciones funcionales de bajo costo es algo que permite el desarrollo de prototipos que pueden ser utilizados en la industria para la solución de diferentes tipos de problemas.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Implementar un sistema de clasificación y reconocimiento de imágenes a partir de objetos utilizando las técnicas de Bolsa de Palabras Visuales (BoVW) y Máquinas de Vector de Soporte (SVM).

OBJETIVOS ESPECÍFICOS

- Utilizar descriptores para obtener puntos de interés generando los vectores característicos en las imágenes de estudio.
- Aplicar métodos de agrupamiento para obtener palabras visuales de los objetos de las imágenes de estudio y emplear la técnica de bolsa de palabras visuales para el reconocimiento de objetos en las imágenes.
- Aplicar algoritmos de clasificación de objetos en las imágenes para su reconocimiento.

4. MARCO TEÓRICO

La representación, la detección y el aprendizaje son temas necesarios de abordar en el diseño de un sistema visual para el reconocimiento de objetos por categorías. El primer desafío es la implementación de modelos que pueden capturar la "esencia" de una categoría, es decir, lo que es común a los objetos que pertenecen a la misma, y sin embargo, que sean lo suficientemente flexibles para adaptarse a la variabilidad del objeto.

La técnica popular y eficiente de representación de imágenes para categorización de objetos BoVW ha utilizado métodos estadísticos para determinar las características locales a partir de la clasificación de puntos de interés basados en técnicas como SIFT y SURF [1,13].

Los puntos de interés definen las características locales y la mayoría de los métodos utilizados buscan acelerar la extracción de puntos de interés.

El método de BoVW [14] busca cuantificar las características en diccionarios visuales por medio de la agrupación y la reconstrucción de las imágenes a través de la distribución de las palabras visuales dentro de ellas. Este método opera en características visuales locales en las imágenes y en palabras con puntos de interés.

4.1 RECONOCIMIENTO DE OBJETOS

El reconocimiento de objetos es una metodología a partir de la cual se puede hacer la comparación de sectores o porciones que sean invariantes a cambios en la escala, la orientación y la iluminación entre imágenes para conformar la semántica de éstas y obtener su reconocimiento.

El reconocimiento de objetos se puede dividir en dos grupos:

4.1.1 Reconocimiento de Instancias. Implica reconocer un objeto previamente conocido dentro de una imagen, cuando es observado desde diferentes puntos de vista incluyendo la presencia de objetos extraños e incluso occlusiones.

4.1.2 Reconocimiento de Categorías. Consiste en reconocer un objeto que no ha sido visto y asignarle una categoría (por ejemplo “es un auto”, “es una persona”).

El reconocimiento de objetos mediante imágenes a color se viene investigando desde hace varios años. Algunos de los enfoques que se han desarrollado a lo largo del tiempo, son:

- ✓ Detección de líneas, contornos y/o superficies para luego compararlas con modelos 2D o 3D.
- ✓ Adquisición de imágenes desde diversas posiciones y orientaciones para representarlas en un espacio vectorial y realizar una descomposición en una base de datos con los valores propios más importantes (Ejemplo: Eigenfaces [15]).
- ✓ Extracción de un conjunto de características locales que tengan propiedades de invarianza frente a cambios de iluminación y punto de vista. Se reconoce una imagen comparando estas características locales con las características de las imágenes de la base conocida (debe de haber una suficiente cantidad de correspondencias y esas correspondencias deben ser coherentes con una transformación que alinee las imágenes). (Ejemplo: Matching con SIFT [16]).

Cuando la cantidad de imágenes consideradas crece, no es posible la comparación 1 a N. Las características locales son mapeadas a un conjunto de “palabras visuales”. Estas “palabras visuales” se pueden aprender mediante k-means sobre un conjunto de entrenamiento. El reconocimiento de una nueva imagen se realiza comparando las palabras visuales contra la frecuencia de aparición de éstas. Esto da un ranking de candidatos que se puede refinar teniendo en cuenta la coherencia espacial de correspondencias [17,18].

4.2 EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características en una imagen consiste en la ejecución de una metodología en la cual la imagen o las áreas de interés se definen con la información más apropiada por medio de descriptores, los cuales pueden ser: Locales, globales o semi-locales. Posteriormente se hace una representación compacta de todo el

conjunto de descriptores existentes y por último, las distancias o similitudes entre estas representaciones se calculan de manera que la imagen actual pueda ser clasificada o comparada con una base de datos que permita obtener el resultado de reconocimiento.

4.3 DESCRIPTORES GLOBALES DE IMÁGENES

El color es una parte importante de la percepción humana, por tal razón, las características globales de las imágenes se basan generalmente en señales de color.

En las imágenes, los colores son codificados en espacios de color. Un espacio de color es un modelo matemático que permite la representación de los colores, como una tupla de componentes de color. Se puede citar el espacio RGB (Red Green Blue), el HSV (Matiz Saturación Value) o los espacios de luminancia-crominancia (YUV) como ejemplos.

Probablemente el más famoso descriptor global de color es el histograma de color. Los Histogramas de color tienen por objeto representar la distribución de colores dentro de la imagen o en una región de la imagen. Cada bin de un histograma representa la frecuencia de un valor de color dentro de esta área. Por lo general, los histogramas se basan en una cuantificación de los valores de color, los cuales pueden ser diferentes de un canal de color a otro. Los histogramas son invariantes a transformaciones geométricas de la región.

Los momentos de color son otra forma de representación de la distribución del color en una imagen o una región de la imagen. Utilizando los momentos de color, una distribución del color puede ser representada de una manera muy compacta [19].

Otros descriptores de color que se pueden mencionar son los denominados descriptores dominantes (DCD) introducidos en el estándar MPEG-7 [19] o en el Descriptor de capa de color (CLD).

4.4 DESCRIPTORES LOCALES DE IMAGEN

Las características más llamativas de los últimos años son las denominadas: Características Locales. La idea fundamental de estas características es centrarse en las áreas que contienen la mayor información discriminante. En particular los descriptores son calculados alrededor de los puntos de interés de la imagen y son asociados a los detectores de un punto de interés.

4.4.1 Puntos de Interés. Los puntos y regiones de interés son zonas detectadas automáticamente por cierta familia de algoritmos. Estos puntos corresponden a zonas donde la textura local maximiza algún criterio. Los puntos de interés son, usualmente, máximos o mínimos locales de operadores diferenciales (filtros) aplicados sobre la imagen.

Si la imagen es transformada (rotación, traslación, cambio de tamaño), su textura también se afecta, es decir las mismas zonas siguen maximizando el mismo criterio de textura local.

Un descriptor local es un vector de características calculado sobre una pequeña región de interés de la imagen. Cada región de interés es generada a partir de un punto de interés. Los puntos y regiones de interés que se deforman junto con la imagen su contenido no varían.

La importancia de los descriptores (vectores de características) es que son invariantes, lo cual permite establecer correspondencias entre dos imágenes cuando se comparan aquellos descriptores que son parecidos. Esta característica permite realizar procesos de reconocimiento de objetos.

Los puntos de interés o puntos característicos, son características de bajo nivel de todas las imágenes. Los detectores de puntos de interés tratan de encontrar características como bordes, esquinas, color, etc. Algunos de estos detectores famosos son el detector de esquinas de Harris [20], el detector de puntos característicos SIFT de Lowe [21] y el detector SURF (Speeded-Up Robust Features) [1]. Para la extracción de puntos de interés se escoge el detector que mejor rendimiento presente y se busca que este sea invariante a posibles cambios de escala, traslación, intensidad y orientación.

Después de la detección de características, cada imagen es representada a través de sus características locales. Los métodos de representación tratan de describir estas características como vectores numéricos llamados descriptores de características. El descriptor debe tener la habilidad de manejar la intensidad, rotación, escala y variaciones afines de la misma dimensión.

4.4.1.1 Detector de Esquinas de HARRIS. Un gran número de algoritmos utilizan como referencia para el *matching*, la detección de bordes (*edges*). Aunque estos algoritmos no son sensibles a cambios de intensidad, muestran problemas cuando se presentan otras transformaciones entre imágenes.

En una imagen al encontrar bordes cercanos al umbral de detección, pueden ocurrir grandes cambios en su topología ante pequeños cambios en la intensidad, lo cual puede ocasionar un error en la búsqueda de correspondencias.

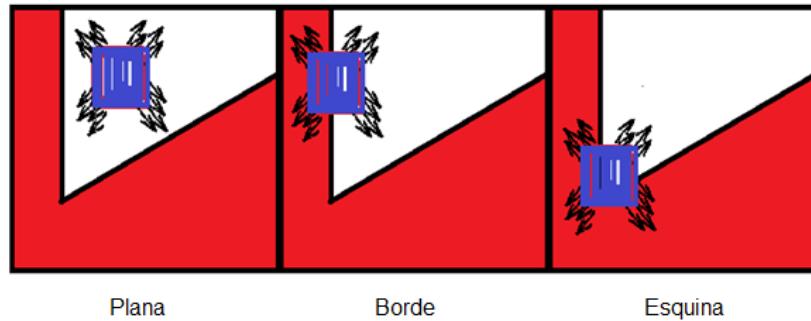
Existen métodos que solucionan la falencia enunciada anteriormente, tal es el caso del **Detector de HARRIS**, el cual consiste en la búsqueda de esquinas. Las esquinas son puntos característicos poco susceptibles a cambios de rotación y escala. Una esquina (*corner*) se caracteriza por ser una región de la imagen con cambios de intensidad en diferentes direcciones. Éste es el principio básico de búsqueda de puntos característicos del **Detector de HARRIS** el cual consiste en filtrar la imagen utilizando una ventana móvil en ocho direcciones para obtener tres tipos de regiones:

- ✓ Plana o *Flat*: No hay cambios en ninguna dirección
- ✓ Borde o *Edge*: No hay cambios en la dirección del propio borde
- ✓ Esquina o *Corner*: Hay cambios significativos en todas direcciones

Una vez detectados los puntos de interés, en este caso las esquinas, se encontrarán las respectivas correspondencias entre estos puntos dentro de las imágenes que se están comparando para su posterior reconocimiento.

En la Figura 1 se muestran los 3 tipos de regiones detectadas en una imagen.

Figura 1. Tipos de Regiones Detectadas



Fuente: Modificación del Autor a [43]

El **Detector de HARRIS** consta de los siguientes pasos:

1. Búsqueda de Puntos de Interés: Este paso consiste en ubicar dentro de la imagen aquellos puntos de interés que se definen como esquinas. Esta ubicación se obtiene por medio de la *Matriz de Autocorrelación* M . Esta matriz de autocorrelación de la imagen es de 2×2 y se obtiene aplicando la derivada horizontal y vertical de primer orden con plantillas 1×3 y 3×1 , respectivamente.

$$[-1 \quad 0 \quad 1] ; \quad \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Los elementos de la matriz de autocorrelación se calculan por medio de la ecuación 1.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (1)$$

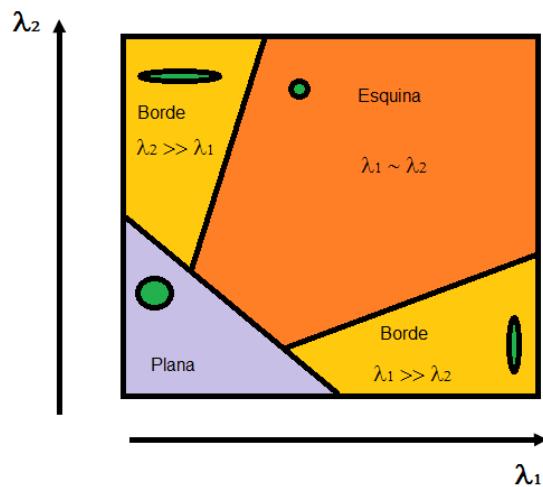
donde:

$$\begin{aligned} A &= \left(\frac{\partial I}{\partial x} \right)^2 \otimes w = X^2 \otimes w \\ B &= \left(\frac{\partial I}{\partial y} \right)^2 \otimes w = Y^2 \otimes w \\ C &= (X * Y) \otimes w \end{aligned} \quad (2)$$

Los elementos de la matriz de autocorrelación se obtienen al elevar al cuadrado las derivadas parciales y posteriormente filtrar como se puede apreciar en la ecuación 2. Se utiliza el filtro W de tipo gaussiano para evitar que la respuesta sea ruidosa, tal y como pasaría con un filtro rectangular.

Este filtro tendrá un factor de filtrado $\sigma = 2$ y un tamaño de ventana de seis veces el factor de filtrado (tamaño: $6 \times \sigma = 12$). Si se definen $\Delta 1$ y $\Delta 2$ como los valores propios de la matriz M calculada por medio de la ecuación 1, se podrán obtener los tres tipos de regiones como se muestra en la Figura 2.

Figura 2. Regiones en función de valores propios de M



Fuente: Modificación del Autor a [43]

De la Figura 2 se puede observar lo siguiente:

- ✓ Si ambos valores son pequeños, indica que la función de autocorrelación es plana, por tanto la zona de la imagen tiene una intensidad aproximadamente constante: Plano (*Flat*).
- ✓ Si uno de los valores es pequeño y otro es elevado, la función de autocorrelación tendrá un cierto rizado: Borde (*Edge*).
- ✓ Si los dos valores son elevados, en la función se observarán picos Bruscos: Esquina (*Corner*).

Se puede justificar el motivo de esta clasificación a partir de los valores propios. $\Delta 1$ y $\Delta 2$ reflejan los modos de variación de las direcciones principales de los gradientes. A partir de esta definición, cuando en una región de la imagen los dos valores son elevados, se deduce que localmente existen dos direcciones importantes de los gradientes y se concluye en que es una esquina.

2. Mapa de Esquinas: Una vez calculados los valores propios de M , se procede a construir el mapa de esquinas. Para ello, se mide la respuesta a éstas a partir de la traza y el determinante de la función M , los cuales se calculan por medio de las ecuaciones 3 y 4 respectivamente.

$$Tr(M) = A + B \quad (3)$$

$$Det(M) = A * B - C^2 \quad (4)$$

La respuesta a las esquinas R , se calcula por medio de la ecuación 5.

$$R = Det(M) - k * (Tr(M))^2 \quad (5)$$

donde k es una constante arbitraria obtenida empíricamente. En este caso se utiliza el valor de $k = 0,04$ como estándar.

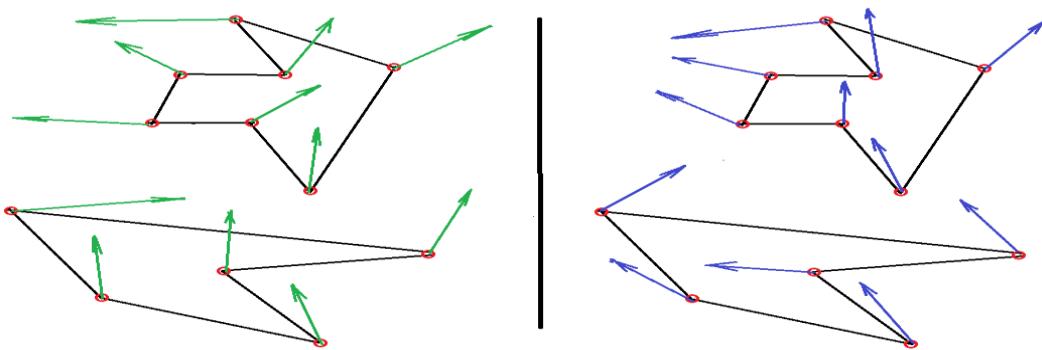
3. Supresión de Valores que no son Máximos: En esta fase de la búsqueda sobre la imagen, ya no se buscan más puntos, sino que se descartan varios de los puntos obtenidos anteriormente.

El primer paso para la supresión de valores es definir un umbral para la función R por encima de un cierto valor y así descartar varios de los píxeles que aparecen marcados como posibles esquinas. Cuanto mayor sea este valor, más restrictivo será el detector en cuanto al número de esquinas detectadas, aunque aumentará su fiabilidad.

Para evitar múltiples detecciones en una misma esquina (nubes de puntos negros), se utiliza el denominado filtro *non-maximal suppression*, el cual se encarga de eliminar todos los puntos en los cuales la dirección del gradiente no sea la máxima en un entorno local.

4. Cálculo de Correspondencias (*matching*). El paso posterior a la búsqueda de puntos característicos que proporciona el detector, es la comparación de descriptores entre pares de imágenes y la búsqueda de correspondencias entre ellas como se puede apreciar en la Figura 3.

Figura 3. Detección de Esquinas y Cálculo de Correspondencia



Fuente: Modificación del Autor a [43]

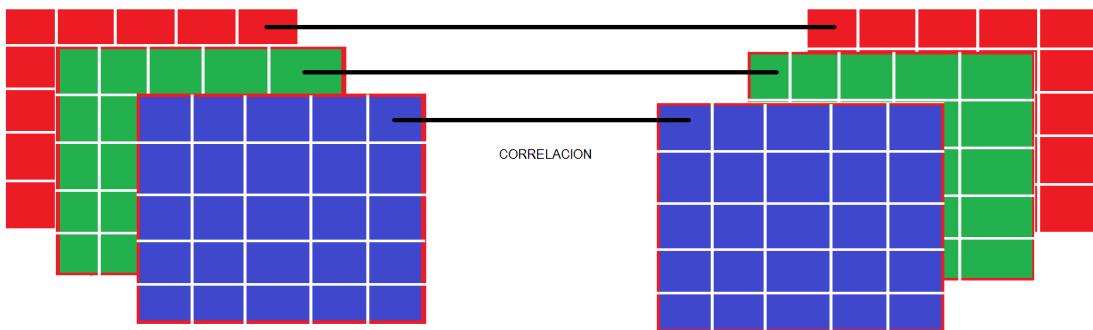
En la Figura 3 se pueden observar dos imágenes con sus esquinas detectadas y marcadas en color rojo y sobre ellas su gradiente correspondiente.

Estos métodos de detección de puntos de interés se encargan de describir la zona alrededor del punto, pero difieren en cómo buscar las correspondencias entre las imágenes.

Un método simple de implementar es la evaluación de las matrices que aportan información del color (RGB), en ventanas $x \times y$ alrededor del punto característico, donde a todos los puntos de la imagen que el detector de *HARRIS* marca como esquinas, se les calcula su gradiente, con lo cual se obtiene la localización y la dirección del cambio máximo, para así obtener invarianza a la orientación.

En la Figura 4 se muestra la correspondencia entre los píxeles de color de dos imágenes.

Figura 4. Esquema de Cálculo de Similitud entre Descriptores



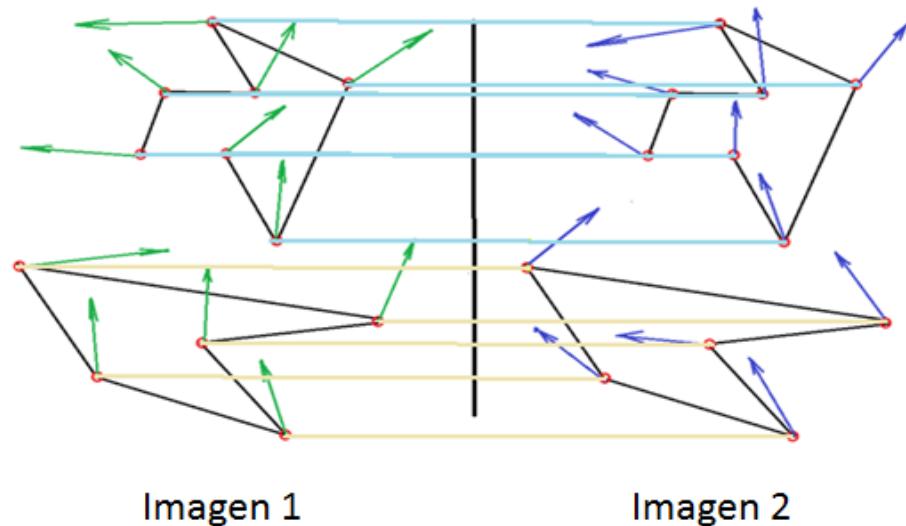
Fuente: Modificación del Autor a [43]

5. Correlación entre Descriptores. Para la correlación, se define una ventana de $x \times y$ píxeles alrededor de cada esquina en función de la orientación del gradiente de cada punto descrito en la imagen. Para el caso de una imagen a color, ya que ésta se compone de tres matrices (R, G y B), en cada esquina se definirán tres ventanas $x \times y$.

El siguiente paso de la correlación es comparar mediante cálculo de correlaciones, los descriptores de las dos imágenes (trío de ventanas $x \times y$), ya que se intentarán asociar los puntos que tengan descriptores más parecidos. Éstos serán los que sean matemáticamente similares, es decir, con una correlación máxima. La comparación se realizará entre TODOS los descriptores obtenidos en ambas imágenes. El proceso se termina con la relación de puntos de la imagen 1 con sus correspondientes en la imagen 2.

En la Figura 5 se muestra la correspondencia entre los puntos de la imagen 1 con los puntos de la imagen 2. Éstos serán los que tienen una correlación máxima.

Figura 5. Correspondencia final entre puntos similares de la imagen 1 y la imagen 2



Fuente: Modificación del Autor a [43]

El proceso de realizar las correlaciones entre las matrices, incrementa el costo computacional y el tiempo de procesado de las imágenes.

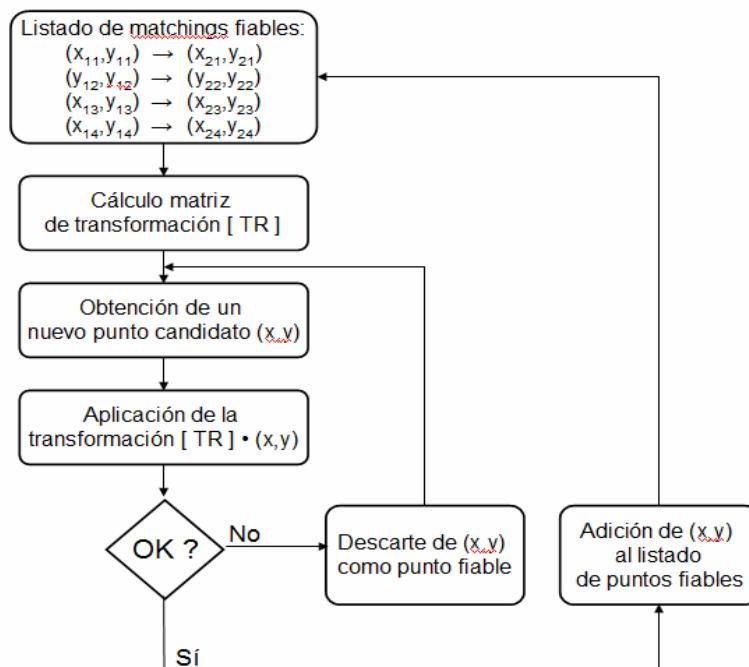
Las comparaciones permiten obtener el punto más similar de la segunda imagen, es decir, un punto de la imagen 1 puede tener su correspondiente en la imagen 2 con un grado de semejanza del 97% (correlación entre ventanas igual a 0,95), mientras que otro punto puede tener el suyo al 50%. En este último caso, es posible que no se correspondan. Esto no significa que no se produzcan fallos en las correspondencias (*Matching*), sino que el proceso se hace más preciso.

Al aplicar a una imagen transformaciones espaciales como traslaciones o rotaciones, los descriptores no sufren grandes cambios (simplemente rotan, si es el caso), pero ante cambios de escala, los descriptores varían considerablemente (ya que el tamaño del objeto es diferente). Por ese motivo, este método es sensible a cambios de escala.

6. Cálculo de Transformación entre Puntos. De todas las correspondencias, se pueden seleccionar las cuatro ‘más fiables’ (aquellas que tienen un grado de similitud por encima del 90%), con lo cual se puede garantizar que esas cuatro correspondencias son correctas.

La esencia de este método es calcular la transformación espacial que se ha producido entre los cuatro puntos de la primera imagen y los cuatro de la segunda. Una vez calculada la transformación TR, se aplica de nuevo el método a un nuevo punto P de la imagen 1. Si la correspondencia obtenida en la imagen 2 es correcta, se añade este punto P a la lista de puntos fiables que ahora pasarán de ser cuatro a ser cinco. Este proceso se repetirá sucesivamente hasta obtener la función óptima TR que transforma una imagen en otra. En la Figura 6 se muestra el diagrama de flujo del algoritmo de comparaciones.

Figura 6. Diagrama de Flujo del Cálculo de la Transformación



Fuente: [43]

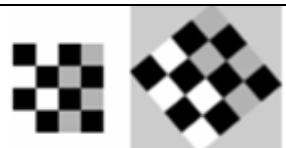
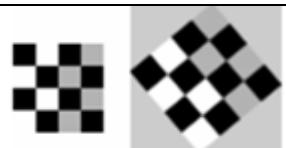
Utilizando este método, a diferencia del método de correlación de descriptores, se evitan de manera significativa errores en el *matching*. Esto es así debido a que los

puntos que no ‘caen’ en el lugar correcto son excluidos y no toman parte en el cálculo de la transformación espacial.

Sin embargo, la gran desventaja es que sólo funciona correctamente si las primeras correspondencias (a partir de las cuales se obtiene TR0) son válidas. La solución a este problema es que el algoritmo sea capaz de decidir si realmente los cuatro primeros puntos iniciales son correctos a medida que avanza la ejecución del código. Es decir, cuando se observa que la mayoría de los puntos siguientes está fallando, hay que ser capaz de descartar los iniciales y escoger otros cuatro.

Existen diferentes modelos para describir los movimientos 2D (Ver Cuadro 1). En función de la complejidad de la transformación, se necesita un mayor o menor número de parejas de puntos iniciales para poder calcularla.

Cuadro 1. Tipos de Transformación Espacial 2D

Tipo de Transformación	Mínimo Número de Puntos	Ejemplo
Traslación Rotación Escala	2	 
Afín	3	 
Proyectiva	4	 
Polinómica	6	 

Fuente: [43]

4.4.2 Detector SIFT (*Scale-Invariant Feature Transform*). SIFT- *Scale-Invariant Feature Transform* [18 BaderAudeh] es un algoritmo de visión artificial que permite detectar y posteriormente describir características en regiones locales de una

imagen que sean invariantes a la escala, orientación, parcialmente a cambios de iluminación, etc. También se puede utilizar para buscar correspondencias entre diferentes puntos de vista de una misma escena.

Estas características locales se almacenan en los denominados descriptores, los cuales describen localmente determinadas variables de zonas importantes de la imagen como el gradiente.

A diferencia del detector de HARRIS, SIFT es más complejo, por lo tanto tiene un coste computacional mayor. Sin embargo, una correcta implementación del algoritmo puede ser utilizada en una aplicación en tiempo real, siempre y cuando la base de datos de búsqueda o entrenamiento no sea muy extensa. Esto se debe principalmente a que las operaciones con un mayor coste se aplican sólo a las localizaciones que han pasado un test o filtro inicial.

El algoritmo está estructurado en cuatro etapas bien diferenciadas:

- ✓ Scale-space extrema detection
- ✓ Keypoint localization
- ✓ Orientation assignment
- ✓ Keypoint descriptor

A continuación, se describirán cada una de las etapas:

- **Scale-space extrema detection.** La primera etapa del algoritmo SIFT, es la encargada de buscar un primer conjunto de puntos de interés o puntos claves de la imagen, también llamados *keypoints*. Los puntos de interés o puntos claves son localizaciones y escalas que se repiten continuamente utilizando diferentes vistas del mismo objeto. Para detectar estas localizaciones que son invariantes a diferentes escalas de la imagen, se buscan características estables a través de todas las escalas utilizando una función continua de la escala conocida como espacio de escala (*Función scale-space*).

Función scale-space: $L(x, y, \sigma)$. La búsqueda se realiza sobre todas las localizaciones y todas las escalas de los objetos de la imagen desde diferentes vistas. Para detectar localizaciones invariantes a cambios de escala se utiliza la función continua conocida como scale-space: $L(x, y, \sigma)$.

Para obtener la función $L(x, y, \sigma)$ a partir de la imagen original $I(x, y)$, se utiliza la función gaussiana tal como se muestra en la ecuación 6.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (6)$$

donde el operador $*$ indica la convolución entre la imagen I y la gaussiana G .

La gaussiana G se calcula por medio de la ecuación 7.

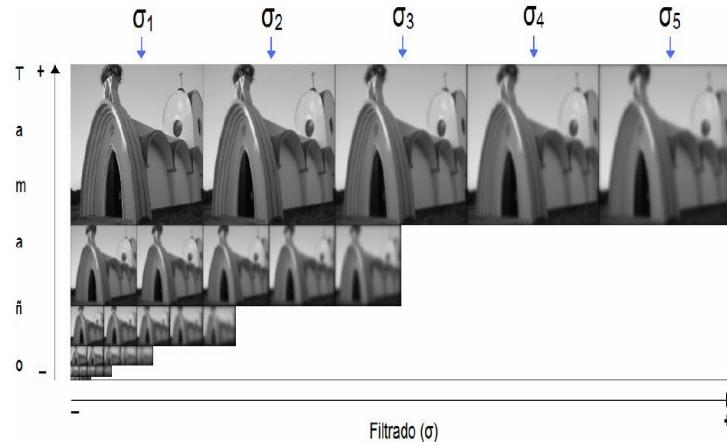
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (7)$$

Para calcular todo el espacio $L(x, y, \sigma)$ se debe construir una pirámide gaussiana convolucionando con diferentes filtros $G(x, y, \sigma)$ y variando el parámetro σ . Para la construcción de esta pirámide es importante tener en cuenta los siguientes conceptos:

- Octava: Conjunto de imágenes del espacio L con el mismo tamaño que difieren en el filtrado σ con el que han sido obtenidas.
- Escala: Conjunto de imágenes del espacio L filtradas con el mismo parámetro σ pero con diferentes tamaños.

En la Figura 7 se muestra un ejemplo de implementación de la pirámide gaussiana para una imagen dada.

Figura 7. Pirámide gaussiana $L(x, y, \sigma)$ compuesta por 5 escalas y 6 octavas



Fuente: [43]

De la Figura 7 se ejecutan los siguientes pasos:

1. Se establece en cinco el número de escalas por octava (por tanto se debe filtrar cuatro veces). El número de octavas total dependerá del tamaño de la imagen original. El factor de escalado entre las diferentes octavas es de $\frac{1}{2}$.
2. Antes de calcular la pirámide L , se realiza un pre-procesado a la imagen original $I(x, y)$, que consiste en suavizarla con un filtro gaussiano de $\sigma_0 = 0,5$ y posteriormente re-escalarla con un factor de 2 mediante interpolación lineal. La imagen resultante, al tener el doble de tamaño, tendrá un valor $\sigma_1 = 1$ y es la que se utiliza como imagen inicial para construir $L(x, y, \sigma)$. Este suavizado previo mejora considerablemente la estabilidad de los *keypoints* que se obtienen más adelante.
3. La condición que tienen que cumplir los diferentes valores de σ es que el penúltimo (σ_4 en este caso) es el doble del primero (σ_1). Por tanto, se divide cada octava en intervalos múltiplos de k como se muestra en las ecuaciones 8 y 9.

$$k = 2^{1/n^{\circ} \text{escalas}-2} = 2^{1/5-2} = 2^{1/3} \quad (8)$$

$$\sigma_i = k^{i-1} = 2^{\frac{i-1}{3}} \quad (9)$$

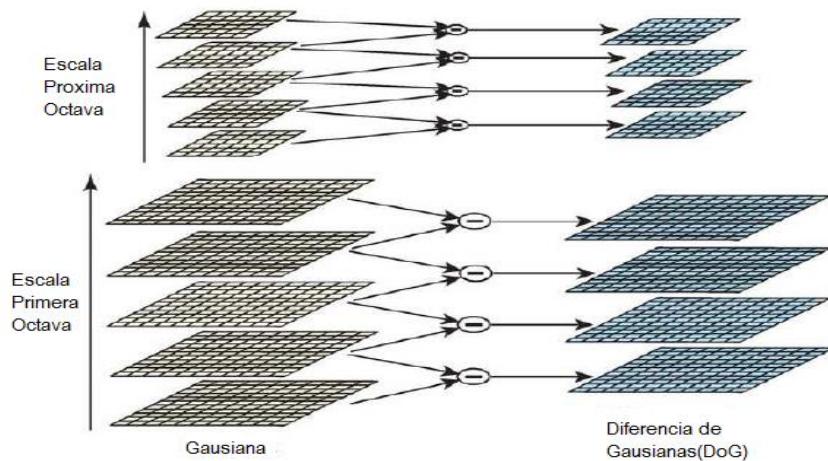
- Una vez completada la primera octava, se toma como referencia la imagen con $\sigma_4 = 2$ como imagen inicial de la siguiente octava, ya que al re-escalarla a la mitad, su factor de filtrado vuelve a ser $\sigma_1 = 1$. Este proceso se va repitiendo hasta completar toda la pirámide.
- Keypoint localization.** Para la detección de los *keypoints* estables en el *scale-space*, no se utiliza la función definida por la ecuación 6, sino que se utiliza una ecuación derivada de ella llamada *Función Difference-of-Gaussian*: $D(x, y, \sigma)$. Esta función puede ser obtenida a partir de las diferencias entre dos escalas vecinas, separadas por un factor k constante y convolucionada con la imagen, la cual se calcula por medio de la ecuación 10.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (10)$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

En la Figura 8 se muestra la construcción de la función de detección por medio de la convolución de la Gaussiana con la diferencia de Gaussianas.

Figura 8. Construcción de la Función de Detección

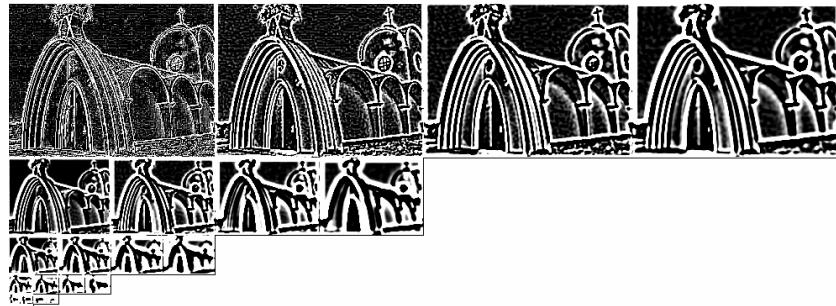


Fuente: [44]

La imagen inicial es convolucionada continuamente con Gaussianas (variando k) para crear un grupo de imágenes (En la parte izquierda de la Figura 8 se puede apreciar la Gaussiana). Las imágenes de las Gaussianas vecinas se restan para producir las imágenes de Diferencia de Gaussianas (DoG), que hay a la derecha de la Figura 8. Luego el mismo proceso se repite, pero variando la escala (s).

Para la imagen de ejemplo de la Figura 7 se pasa a tener cuatro imágenes por octava (Diferencia de Gaussianas-DoG) como se muestra en la Figura 9.

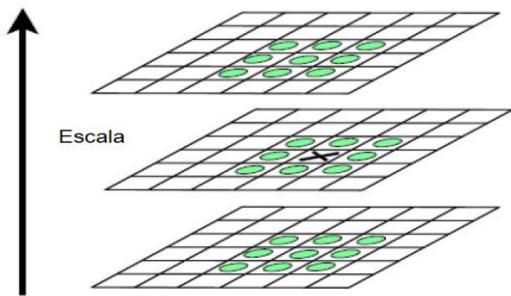
Figura 9. Pirámide Difference-of-Gaussian $D(x, y, \sigma)$



Fuente: [43]

De todos los puntos hallados anteriormente, se eliminan aquellos que no son relevantes, por lo cual a cada punto se le aplica un modelo para saber su localización y escala. A partir de los cálculos anteriores, se calculan los máximos y mínimos locales del espacio $D(x, y, \sigma)$. Todos los píxeles de cada imagen de la pirámide son comparados con sus ocho vecinos de la propia imagen y con los nueve vecinos de la misma imagen en escala anterior y posterior como se puede apreciar en la Figura 10.

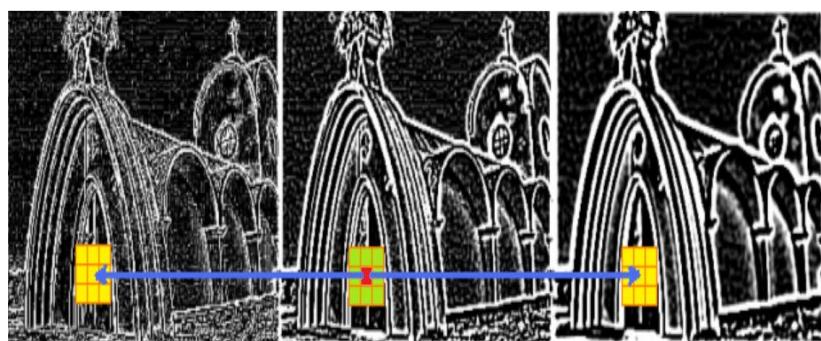
Figura 10. Comparación de Píxeles con sus vecinos



Fuente: [43]

Un punto será seleccionado como *keypoint* sólo si es mayor que sus 26 vecinos o menor que todos ellos. Para la imagen de ejemplo de la Figura 7, se muestra la representación de los píxeles en la imagen por medio de la Figura 11.

Figura 11. Representación de los píxeles en la imagen



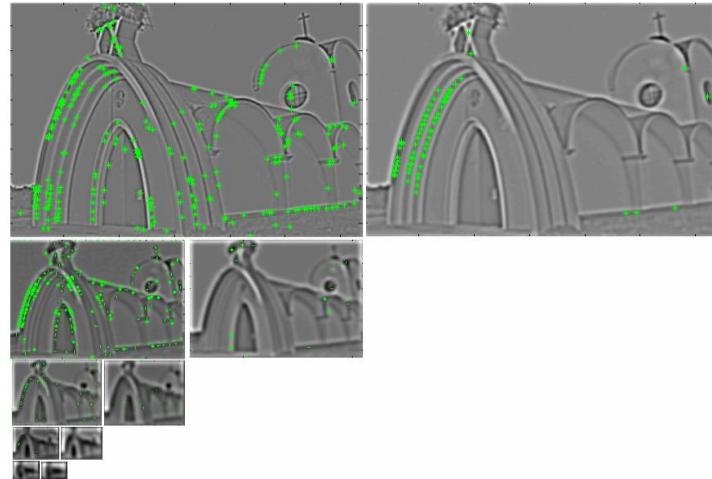
Fuente: [43]

En la Figura 11 se pueden realizar las siguientes apreciaciones:

- En rojo: Píxel en estudio.
- En verde: Vecinos en escala actual.
- En amarillo: Vecinos de escala anterior y posterior.

En la Figura 12, se pueden apreciar los *keypoints* detectados en color verde.

Figura 12. Keypoints detectados en la imagen



Fuente: [43]

La siguiente fase del método SIFT consiste en almacenar toda la información disponible de cada *keypoint*. Para cada punto de interés encontrado se guarda la escala y octava de la pirámide a la que pertenece, además de la posición [fila, columna] dentro de la imagen correspondiente.

Con los datos almacenados se pueden descartar varios puntos, teniendo en cuenta:

- Puntos con bajo contraste
- Puntos localizados a lo largo de bordes

Los puntos descartados por esas dos condiciones no son de interés debido a que serían muy sensibles al ruido.

Para descartar los puntos con bajo contraste se utiliza la ecuación 11.

$$D(p) = D + \frac{\partial D^T}{\partial p} p + \frac{1}{2} p^T \frac{\partial^2 D}{\partial p^2} p \quad (11)$$

donde $p = (x, y, \sigma)^T$

El extremo \hat{p} se determina derivando la expresión e igualándola a cero. De tal manera se obtiene la ecuación 12.

$$\hat{p} = - \left(\frac{\partial^2 D^{-1}}{\partial p^2} \cdot \frac{\partial D}{\partial p} \right) = - \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial y \partial x} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix} \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \end{bmatrix} \quad (12)$$

El vector \hat{p} , resultado de la operación, se define como *offset* del punto. A partir de este *offset*, se calcula el contraste de su *keypoint* correspondiente.

Al sustituir la ecuación 12 en la ecuación 11, se obtiene la ecuación 13.

$$D(\hat{p}) = D + \frac{1}{2} \frac{\partial D^T}{\partial p} \hat{p} \quad (13)$$

La función $D(\hat{p})$ es útil para descartar puntos de bajo contraste estableciendo un umbral mínimo al que deben llegar los *keypoints*.

Ya que la función $D(x, y, \sigma)$ es muy sensible ante los puntos situados sobre los bordes, y dado que la función Diferencia de Gaussianas devuelve muchos puntos clave a lo largo de bordes y esquinas de los objetos que deben eliminarse para mantener la estabilidad de los puntos (extremos). Para ello se calculan las principales curvas a través de una matriz Hessiana H , calculada en la localización y escala de los puntos clave (máximos y mínimos) por medio de la ecuación 14.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (14)$$

Los vectores propios de H (α y β) son proporcionales a la respuesta de D y definen la traza de la matriz Tr que se calcula a través de la ecuación 15.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (15)$$

El determinante de la matriz se obtiene por medio de la ecuación 16.

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2$$

$$Det(H) = \alpha\beta$$
(16)

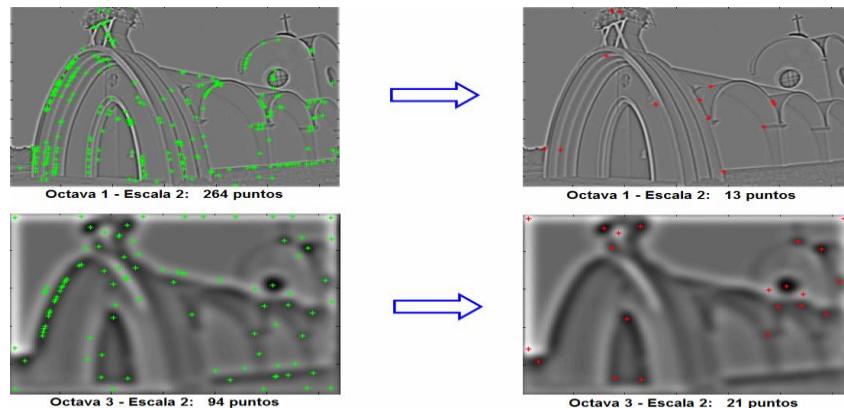
Si el determinante es negativo, las curvas son de diferente signo y por lo tanto el punto se descarta por no ser un extremo.

Si r es la relación entre α y β ($\alpha = r\beta$), entonces, para comprobar si la relación de las principales curvas es menor que un umbral específico r , sólo es necesario comprobar si se cumple la desigualdad 17.

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$
(17)

Cuando los autovalores son iguales, el valor de $\frac{(r+1)^2}{r}$ es mínimo, y va creciendo según va aumentado el valor de la relación r . Si la relación es mínima, se considera que el punto pertenece a un borde. A partir de la imagen del ejemplo Figura 7, se muestran aquellos puntos que son realmente invariantes al cambio por medio de la Figura 13.

Figura 13. Puntos realmente invariantes al cambio



Fuente: [43]

En la Figura 13, los *Keypoints* iniciales se encuentran en color verde y los *Keypoints* no descartados se encuentran en color rojo.

Como se puede apreciar en la Figura 13, un alto porcentaje de puntos que provienen de la fase inicial de SIFT son rechazados; y de esta manera, sólo quedan aquellos puntos los que son realmente invariantes al cambio.

- **Orientation assignment.** Una vez excluidos aquellos puntos que son sensibles al ruido o se encuentran en bordes o esquinas de los objetos, se pasa a asignar una orientación a cada punto característico. Para cada imagen $L(x, y)$ con una determinada escala, se define una región de 16x16 píxeles alrededor del punto y se calcula la magnitud del gradiente $m(x, y)$ y la orientación $\theta(x, y)$ utilizando diferencias entre píxeles.

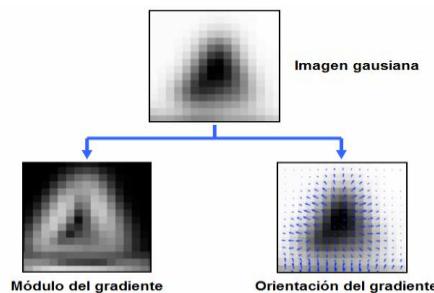
El gradiente m y la orientación θ se pueden calcular mediante las ecuaciones 18 y 19 respectivamente.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (18)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (19)$$

En la Figura 14 se muestra la ventana de 16x16 pixeles alrededor del keypoint.

Figura 14. Ventana de 16x16 pixeles alrededor del keypoint



Fuente: [43]

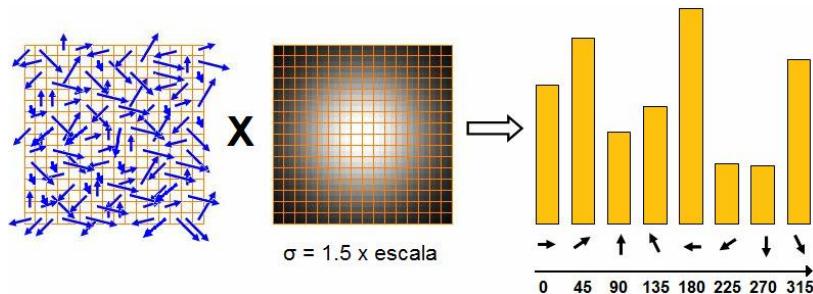
En la Figura 14 se muestra en la parte de arriba una ventana 16×16 alrededor del keypoint, abajo a la izquierda: el módulo de gradiente $m(x, y)$ y abajo a la derecha la orientación de $\theta(x, y)$.

De todo el proceso realizado para el módulo y orientación del gradiente, esta información se toma y se agrupa en forma de histograma, uno para cada *keypoint*. De tal forma, cada histograma de orientaciones está conformado por 36 *bins* para cubrir el rango total de 360° .

A medida que se añade al histograma cada orientación $\theta(x_1, y_1)$ de la región 16×16 , dicho valor se pondera por su módulo $m(x_1, y_1)$ y por una ventana circular gaussiana con valor σ igual a 1,5 veces la escala del *keypoint*.

En la Figura 15 se muestra la región de 16×16 y una ventana circular gaussiana con valor σ igual a 1,5 veces la escala del *keypoint*.

Figura 15. Descripción de orientación de keypoint e histograma



Fuente: [43]

De la Figura 15 se observa a la izquierda, la región de gradientes 16×16 , en el centro la ventana circular gaussiana y a la derecha el histograma final del *keypoint*.

Existen diversos motivos para realizar la ponderación del módulo y del valor σ de la ventana circular gaussiana:

- Dar mayor peso a las orientaciones con módulos elevados, que por tanto son más importantes.

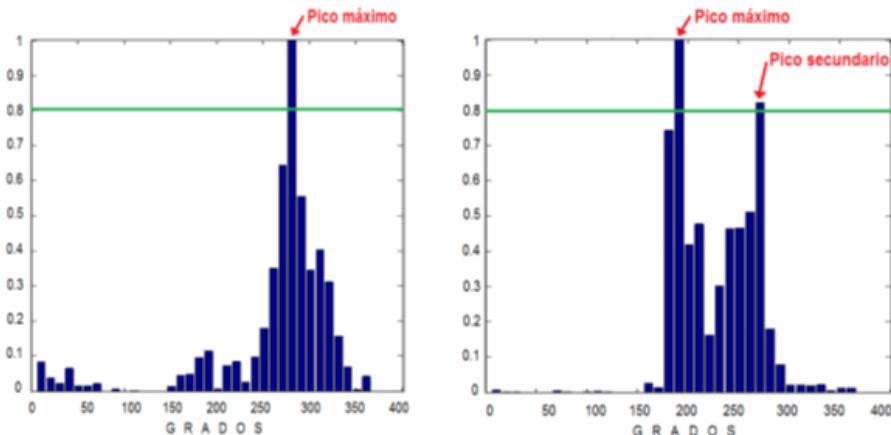
- Dar mayor importancia a los puntos cercanos al *keypoint*, es decir, los puntos centrales de la ventana.

Los picos más altos de cada histograma son las direcciones dominantes de los gradientes locales y por lo tanto la orientación final del *keypoint*. Sin embargo, en algunas ocasiones no basta sólo con definir el pico más alto, sino que se buscan otras características como:

- Detección del pico mayor
- Búsqueda de picos secundarios que tengan una altura mayor al 80% del principal. Si no hay ninguno, se trabaja sólo con el mayor.
- A cada pico seleccionado se le interpola su posición para lograr una mayor precisión, este proceso se lleva a cabo mediante la construcción de una parábola entre él mismo y sus vecinos laterales.

Para localizaciones con múltiples picos elevados de similar magnitud, se obtienen varias orientaciones para un mismo punto de la imagen. Es decir, al construir los descriptores en la siguiente etapa del algoritmo, los *keypoints* con orientaciones múltiples tendrán asignados varios descriptores, que sólo difieren en su inclinación como se puede apreciar en la Figura 16.

Figura 16. Histogramas de keypoints con pico máximo

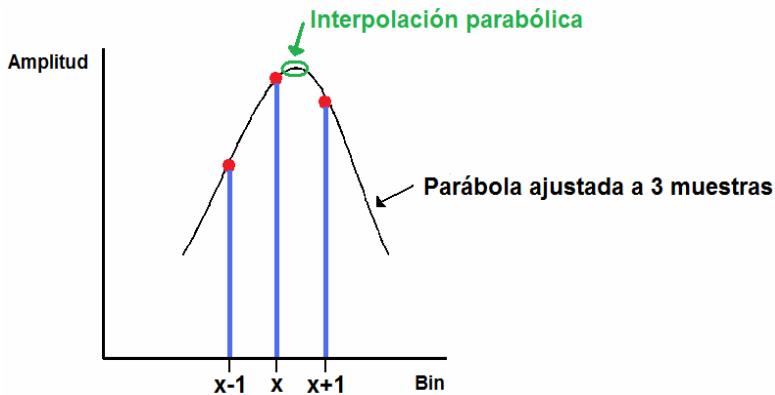


Fuente: [43]

De la Figura 16 se observa a la Izquierda un ejemplo de histograma con orientación simple y a la derecha un ejemplo de histograma con orientaciones múltiples.

En la Figura 17 se puede observar esquemáticamente el proceso de interpolación de los picos a partir del *bin* anterior y posterior.

Figura 17. Interpolación parabólica del máximo utilizando las dos muestras vecinas al pico



Fuente: [43]

Antes de la construcción de los descriptores, de cada región importante de la imagen, se debe tener almacenada toda la información de localización, octava, escala y las principales orientaciones.

- **Keypoints descriptors.** Los parámetros calculados hasta ahora hacen parte de un sistema coordenado de 2 dimensiones que describe localmente cada región de la imagen, y por tanto proporciona invariancia a estos parámetros.

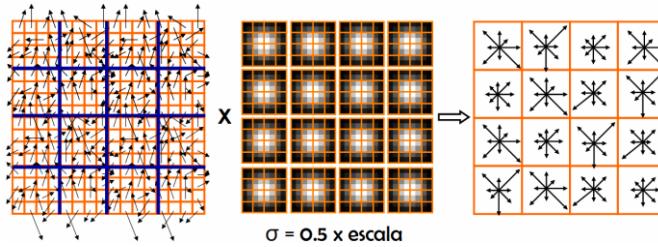
Con esta información, se obtiene un descriptor para cada zona de interés. Éste proporciona robustez ante las posibles variaciones de iluminación y cambios de puntos de vista 3D.

El proceso parte de regiones 16×16 multiplicadas por la ventana gaussiana con $\alpha = 1,5$ veces la escala. Éstas se dividen en subregiones de 4×4 píxeles con el fin de

resumir toda esa información en pequeños histogramas de sólo 8 *bins*, es decir, 8 orientaciones. Antes de realizar esta adaptación de la información, cada gradiente de la ventana 16x16 se rota tantos grados como especifique la orientación principal del *keypoint* (calculada en la etapa anterior), y así será independiente a la inclinación de la imagen.

Para cada *keypoint*, mientras antes se tenía un gran histograma con 36 posibles orientaciones (que provenía de 256 muestras alrededor del punto), ahora se tienen 16 pequeños histogramas de 8 *bins* cada uno de ellos. Para evitar cambios abruptos entre fronteras, cada subregión es filtrada de nuevo por una ventana circular gaussiana (en esta ocasión de tamaño 4x4) con un factor $\alpha = 0,5 \times$ escala como se puede observar en la Figura 18.

Figura 18. Reducción de histogramas



Fuente: [43]

De la Figura 18 se observa a la izquierda las subdivisiones 4x4, en el centro, las ventanas circulares gaussianas y a la derecha el descriptor compuesto por 16 histogramas de 8 *bins*.

Existen dos parámetros que marcan la complejidad del descriptor: el número de orientaciones de cada histograma (r) y el ancho de la matriz (n). En este caso, el tamaño será de 128 elementos como se indica en la ecuación 20.

$$\text{Tamaño} = r \times n^2 = 8 \times 4^2 = 128 \text{ [elementos]} \quad (20)$$

Es necesario realizar una serie de modificaciones para conseguir mayor robustez ante posibles cambios de iluminación. El objetivo es ser invariante a tres tipos de variación:

- Luminosidad
- Contraste
- No linealidades

Este descriptor es invariante a la luminosidad ya que los gradientes se calcularon mediante diferencias entre píxeles vecinos.

Para que este descriptor sea robusto al contraste, se normaliza a la unidad cada uno de los ‘sub-histogramas’ del total de 16 que tiene cada descriptor. Así, un cambio de contraste en el que cada píxel y gradiente sean multiplicados por una constante, será automáticamente cancelado por la normalización.

La variación no lineal de la luz, se puede producir por la saturación de la cámara o cambios de iluminación sobre superficies 3D. Una forma de controlar esta situación, es definir un umbral de valor pequeño a los histogramas normalizados (se eliminan los valores superiores a este valor), y posteriormente se renormaliza de nuevo a la unidad.

Aquí finaliza todo el proceso de construcción del descriptor SIFT. El siguiente paso se centra en cómo comparar los descriptores de las diferentes imágenes para su correcto *matching* entre puntos correspondientes.

4.4.2.1 Cálculo de correspondencias (*matching*). Un descriptor SIFT es un conjunto de 128 elementos que indican las orientaciones más importantes alrededor de un punto de interés.

Si dos imágenes contienen descriptores muy similares, es probable que ambos estén describiendo una misma zona y por tanto sean correspondientes.

Para calcular el grado de similitud entre ellos existen varias alternativas. Una de las más utilizadas es la denominada diferencia euclídea, la cual se calcula por medio de la ecuación 21.

$$dif_i = \sqrt{(a_i - b_i)^2}$$

$$dif_{total} = \sum_1^{128} dif_i \quad (21)$$

donde dif_i es la diferencia euclídea entre el elemento a_i de un descriptor de la imagen A y b_i su correspondiente descriptor de la imagen B. La variable dif_{total} es la suma de las diferencias euclídeas que hay entre los 128 elementos de ambos descriptores.

Realizando estas operaciones entre cada descriptor de la imagen A con cada descriptor de la imagen B, se deducen cuales comparaciones son correspondientes con mayor probabilidad, eligiendo siempre las que hayan dado una diferencia euclídea total menor.

El costo computacional de todas las comparaciones es muy elevado. Por lo tanto, en función del tipo de aplicación que se quiera implementar y de la información que se tenga a priori, se puede optimizar de una u otra forma el código, para esto se deben fijar restricciones previas.

4.4.3 Detector SURF (Speeded Up Robust Features). SURF (Speeded Up Robust Features) [22] es otro algoritmo utilizado para la extracción de puntos de interés invariantes. Este algoritmo tiene gran robustez ante las transformaciones de la imagen. Al comparar el algoritmo **SURF** con el algoritmo **SIFT**, se presentan las siguientes ventajas:

- Velocidad de cálculo considerablemente superior sin ocasionar perdida del rendimiento porque reduce la dimensión y la complejidad del descriptor.
- Mayor robustez ante posibles transformaciones de la imagen.

Estas ventajas se consiguen mediante la reducción de la dimensionalidad y de la complejidad en el cálculo de los vectores de características de los puntos de interés obtenidos, mientras continúan siendo suficientemente característicos e igualmente repetitivos.

El procedimiento del algoritmo SURF para detectar puntos de interés (keypoints), asignación de la orientación y la obtención del descriptor SURF propiamente dicho, se describe a continuación:

4.4.3.1 Detección de Puntos de Interés. La fase de detección de puntos de interés del descriptor SURF es idéntica a la del descriptor SIFT.

El descriptor SURF utiliza el valor del determinante de la matriz Hessiana para la localización y la escala de los puntos. Lo novedoso de la detección de puntos de interés del descriptor SURF es que no utiliza diferentes medidas para el cálculo de la posición y la escala de los puntos de interés individualmente, sino que utiliza el valor del determinante de la matriz Hessiana en ambos casos. Dado un punto $p = (x, y)$ y una imagen I , la matriz Hessiana $H(p, \sigma)$ en el punto p con una escala σ se define a través de la ecuación 22.

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (22)$$

donde $L_{xx}(p, \sigma)$ es la convolución de segundo orden de la Gaussiana $\frac{\delta^2}{\delta x^2} g(\sigma)$ con la imagen I en el punto (x, y) , de igual forma para $L_{xy}(p, \sigma)$ y $L_{yy}(p, \sigma)$.

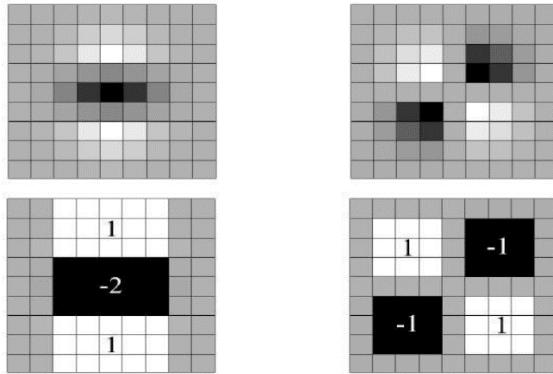
Las aproximaciones de las derivadas parciales se denotan como D_{xx} , D_{xy} y D_{yy} y el determinante se calcula a partir de la ecuación 23.

$$\det(H_{aprox}) = D_{xx}D_{yy} - (0,9D_{xy})^2 \quad (23)$$

donde el valor de 0,9 está relacionado con la aproximación del filtro Gaussiano.

En la Figura 19 se muestra la representación de la derivada parcial de segundo orden de un filtro gaussiano discretizado y la aproximación de la derivada para el caso del descriptor SURF.

Figura 19. Representación de la derivada parcial de segundo orden de un filtro gaussiano discretizado y la aproximación de la derivada en el caso del descriptor SURF



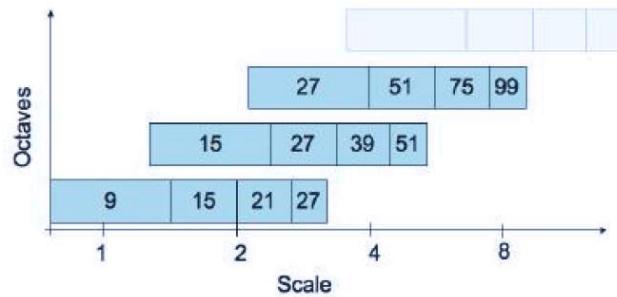
Fuente: [44]

La imagen obtenida después de la convolución de la imagen original con un filtro de dimensiones 9x9, que corresponde a la derivada parcial de segundo orden de una gaussiana con $\sigma = 1,2$, se considera como la escala inicial o también como la máxima resolución espacial s ($s = 1,2$ correspondiente a una gaussiana con $\sigma = 1,2$). Las capas sucesivas se obtienen aplicando gradualmente filtros de mayor dimensión, evitando así los efectos de aliasing en la imagen. El espacio escala para el descriptor SURF, también está dividido en octavas. Sin embargo, en el descriptor SURF, las octavas están compuestas por un número fijo de imágenes como resultado de la convolución de la imagen original con una serie de filtros cada vez más grande. El incremento de los filtros dentro de una misma octava es el doble respecto al de la octava anterior, al mismo tiempo que el primero de los filtros de cada octava es el segundo de la octava predecesora.

Finalmente, para calcular la localización de todos los puntos de interés en todas las escalas, se procede mediante la eliminación de los puntos que no cumplen la condición de máximo en un vecindario de 3x3x3. De esta manera, el máximo determinante de la matriz Hessiana es interpolado en la escala y posición de la imagen.

En la Figura 20 se muestra el escalamiento de un descriptor SURF.

Figura 20. Escala para el descriptor SURF



Fuente: [44]

4.4.3.2 Asignación de la Orientación. En esta etapa se otorga al descriptor de cada punto de interés la invariancia ante la rotación mediante la asignación de la orientación del mismo.

El primer paso para otorgar la mencionada orientación consiste en el cálculo de la respuesta de Haar en ambas direcciones x e y mediante el cálculo de las respuestas HAAR.

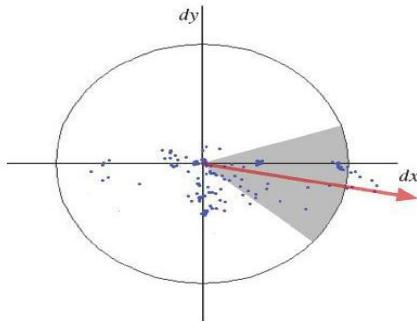
La etapa de muestreo depende de la escala y se toma como valor s , siendo s la escala en la que el punto de interés ha sido detectado, donde a mayor valor de escala mayor es la dimensión de las funciones onduladas.

Una vez realizados todos estos cálculos, se utilizan nuevamente imágenes integrales para proceder al filtrado mediante las máscaras de Haar y obtener así las respuestas en ambas direcciones. Son necesarias sólo 6 operaciones para obtener la respuesta en la dirección x e y . Una vez que las respuestas onduladas han sido calculadas, son ponderadas por una gaussiana de valor $\sigma = 2,5s$ centrada en el punto de interés. Las respuestas son representadas como vectores en el espacio colocando la respuesta horizontal y vertical en el eje de abscisas y ordenadas respectivamente.

Finalmente, se obtiene una orientación dominante por cada sector mediante la suma de todas las respuestas dentro de una ventana de orientación móvil cubriendo un

ángulo de $\frac{\pi}{3}$. En la Figura 21 se muestra la orientación dominante de la suma de todas las respuestas dentro de una ventana de orientación móvil con ángulo $\frac{\pi}{3}$.

Figura 21. Cálculo de direcciones x e y



Fuente: [44]

4.4.3.3 Descriptores SURF. Se construye como primer paso una región cuadrada de tamaño $20s$ alrededor del punto de interés y orientada en relación a la orientación calculada previamente. Esta región es a su vez dividida en 4×4 sub-regiones dentro de cada una de las cuales se calculan las respuestas de Haar de puntos con una separación de muestreo de 5×5 en ambas direcciones. Por simplicidad, se consideran dx y dy las respuestas de Haar en las direcciones horizontal y vertical respectivamente relativas a la orientación del punto de interés.

Para dotar a las respuestas dx y dy de una mayor robustez ante deformaciones geométricas y errores de posición, éstas son ponderadas por una gaussiana de valor $\sigma = 3,3s$ centrada en el punto de interés. En cada una de las sub-regiones se suman las respuestas dx y dy obteniendo así un valor de dx y dy representativo por cada una de las sub-regiones. Al mismo tiempo se realiza la suma de los valores absolutos de las respuestas $|dx_i|$ y $|dy_i|$ en cada una de las sub-regiones, obteniendo de esta manera, información de la polaridad sobre los cambios de intensidad.

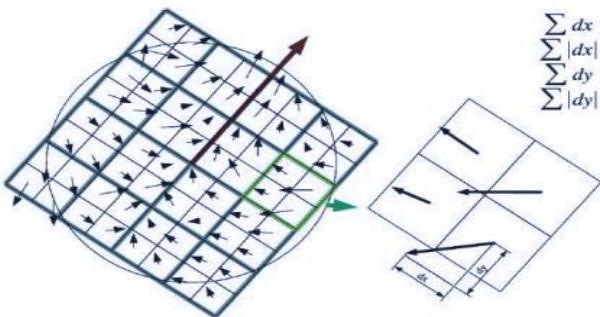
En resumen, cada una de las sub-regiones queda representada por un vector v de componentes definido por la ecuación 24.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (24)$$

Uniendo las 4x4 sub-regiones, resulta un descriptor SURF con una longitud de 64 valores para cada uno de los puntos de interés identificados.

En la Figura 22 se muestra el resultado de un descriptor SURF de longitud 64.

Figura 22. Descriptores SURF



Fuente: [44]

4.5 DESCRIPTORES DE IMÁGENES SEMI-LOCALES

La mayoría de los descriptores de forma pertenecen a esta categoría. Estos descriptores se basan en la extracción de contornos precisos de las formas dentro de la imagen o la región de interés. La segmentación de la imagen suele ser útil en la etapa de pre-procesamiento. Para que un descriptor sea robusto con respecto a transformaciones afines de objetos, es necesario suponer una segmentación casi perfecta de formas de interés.

En la literatura se pueden encontrar muchos descriptores de forma, algunos de los más comunes son: Descriptor de Curvatura de Espacio Escalar (CSS) [23], Transformación Angular Radial (ART) y Descriptores en el estándar MPEG-7.

4.6 MÉTODO DE BOLSA DE PALABRAS VISUALES (BAG OF VISUAL WORDS-BOVW)

Los Descriptores SIFT y SURF han sido ampliamente utilizados para la recuperación de objetos en imágenes. La extracción local de características conlleva a un conjunto desordenado de vectores de características. La principal dificultad del reconocimiento, la recuperación o los pasos de clasificación consiste en encontrar una representación compacta de todas estas características y sus medidas de similitud (disimilitud) asociada. Un método eficiente que ha sido ampliamente utilizado es el llamado Bolsa de Palabras Visuales (BoVW) [24].

El método de Bolsa de Palabras Visuales (BoVW) tiene cuatro etapas principales:

1. La construcción de un diccionario visual por agrupamiento de las características visuales extraídas de un conjunto de imágenes u objetos de entrenamiento.
2. La cuantificación de las características.
3. La selección de la representación de la imagen utilizando el diccionario.
4. La comparación de estas imágenes de acuerdo con su representación.

A continuación se describirán cada uno de estos conceptos:

4.6.1 Diccionarios Visuales. El método más popular y eficaz para representar contenidos visuales hoy en día se basa en los diccionarios visuales que generan la llamada representación de Bolsa de Palabras Visuales (BoVW).

Uno de los beneficios de usar tal representación es la habilidad para codificar propiedades locales en un vector de características único para cada imagen. Para generar una representación de Bolsa de Palabras Visuales, se puede crear el diccionario visual.

Para crear un diccionario visual, las características locales pueden ser obtenidas a partir de un conjunto de imágenes de entrenamiento usualmente extraídas de parches locales y su descripción. Los parches pueden ser tomados alrededor de

puntos de interés [25] o por muestreos densos [26] y descriptores de imágenes como el popular SIFT [27], el cual se utiliza para extraer vectores característicos de cada uno de ellos. Este diccionario, $V = V_i$ con $i = \{1, \dots, K\}$ es construido por la agrupación de estas características en un cierto número de K clases o "Palabras Visuales".

Cada agrupación representa una palabra visual y tiende a contener parches con apariencias similares. Aunque k-means es un método muy popular utilizado en el paso de agrupamiento, debido al caso de la dimensionalidad, una selección aleatoria de puntos en el espacio de características crea diccionarios de calidad similar [28,29] y ahorra mucho tiempo en la generación del diccionario. Para calcular la representación de la imagen, los parches locales de la imagen son asignados a una o varias palabras visuales en el diccionario. La asignación dura proporciona un parche local a la palabra visual más cercana en el espacio de características. De otra mano, la asignación suave reduce el efecto de los pobres resultados obtenidos por el método de agrupación asignando múltiples palabras visuales a los parches locales [17, 18, 19].

Un método computacional efectivo de asignación suave se describe por medio de la ecuación 25.

$$\alpha_{i,j} = \frac{k_\sigma(D(v_i, w_j))}{\sum_{l=1}^k k_\sigma(D(v_i, w_l))} \quad (25)$$

donde:

j Varía desde 1 hasta el tamaño del diccionario (k)

v_i Es el vector característico del parche i

w_j Es el vector correspondiente a la palabra visual j

$$k_\sigma(x) = \frac{1}{2\pi\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$D(v, w)$ Es la distancia entre los vectores v y w

El parámetro σ indica la suavidad de la función gaussiana: El valor y el número de regiones vecinas consideradas.

Después de representar cada parche local de la imagen de acuerdo al diccionario, el conjunto de parches se suma y se define como un vector singular de características por técnicas de agrupamiento [30]. Los métodos de agrupamiento más populares están basados en cálculos de asignación de valor promedio para cada palabra visual en la imagen y consideran la máxima activación de la palabra visual. Los resultados en la literatura muestran un mejor rendimiento en experimentos de clasificación cuando se utiliza la agrupación máxima [23], la cual está dada por ecuación 26.

$$h_j = \max_{i \in N} \alpha_{ij} \quad (26)$$

donde α es obtenido en el paso de asignación, N es el número de puntos en la imagen y j varía de 1 al tamaño del diccionario (k).

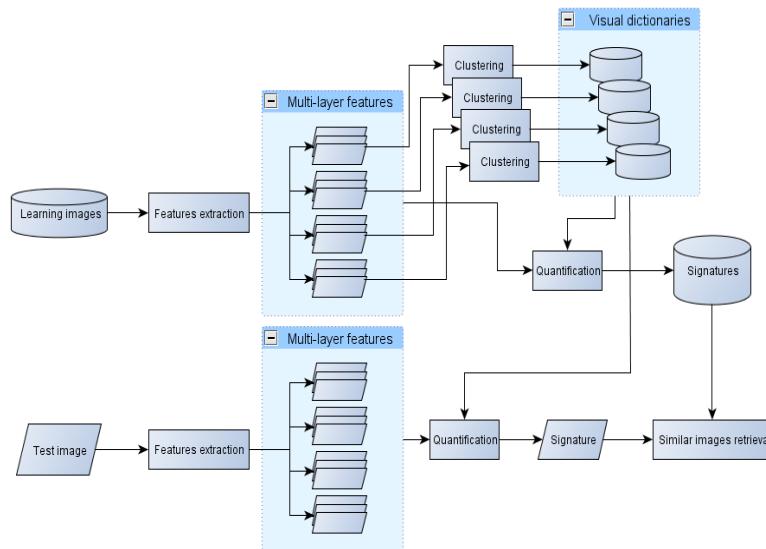
4.6.2 Cuantificación de Características. El conjunto de vectores característicos para el aprendizaje se utiliza para cuantificar el espacio de características (utilizando por ejemplo Agrupamiento K-Means) y para seleccionar el libro de códigos de los vectores característicos que representarán el conjunto de entrenamiento. Con respecto a la cuantificación, es bien sabido que el algoritmo k-means no da garantías para converger en un óptimo global y que depende de la inicialización de los centros de las agrupaciones. Una versión mejorada de este algoritmo, conocido como k-means ++ se ha propuesto en [26] con el fin de incrementar la cantidad de imágenes en un conjunto de datos para construir una cuantificación jerárquica. Por ejemplo, un agrupamiento jerárquica k-means, llamado vocabulario árbol fue propuesto en [31]. El vocabulario árbol tiene una alta calidad de recuperación y de eficiencia comparada con el método de BoVW inicial de [17]. Hasta ahora, sólo se ha mencionado la Bolsa de palabras Visuales para un solo tipo de características, pero hay que tener en cuenta que si diferentes tipos de características son extraídas de las imágenes, el método de BoVW también se puede aplicar directamente. Una vez el diccionario está disponible, las imágenes son representadas por información estadística acerca de cómo se activan las palabras visuales. El conjunto de todos los vectores de características de una imagen se denomina generalmente como "Bolsa de características".

4.6.3 Selección de la Representación. Cuando se crea la representación de una imagen para cualquier aplicación donde se quiera replicar parcial o totalmente una imagen, se requiere la creación de representaciones realmente discriminantes,

diferencias muy pequeñas entre las imágenes u objetos deben ser codificadas, sin dejar de ser válidas para especificar transformaciones fotométricas / geométricas relacionadas con el dominio. Sin embargo la representación debe ser muy precisa.

En la Figura 23 se muestra el esquema general del método de Bolsa de Palabras Visuales.

Figura 23. Esquema General del Método de Bolsa de Palabras Visuales para la recuperación de imágenes



Fuente: [21]

4.6.4 Comparación de Imágenes basada en su Representación. Una aplicación de Búsqueda semántica requiere una representación precisa, pero, al mismo tiempo, lo suficientemente general como para comprender las variaciones entre clases [6]. Aquellas representaciones que son menos específicas para una determinada aplicación pueden ser más adecuadas para abordar problemas de gran cantidad de datos, donde el gran volumen de datos hace que sea más complicado, en términos de tiempo de extracción y de almacenamiento, la extracción de algunas categorías de características de escenarios específicos.

En la recuperación de escenarios, una función de distancia se puede utilizar para comparar vectores de características. La idea de esta función es evaluar si las imágenes contienen las mismas palabras visuales con la misma disposición espacial. Por lo tanto, las distancias entre los puntos se calculan sólo entre las

correspondientes palabras visuales, que presentan una disposición espacial similar. Inicialmente, se buscan las palabras visuales iguales y luego se aplica la verificación espacial para la recuperación de los escenarios.

La razón es que, como la información espacial ya está incluida en el vector de características, la verificación espacial se puede realizar al pasar por el vector de características. Por "post-proceso", se puede entender que, después de encontrar las palabras visuales iguales, se puede calcular la información espacial y luego realizar la verificación espacial como ocurre con los métodos presentados en [32,33]. El vector de características de la imagen final comúnmente se llama bolsa de palabras Visuales (BoVW).

Los métodos de BoW utilizan libros de códigos para crear descriptores basados en el contenido visual de una imagen, donde el libro de códigos es un conjunto de palabras visuales que representan la distribución de las características locales de una colección de imágenes en el espacio característico. Todos los puntos de interés de una imagen están asociados a una o más palabras visuales y un BoW de una imagen está definido de acuerdo a la discretización del espacio de características de acuerdo al libro de códigos y a un histograma de ocurrencias de palabras visuales.

En algunas aplicaciones la semántica asociada con el contenido de una imagen es percibida en términos de la distribución espacial de las propiedades visuales de la imagen. Una de las limitaciones de los métodos de bolsa de palabras es la inhabilidad para codificar la distribución espacial de las palabras visuales dentro de la imagen.

Especial atención se ha prestado a la falta de información geométrica codificada por la tradicional representación de bolsa de palabras visuales [34,35]. El arreglo espacial de palabras visuales en imágenes es importante para entender la semántica de las imágenes y es frecuentemente crucial para distinguir diferentes clases de escenas u objetos, es en esta dirección que muchas propuestas actualmente se han hecho vigentes para la clasificación y recuperación de imágenes [23,25,26,36,37,38]. En la clasificación de escenarios, utilizando Máquinas de soporte vectorial (SVMs), por lo general la alta dimensionalidad de los vectores no degrada la efectividad, ya que las SVMs sufren menos de la maldición de la dimensionalidad.

El popular método de Pirámide espacial es muy exitoso para la clasificación de imágenes, pero sus vectores tienen una alta dimensionalidad [37]. Sin embargo, para los experimentos de recuperación, que se basan generalmente en el cálculo de distancias entre vectores utilizando la distancia euclídea.

Por ejemplo, los vectores deben ser compactos o incrustados en una estructura indexada para evitar la maldición de la dimensionalidad [21,39,40]. Como la dimensionalidad crece, la distribución de las distancias entre las características tiende a ser estrechamente concentrada, alrededor de un valor medio, lo que reduce el contraste entre las características similares y disímiles. Sin embargo para crear una representación de imágenes que trabaje bien tanto en clasificación como en recuperación de escenarios, se debe ser consciente del tamaño del vector característico.

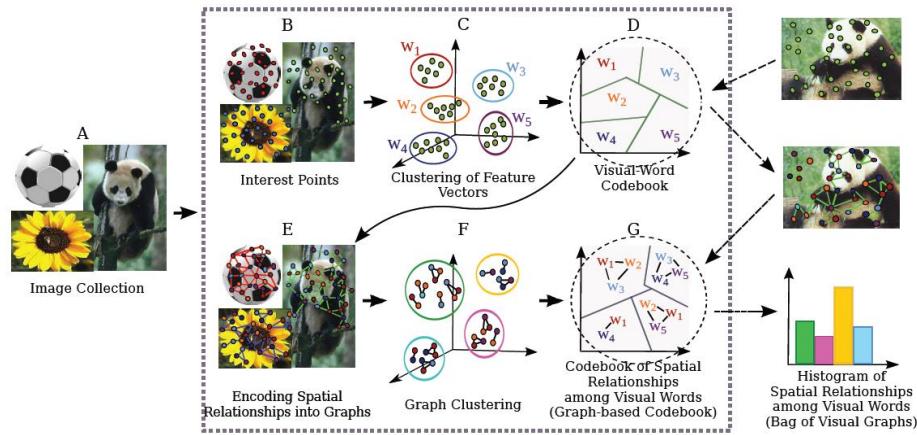
Muchos de los métodos existentes por agrupación espacial que han sido utilizados en recuperación de escenarios, dejan la verificación espacial como un paso de posprocesamiento [29,30]. Ellos calculan una simple representación y después buscan las palabras visuales que coincidan entre imágenes.

4.7 MÉTODO DE GRAFOS VISUALES

El método de grafos ha sido utilizado para representar la relación entre los objetos dentro de la imagen. La Figura 24 muestra la visión general de la Bolsa de Gráficos Visuales.

Como se puede apreciar en la Figura 24, de una imagen de Colección (A), se detectan todos los puntos de interés (B), entonces se agrupan los descriptores de los puntos de interés en el espacio de características (C) y se genera el libro de palabras visuales (D) a partir de los prototipos de las agrupaciones.

Figura 24. Visión general de la Bolsa de Gráficos Visuales



Fuente: [20]

Utilizando el libro de códigos y la triangulación de Delaunay en los puntos de interés de cada imagen se puede construir un conjunto de grafos conectados (E) para representar la imagen, con la cual se codifica la relación espacial de las palabras visuales. En una nueva agrupación en el paso (F), se seleccionan las palabras visuales del nuevo vocabulario (G), los Grafos Visuales. El proceso para generar el descriptor de la Bolsa de Grafos Visuales de una imagen utiliza el libro de códigos basado en Grafos (G) para calcular un histograma, el cual cuenta la frecuencia de los Grafos Visuales dentro de una imagen.

4.7.1 Bolsa de Grafos Visuales (BoVG). La Bolsa de Grafos Visuales (BoVG) es un método que combina la posición espacial de los puntos de interés y sus etiquetas definidas en términos del libro de códigos de palabras visuales tradicional para definir un conjunto de grafos conectados.

El conjunto de grafos conectados codifica la relación espacial de las palabras visuales y se utiliza para crear un segundo vocabulario, el Libro de Códigos de Grafos Visuales. El Libro de códigos de grafos cuantifica el espacio de grafos conectados y permite la construcción de un histograma para describir la imagen. La Figura 24 muestra los pasos para generar el libro de códigos de grafos visuales propuesto y el descriptor final de la imagen. Este método necesita 2 libros de códigos uno en los puntos de interés del espacio de características y otro en el espacio de grafos conectados.

4.7.1.1 Libro de Códigos de Palabras Visuales. Este método necesita un diccionario visual o libro de códigos, para asignar cada punto de interés encontrado dentro de una imagen para definir la palabra visual en el libro de códigos. Primero se inicia con un conjunto de puntos de interés obtenido de un muestreo denso como resultado de un detector de puntos de interés [17,19].

El segundo paso es la cuantificación del bajo nivel del espacio del descriptor y la construcción de su propio libro de códigos. Un método común para esto es el K-Means clustering [29,30], aunque una selección simple aleatoria de puntos produce resultados también efectivos [39].

Una vez se tiene el libro de códigos, éste se puede utilizar para crear un descriptor global para describir una nueva imagen, lo cual conlleva a realizar 2 pasos:

La asignación de puntos y el paso de agrupamiento. Aquí serán 2 métodos fundamentales para asignación: Asignación Dura y Asignación Blanda [14,30]. En la asignación dura, los puntos son asignados a su propia región en el espacio de características cuantificado. En la asignación blanda se etiqueta un punto para cualquier región basado en diferentes criterios.

El paso de agrupamiento genera un descriptor de BoW de la imagen [14]. La suma del agrupamiento representa una imagen por un histograma, calculando la suma de asignación de cada región. El uso de una suma de asignaciones normalizada corresponde a un agrupamiento promedio. El máximo agrupamiento considera el máximo valor de actividad de cada región.

El agrupamiento, es la formación de la representación final de la imagen. Una buena representación debe ser robusta para diferenciar las transformaciones de la imagen y el ruido; y debe ser lo más compacta posible. El operador de agrupamiento agrega las proyecciones de todos los vectores de características de entrada en el diccionario visual para obtener un solo valor escalar de cada palabra de código.

El estándar BoVW [47] considera el tradicional método de recuperación de texto como un operador de suma como se aprecia en la ecuación 27.

$$\forall j = 1 \dots \dots k, z_i = \sum_{i=1}^N \alpha_{ij} \quad (27)$$

donde el máximo agrupamiento, se define como:

$$\forall j = 1 \dots \dots k, z_i = \max_{i=1,\dots,N} \alpha_{i,j}$$

El asocio con la codificación dispersa permite obtener un alto rendimiento que resume el agrupamiento [33]. Tanto el máximo agrupamiento como la suma de agrupamiento han sido estudiados en [29]. Otras variantes a estos dos operadores tradicionales de agrupamiento se han propuesto recientemente, algunas se centran en la aplicación de la etapa de agrupamiento en áreas más locales. El más poderoso es, probablemente, el método de comparación de Pirámide espacial (SPM) [30]. Una pirámide espacial de imágenes predeterminadas es primeramente calculada. El BoVW se construye en particiones anidadas del plano de la imagen que van desde particiones gruesas a particiones de finas resoluciones. En otras palabras, el agrupamiento se realiza sobre las celdas de una pirámide espacial en lugar de hacerlo sobre toda la imagen. En [1] un método llamado "Frases Visuales" se introduce para agrupar palabras visuales de acuerdo a su proximidad en el plano de la imagen como una secuencia de características.

Las frases visuales son representadas por un histograma que contiene la distribución de las palabras visuales en la frase. La información espacial también se ha tenido en cuenta en [24]. De hecho, se propone una inmersión espacial de características utilizando diagramas locales de Delaunay.

La ventaja de la triangulación de Delaunay es que es invariante a Transformaciones afines del plano de la imagen preservando los ángulos. Otra mejora de BoVW que pertenece a la clase de codificación agregada es el método de Kernel de Fisher propuesto en [39], éste se basa en el uso del núcleo de Fisher con una mezcla de Modelos Gaussianos (GMM) estimados sobre todo el conjunto de imágenes.

4.7.1.2 Libro de Códigos Basado en Grafos. Se propone el uso de grafos para codificar la relación espacial entre palabras visuales. Suponiendo que $G = (V, E)$ es un grafo ponderado asociado con la imagen I, donde V es un conjunto de vértices y E es un conjunto de aristas que unen dos vértices en V. En este método el conjunto de vértices V tiene un máximo de tres vértices con los cuales se representan los puntos de interés marcados, es decir, puntos asociados con las palabras visuales, las aristas ponderadas están asociados con las distancias entre vértices interconectados, mayor distancia, mayor peso.

Se utiliza el método propuesto por Hashimoto [42] para definir las aristas entre vértices, las cuales utilizan la triangulación de Delaunay para definir las aristas entre los puntos de interés. El paso siguiente es la selección de las aristas basado en sus respectivos pesos. Las aristas con bajo peso se eliminan debido a que codifican relaciones entre puntos cercanos, los cuales no aportan mucho en la definición de los arreglos espaciales de las palabras visuales. Las aristas con alto peso también se eliminan ya que están asociadas con estructuras no locales. Sin embargo, como todas las imágenes deben tener al menos un grafo, estas limitaciones no son tenidas en cuenta cuando una imagen no tiene suficiente cantidad de puntos de interés.

En resumen cualquier imagen puede ser representada por un conjunto de grafos conectados que codifican la distribución espacial de estas palabras visuales.

4.7.2 Creación de un Libro de Códigos Basado en Grafos. El proceso para generar un libro basado en grafos es similar al proceso descrito en la sección 4.7.1.2. Se define un nuevo vocabulario teniendo en cuenta la similaridad de los grafos extraídos de una colección de imágenes de entrada.

Una palabra en el libro de palabras llamado Grafo Visual se refiere a un grupo de arreglos espaciales similares de palabras visuales. Aquí de nuevo se puede utilizar un método de agrupamiento o una simple selección aleatoria para la definición de grupos de grafos.

Tanto para agrupación como para asignación, se necesita una función de emparejamiento gráfico, para medir la distancia de cada grafo de una imagen al libro de palabras basado en grafos. Se utiliza una función basada en Jouili [40]. En Este método la distancia entre 2 grafos es definida en términos de las firmas de los vértices.

Se supone que $G = (V, I)$ es un grafo de una imagen I . Cada vértice $\in V$ y tiene la forma de la ecuación 28.

$$S(v_i) = \{l_i, grado(v_i), e_{i1}, e_{i2}\} \quad (28)$$

donde l_i es la etiqueta de la palabra visual asociada con el vértice v_i , y el grado (v_i) es el grado del vértice y e_{ij} es una firma basada en cierta textura asociada con la arista que enlaza a v_i .

Como en [16], dados los grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, la función de distancia que calcula la disimilaridad está dada por la ecuación 29.

$$D(G_1, G_2) = \frac{\hat{C}}{|C|} + ||G_1| - |G_2|| \quad (29)$$

Donde $|G_1|$ es el número de vértices en el grafo G_i , \hat{C} es el costo del máximo ajuste del grafo y $|C|$ es una constante de normalización que se refiere al número de vértices comparados.

El costo máximo de comparación de un par de grafos es calculado aplicando el método Húngaro en las dos matrices de distancia C_1 y C_2 . Cada elemento de ambas matrices corresponden a la distancia entre un vértice de un grafo G_1 y un vértice de un grafo G_2 . Las matrices C_1 y C_2 difieren en como la distancia entre firmas de aristas es calculada. En C_1 la función de disimilaridad de vértices considera que las aristas deberán ser asignadas con respecto a todas las firmas de los vértices en dirección antihoraria. En C_2 la función de disimilaridad es calculada utilizando la dirección opuesta para cada arista asignada en cada grafo: para el grafo G_1 se asumen sus aristas en dirección antihoraria, mientras que las aristas del grafo G_2 se asumen en dirección horaria.

Para las matrices C_1 y C_2 el costo de ajuste óptimo se define por $\hat{C} = \min(\hat{C}_1, \hat{C}_2)$, donde \hat{C}_i corresponde al resultado de aplicar el método Húngaro a la matriz C_i . El uso del Método Húngaro en ambas matrices tiene como objetivo en el manejo de las transformaciones de reflexión.

La distancia entre 2 vértices es calculada en términos de la superposición de distancias entre los vértices etiquetados y la distancia de Manhattan normalizada para las firmas de las aristas.

Considerando los grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, la distancia entre los vértices $v_i \in V_1$ y $v_j \in V_2$ está dada por la ecuación 30.

$$d(v_i, v_j) = f(l_i, l_j) + \sum_{k=1}^N \frac{|e_{i1k} - e_{j1k}|}{N} + \sum_{k=1}^N \frac{|e_{i2k} - e_{j2k}|}{N} \quad (30)$$

donde:

$$f(l_i, l_j) = \begin{cases} 0 & \text{si } l_i = l_j \\ 1 & \text{en otro caso} \end{cases}$$

y v_i y v_j tienen un número diferente de aristas y la distancia entre las firmas de aristas asume un valor máximo igual a 1.

4.7.3 Creación de una Bolsa de Grafos Visuales. Después de describir una imagen con un conjunto de grafos conectados, se utiliza el libro de códigos basado en grafos para asignar cada grafo conectado de la imagen a un grafo visual. Una imagen puede estar representada por una Bolsa de Palabras Visuales, es decir un vector que describa la distribución de Grafos visuales dentro de la imagen. Para esta tarea se calcula un histograma normalizado utilizando una asignación dura y una agrupación promedio.

Dado que los costos computacionales para la creación del libro de códigos de palabras visuales y la clasificación de imágenes son los mismos que los observados en el método de BoW, este método sólo tiene un costo adicional relacionado con la creación del libro de códigos basado en grafos.

4.8. MÁQUINAS DE VECTORES DE SOPORTE

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

Estos métodos están propiamente relacionados con problemas de clasificación y regresión.

Dado un conjunto de ejemplos de entrenamiento (de muestras) se pueden etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio

possible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo en función de su proximidad pueden ser clasificadas a una u otra clase.

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

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

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

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

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

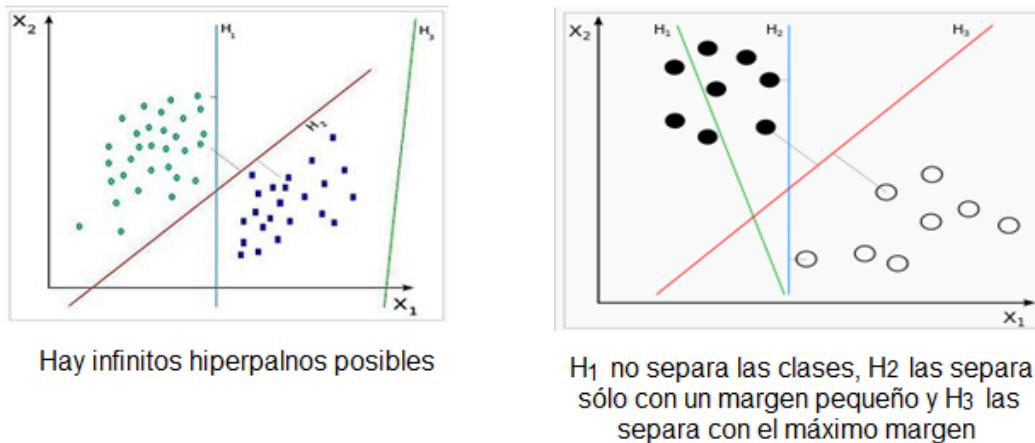
Los algoritmos SVM pertenecen a la familia de los clasificadores lineales. También pueden ser considerados un caso especial de la regularización de Tikhonov.

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

Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte. Los modelos basados en SVMs están estrechamente relacionados con las redes neuronales. Usando una función kernel resultan un método de entrenamiento alternativo para clasificadores polinomiales, funciones de base radial y perceptrón multicapa.

En la Figura 25 se tiene un ejemplo idealizado para 2-dimensiones.

Figura 25. Ejemplos de los hiperplanos de la máquina de vector de soporte



Fuente: [32]

En la Figura 25 se puede apreciar que la representación de los datos a clasificar se realiza en el plano x-y. El algoritmo SVM trata de encontrar un hiperplano 1-dimensional (en el ejemplo se trata de una línea) que une a las variables predictoras y constituye el límite que define si un elemento de entrada pertenece a una categoría o a la otra.

Existe un número infinito de posibles hiperplanos (líneas) que realicen la clasificación pero, ¿cuál es la mejor y cómo se define?

La mejor solución es aquella que permita un margen máximo entre los elementos de las dos categorías.

Se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible.

4.8.1.1 Errores de Entrenamiento. Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (overfitting).

Con el fin de permitir cierta flexibilidad, los SVM manejan un parámetro C que controla la compensación entre los errores de entrenamiento y los márgenes rígidos, creando así un margen blando (soft margin) que permita algunos errores en la clasificación a la vez que los penaliza.

4.8.1.2 Función Kernel. La manera más simple de realizar la separación es mediante una línea recta, un plano recto o un hiperplano N-dimensional. Desafortunadamente los universos a estudiar no se suelen presentar en casos idílicos de dos dimensiones como en el ejemplo mostrado por medio de la Figura 5, sino que un algoritmo SVM debe tratar con a) más de dos variables predictoras, b) curvas no lineales de separación, c) casos donde los conjuntos de datos no pueden ser completamente separados, d) clasificaciones en más de dos categorías.

Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de las máquinas de aprendizaje lineal, es decir, se mapeará el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad (Hilbert) como se muestra en la ecuación 32.

$$F = \{\varphi(x) / x \in X\} \quad (32)$$

donde:

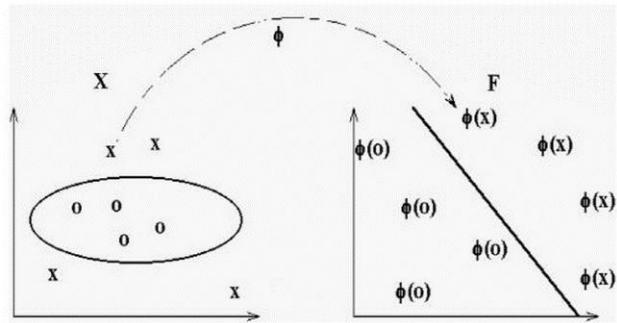
$$x = \{x_1, x_2, \dots, x_n\} \rightarrow \varphi(x) = \{\varphi(x_1), \varphi(x_2), \dots, \varphi(x_n)\}$$

Tipos de Funciones Kernel (Núcleo):

- Polinomial - Homogénea:

$$K(x_i, x_j) = (x_i \cdot x_j)^n$$

Figura 26. Función polinomial homogénea

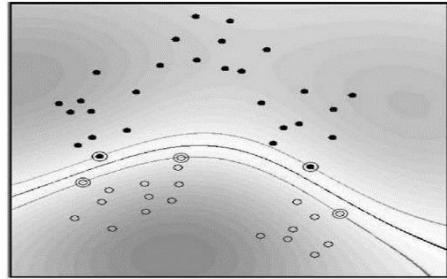


Fuente: [32]

- Perceptron:

$$K(x_i, x_j) = \|x_i \cdot x_j\|$$

Figura 27. Función Perceptron

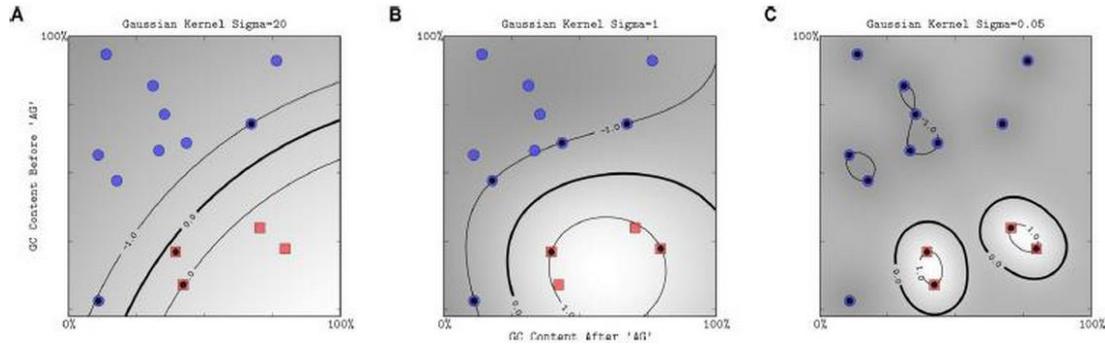


Fuente [32]

- Función de base radial Gaussiana: Separado por un hiperplano en el espacio transformado.

$$K(x_i, x_j) = e^{\frac{-(x_i - x_j)^2}{2\sigma^2}}$$

Figura 28. Función de base radial Gaussiana



Fuente: [32]

- Sigmoid:

$$K(x_i, x_j) = \tanh(x_i \cdot x_j - \theta)$$

4.8.1.3 SVR Regresión. Una nueva versión de SVM para regresión fue propuesta en 1996 por Vladimir Vapnik, Harris Drucker, Chris Burges, Linda Kaufman y Alex Smola. La idea básica de SVR consiste en realizar un mapeo de los datos de entrenamiento $x \in X$ a un espacio de mayor dimensión F a través de un mapeo no lineal $\varphi: X \rightarrow F$ donde se puede realizar una regresión lineal.

Multiclasificación: Hay dos filosofías básicas para resolver el problema de querer clasificar los datos en más de dos categorías:

- a) Cada categoría es dividida en otras y todas son combinadas.
- b) Se construyen $\frac{k(k-1)}{2}$ modelos donde k es el número de categorías.

4.8.1.4 Comparación entre las Redes Neuronales Artificiales y las Máquinas de Vector de Soporte. El cuadro 2 muestra un comparativo entre las Redes Neuronales Artificiales y las Máquinas de Vector de Soporte.

Cuadro 2. Comparativo entre redes neuronales artificiales y máquinas de vector de soporte

Comparación ANN versus SVM

ANNs

- Capas ocultas transforman a espacios de cualquier dimensión
- El espacio de búsqueda tiene múltiples mínimos locales
- El entrenamiento es costoso
- La clasificación es muy eficiente
- Se diseña el número de capas ocultas y nodos
- Muy buen funcionamiento en problemas típicos

SVMs

- Kernels transforman a espacios de dimensión muy superior
- El espacio de búsqueda tiene sólo un mínimo global
- El entrenamiento es muy eficiente
- La clasificación es muy eficiente
- Se diseña la función kernel y el parámetro de coste C
- Muy buen funcionamiento en problemas típicos
- Extremadamente robusto para generalización, menos necesidad de heurísticos para entrenamiento

Fuente: [32]

5. METODOLOGÍA

Por medio de este trabajo se diseñó un sistema que permite el reconocimiento de imágenes utilizando el método de Bolsa de Palabras Visuales (BoVW) y Máquinas de Vector de Soporte (SVM). Para ello, se utilizaron las herramientas de Matlab: Código de programación, toolbox de computer visión y paquete de procesamiento de imágenes VL_FEAT.

En el desarrollo de este proyecto de grado se implementaron varios esquemas de tratamiento de imágenes que conllevan a la implementación del sistema de reconocimiento de imágenes utilizando la técnica de Bolsa de Palabras Visuales (B of V W).

A continuación se hace la descripción de los diferentes esquemas.

5.1 ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES

Inicialmente se implementó un esquema que muestra la potencialidad de la extracción de características de las imágenes y su fiabilidad como un soporte fundamental para las tareas de clasificación de objetos en las imágenes y el posterior reconocimiento de éstas. Para ello se implementaron en MATLAB programas y GUI's que muestran la extracción de características de las imágenes utilizando varios de los descriptores definidos en el documento, como el detector de esquinas de HARRIS, los descriptores SIFT y SURF y su aplicación en el reconocimiento de objetos de las imágenes a partir de la comparación de similitud entre las características de estas.

El método de búsqueda de coincidencias entre los puntos clave, se basa en la distancia euclídea, la cual se define como la distancia más corta medida entre dos puntos del espacio. En el caso general, esta distancia está dada por la línea recta que une los dos puntos y se calcula por medio de la ecuación 33.

$$D(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (33)$$

donde P_1 y P_2 son los puntos a tratar y $(x_1, x_2), (y_1, y_2)$ son las posiciones de dichos puntos en el espacio.

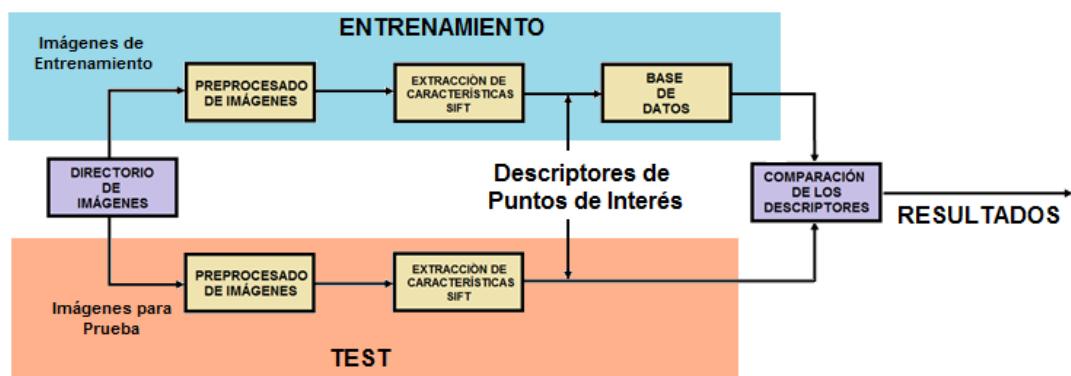
Para encontrar las coincidencias en las imágenes en comparación se obtienen dos vectores *matches* y *scores*.

El vector *matches* es un listado de dos columnas. En la primera, se indica el número correspondiente a los puntos de interés (keypoints) de la imagen de entrada o imagen de prueba y en la segunda el número correspondiente a los puntos de interés (keypoints) de la imagen almacenada en el entrenamiento o imagen de entrenamiento, con la que se está realizando la búsqueda de coincidencias en ese momento y en el que se haya encontrado un parecido razonable (una distancia euclídea que indique una posible coincidencia entre keypoints). El vector *scores* es un listado que contiene el valor de la distancia euclídea de las coincidencias obtenidas entre los descriptores de las dos imágenes tratadas en cada momento (la de entrada y la almacenada en el entrenamiento). Se obtiene un vector de cada tipo para cada una de las imágenes del entrenamiento con las que se compara la imagen de entrada o imagen de prueba.

Tras la búsqueda de coincidencias y dentro del bloque de comparación de los descriptores, se realiza un filtrado de los resultados obtenidos, pues no todos serán válidos. Imponiendo un umbral, se recorre el vector de distancias euclídeas y se almacenan las que tengan un valor inferior a dicho umbral.

Este valor, será uno de los que servirá para la realización de las pruebas del sistema y que llevará a unas conclusiones en la búsqueda de los mejores valores y métodos para el reconocimiento de objetos en imágenes mediante la Transformada SIFT. El diagrama de bloques del esquema implementado se muestra en la Figura 29.

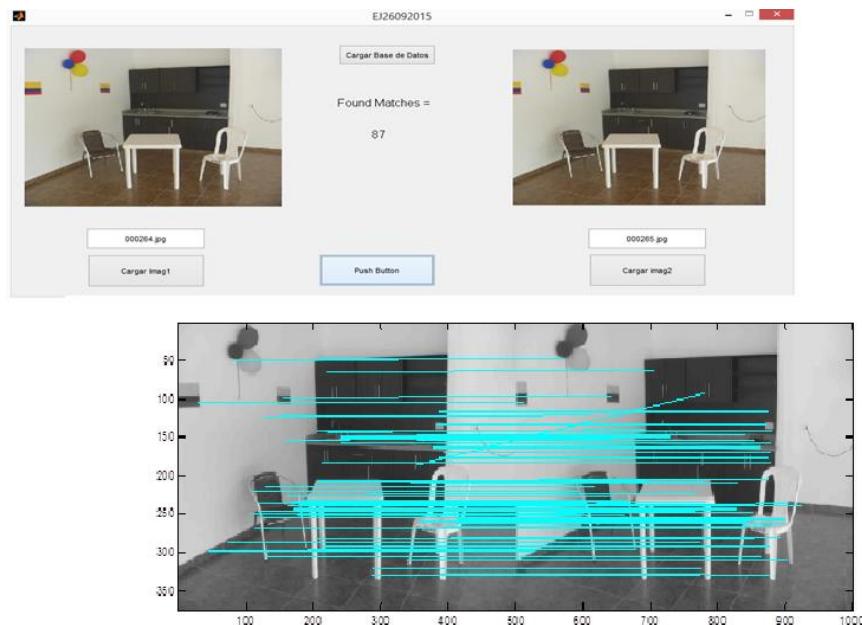
Figura 29. Diagrama de bloques del esquema implementado



Fuente: Modificado por el Autor de [17]

El resultado de la búsqueda de coincidencias puede apreciarse por medio del ejemplo de la Figura 30.

Figura 30. Resultado de la búsqueda de coincidencias en uno de los experimentos realizados.



Fuente: Autor

5.2 ESQUEMA DE DETECCIÓN DE OBJETOS

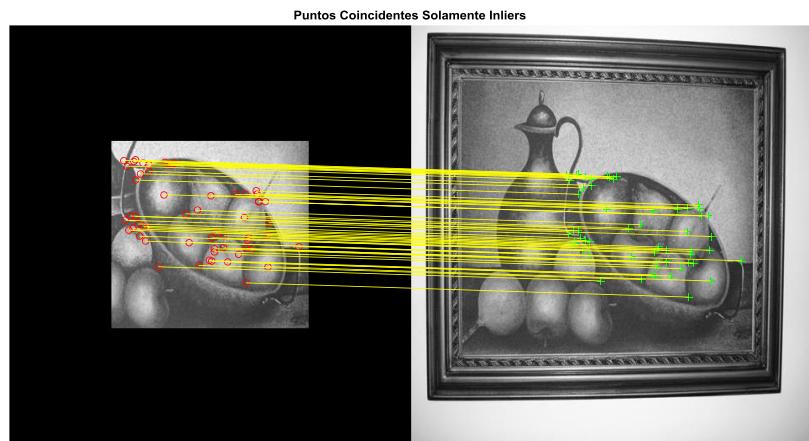
Detección de objetos en una imagen, basado en la búsqueda de correspondencias entre puntos característicos de las imágenes de destino y de referencia.

En segunda instancia, se presenta un esquema para la detección de un objeto específico que hace parte de una imagen, basado en la búsqueda de correspondencias entre puntos de las imágenes de destino y de referencia. Este esquema es capaz de detectar objetos a pesar de un cambio de escala o de rotación. También es robusto a pequeñas variaciones de la rotación y a occlusiones fuera de plano.

Este esquema de detección de objetos funciona mejor para objetos que exhiben patrones de textura no repetidos, lo cual permiten patrones únicos de comparación de características. Este esquema probablemente no funciona bien para objetos de color uniforme o para los objetos que contienen patrones que se repiten. Note que este algoritmo es óptimo para detectar un objeto específico.

La Figura 31 muestra un ejemplo con el resultado de la aplicación del esquema de detección de objetos específicos en imágenes, basado en la búsqueda de correspondencias entre puntos característicos de las imágenes.

Figura 31. Detección de un jarrón de frutas en la imagen de un bodegón



Fuente: Autor

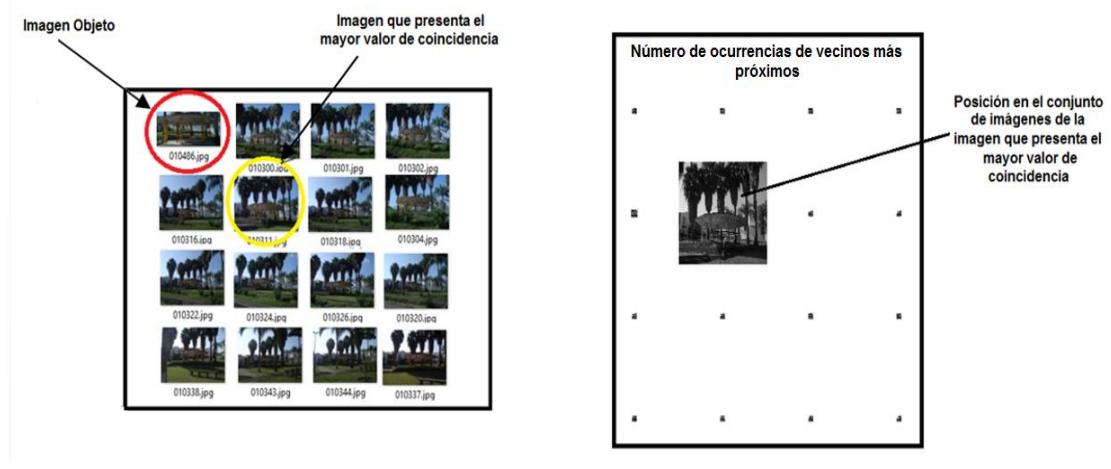
5.3 ESQUEMA DE BÚSQUEDA

Esquema de búsqueda de objetos específicos, en un conjunto de imágenes a partir de un alto grado de coincidencia.

Este esquema muestra cómo buscar un objeto específico en un conjunto de imágenes, dada una imagen representativa del objeto. Permite encontrar de manera eficiente el vecino más cercano por comparación de los puntos SURF de las imágenes de destino y de referencia. Dado que utiliza características de

descriptores SURF, este método de correspondencia es robusto a rotaciones y a cambios en la escala. La imagen ejemplo se muestra en la Figura 32.

Figura 32. Colección de Imágenes



Fuente: Autor

En la Figura 32 se puede observar que el objeto específico de la imagen de referencia es un kiosco y el resto de imágenes del conjunto de imágenes destino, tienen incluido el kiosco visto desde diferentes puntos de vista, el sistema selecciona del conjunto de imágenes destino, la imagen que presenta el mayor alto grado de coincidencia con la imagen de referencia.

5.4 SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES UTILIZANDO LA TÉCNICA DE BOLSA DE PALABRAS VISUALES (BOVW)

Uno de los métodos utilizados en visión por computador es el denominado sistema CBIR (Content-Based Image Retrieval System). Los sistemas CBIR se utilizan para recuperar imágenes de una colección de imágenes que son similares a la imagen de consulta. La aplicación de este tipo de sistemas se puede encontrar en muchas áreas, tales como búsqueda de productos en la web, sistemas de vigilancia, identificación visual de lugares específicos.

El sistema de recuperación de imágenes utiliza diferentes funciones como BoVW- Bolsa de Palabras Visuales, diferentes Descriptores de imagen, para representar el conjunto de datos de las imágenes. Las imágenes son indexadas para crear un mapeo de palabras visuales. El índice define el número de apariciones de cada palabra visual en la colección de imágenes. La comparación entre la imagen de consulta y el índice de apariciones, proporciona aquellas imágenes más similares a la imagen de consulta.

Otro de los sistemas utilizados fue el sistema de clasificación y reconocimiento de imágenes utilizando la técnica de bolsa de palabras visuales (BoVW).

Para la implementación de este sistema, en primer lugar fue necesario diseñar las fases más importantes del sistema. La primera fase, denominada fase de entrenamiento se encarga de extraer las características de las imágenes de entrenamiento. La segunda fase denominada fase de validación y reconocimiento es la encargada de introducir las imágenes de consulta para que el sistema realice la tarea de reconocimiento.

A continuación se describe el sistema de clasificación y reconocimiento de imágenes utilizando la técnica de Bolsa de Palabras Visuales (BofVW) a través de tareas de reconocimiento de objetos en una imagen dada usando sus vectores de características.

La estructura del vector de característica consta del cálculo de características SIFT sobre una rejilla regular a través de la imagen ('densa SIFT') y la cuantificación del vector en palabras visuales. El clasificador es una máquina lineal de vector de soporte (SVM). En clasificación de imágenes, una imagen se clasifica de acuerdo a su contenido visual. Una aplicación importante es la recuperación de imágenes, la cual consiste en buscar a través de una base de datos de imágenes aquellas imágenes con un contenido visual particular que coincide con la imagen de consulta.

Un sistema de clasificación de imágenes consta de fases de entrenamiento y prueba de las imágenes a clasificar. Dicho procedimiento consta de las siguientes etapas:

Etapa 1: Preparación de Datos

La base de datos se compone de diferentes clases de imágenes. A las imágenes de la base de datos se hace necesario realizar el pre-procesamiento del vector de características para cada una de las imágenes. El vector de características consiste

en el cálculo de características SIFT sobre una rejilla regular a través de la imagen ('densa SIFT') y la cuantificación del vector en palabras visuales.

Etapa 2: Entrenamiento del sistema clasificador de imágenes

El sistema clasificador es una máquina lineal de vector de soporte (SVM). Primero se evalúa cualitativamente, el rendimiento del clasificador usándolo para clasificar todas las imágenes de entrenamiento.

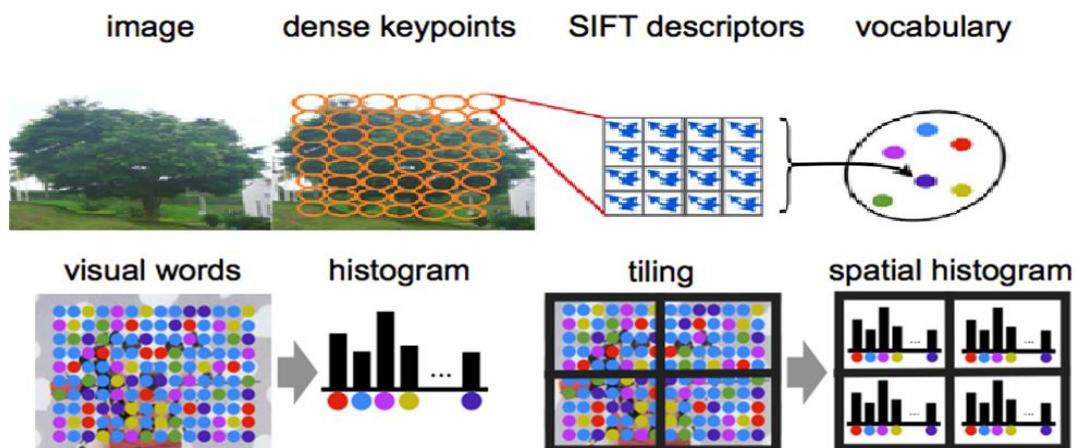
Etapa 3: Clasificación de las imágenes de prueba y evaluación del desempeño

En esta etapa, una vez entrenado el sistema clasificador, se inicia la clasificación de las imágenes de prueba y la evaluación de su rendimiento, utilizando la medida de desempeño del clasificador para clasificar todas las imágenes de prueba.

Etapa 4: Entrenamiento del clasificador para las otras clases de imágenes y evaluación de su desempeño

En esta etapa, se entrena el clasificador para otra clase de imágenes que se deseen clasificar por parte del usuario, se repiten las etapas (2) y (3) para cada una de las otras clases. En la Figura 33 se resumen todas las etapas del proceso de clasificación.

Figura 33. Etapas del proceso de clasificación



Fuente: Autor

Todo el procedimiento del sistema de clasificación y reconocimiento de imágenes ejecuta las siguientes funciones:

- Búsqueda de un directorio de imágenes: Se realiza por medio de la función: `getImageSet.m`.
- Re-escalamiento de las imágenes de la base de datos a un formato estándar: Se realiza a través de la función: `standardizeImage.m`.
- Cálculo de puntos de interés y descriptores (SIFT Denso): Se realiza mediante la función: `computeFeatures.m`.
- Cuantificación de descriptores visuales en palabras visuales: Se realiza por medio de la función: `quantizeDescriptors.m`.
- Cálculo de histogramas espaciales de palabras visuales: Se realiza sobre las características mediante la función: `computeHistogram.m`.
- Reducción de los histogramas espaciales a histogramas de una dimensión: Se realiza a través de la función: `removeSpatialInformation`.
- Cálculo de histogramas de las palabras visuales: Este cálculo se realiza sobre las imágenes utilizando la función: `computeHistogramFromImage.m`.
- Aplicación del histograma de las palabras visuales sobre la lista de imágenes de la base de datos de las imágenes a clasificar: Se realiza por medio de la función: `computeHistogramsFromImageList.m`.
- Entrenamiento de la máquina lineal de vector de soporte: Se realiza mediante la función: `trainLinearSVM.m`.
- Visualización del rendimiento y puntuación del clasificador para clasificar todas las imágenes de prueba del subconjunto de imágenes: Se la realiza a través de la función: `displayRankedImageList.m`.

Estas funciones son del toolbox VLFeat elaborado por Andrea Velardi y su grupo de investigación, las cuales fueron adaptadas y modificadas por el autor del proyecto para su sistema específico.

La Figura 34 muestra el esquema general del sistema de clasificación y reconocimiento de Imágenes implementado.

Figura 34. Esquema general del sistema de clasificación y reconocimiento de Imágenes.



Fuente: Autor

6. RESULTADOS OBTENIDOS DEL SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES

El sistema diseñado en el presente proyecto, es un sistema que clasifica y reconoce imágenes utilizando las técnicas de “Bolsa de Palabras Visuales” y “Máquinas de Vectores de Soporte”.

Inicialmente se analizó la efectividad de los descriptores de características para hallar los puntos de interés de algunas de las imágenes de la base de datos de imágenes.

Para la realización de las pruebas de clasificación y reconocimiento del sistema, se utilizó una colección de diferentes clases de imágenes, dicha colección está conformada por aproximadamente 2000 imágenes tomadas por el autor sobre espacios interiores como salas, cocinas y cuadros de diferentes tipos (bodegones, vírgenes, flores, etc.); espacios exteriores como kioscos, árboles, paisajes, flores, etc. Además de las imágenes propias del autor, se utilizaron imágenes de bases de imágenes públicas disponibles en la web.

El procedimiento empleado para verificar el correcto funcionamiento del sistema, inicia con la comprobación de los distintos esquemas implementados, para lo cual, se utilizaron imágenes cuyas características fueron modificadas. Entre estas características se pueden mencionar: Rotación, escala, iluminación, oclusiones e introducción de objetos en algunas imágenes.

Dado que los descriptores locales son una representación de vector muy compacta de un vecino local, al utilizarlos se pueden mejorar los algoritmos que detectan características como cambios de escala, rotación y oclusiones.

En este proyecto, se hace un tratamiento extensivo de diferentes métodos, tales como: La detección de esquinas utilizando FAST, Harris y Shi & Tomasi, la detección de zonas de características como SURF y MSER, la descripción de puntos de interés como SURF, FREAK, BRISK y HOG.

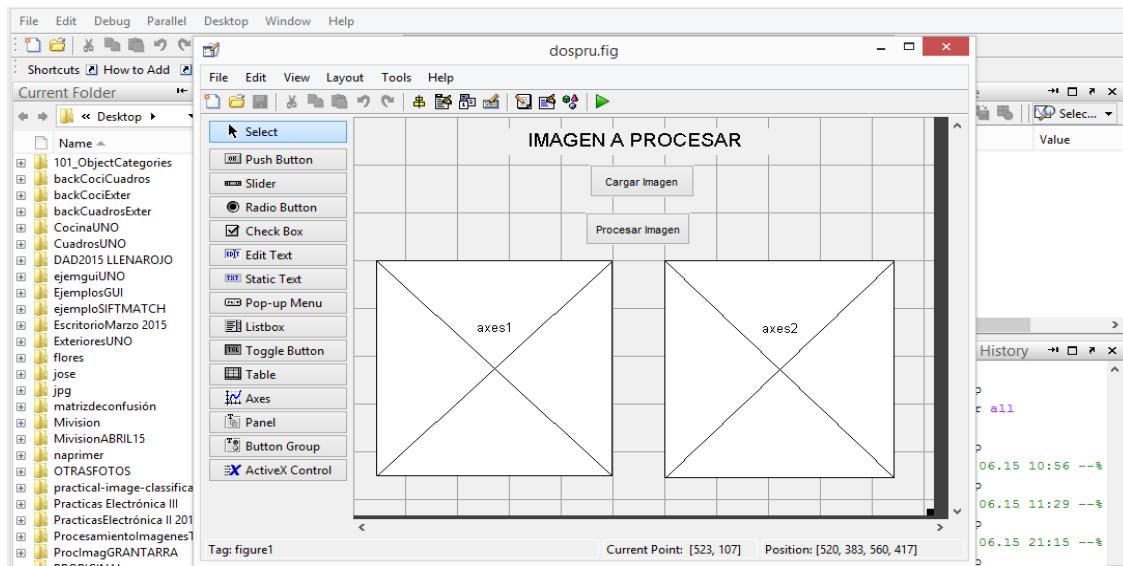
6.1 ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES

6.1.1 Ejemplo de Localización de Puntos de Interés. A continuación se presentan algunos ejemplos de localización de puntos de interés.

Fase 1.

Inicialmente se implementó una GUI en Matlab para la realización de diferentes funciones como se puede apreciar en la Figura 35.

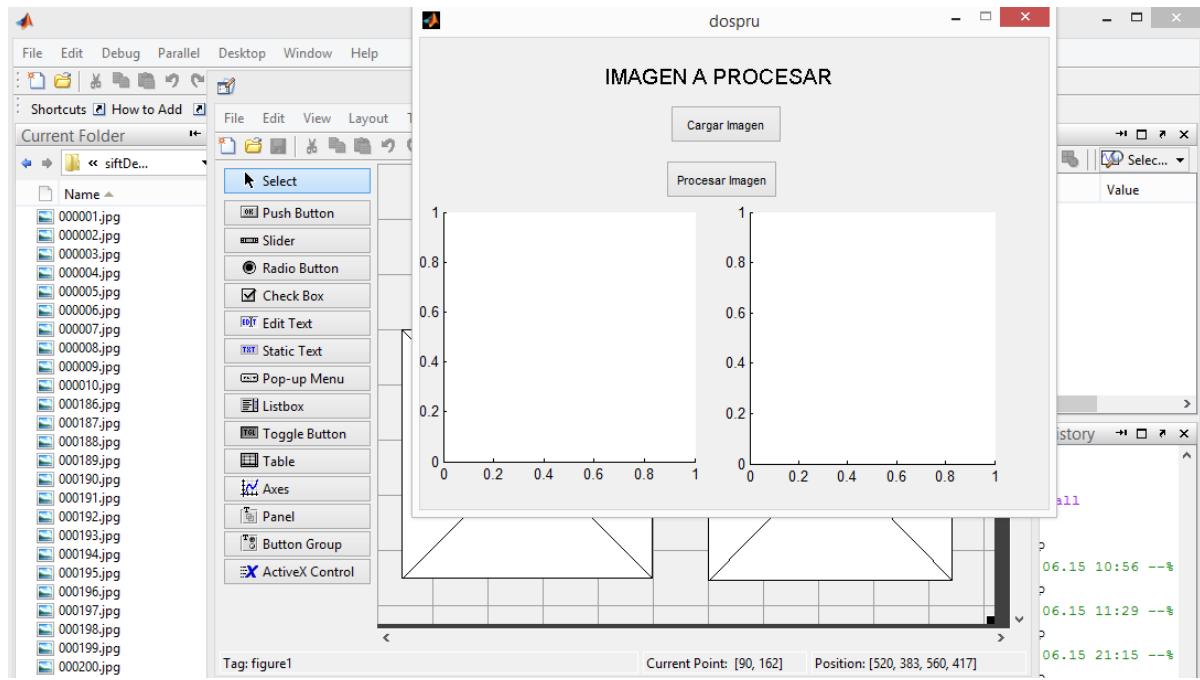
Figura 35. GUI implementada para la realización de las diferentes funciones con imágenes



Fuente: Autor

La GUI implementada permite seleccionar cualquier tipo de imagen almacenada en una carpeta que contiene **una base de datos de imágenes** y representar tanto en la imagen sus puntos de interés, así como sus descriptores.

Figura 36. GUI para ejecutar las funciones de extracción de puntos de interés

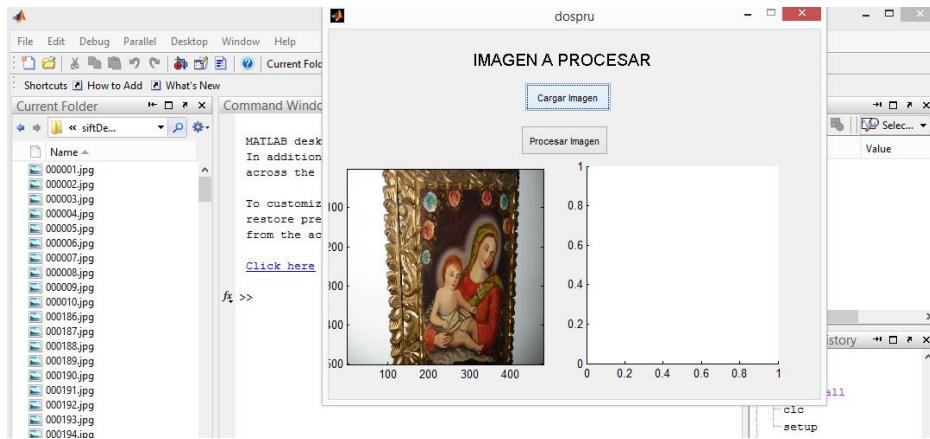


Fuente: Autor

La GUI de la Figura 36 permite seleccionar cualquier tipo de imagen almacenada en la carpeta que contiene **una base de datos de imágenes** y representar los puntos de interés en la imagen seleccionada.

En la Figura 37, se muestra un ejemplo de la GUI de la Figura 36 con una imagen seleccionada para la cual se localizarán los puntos de interés.

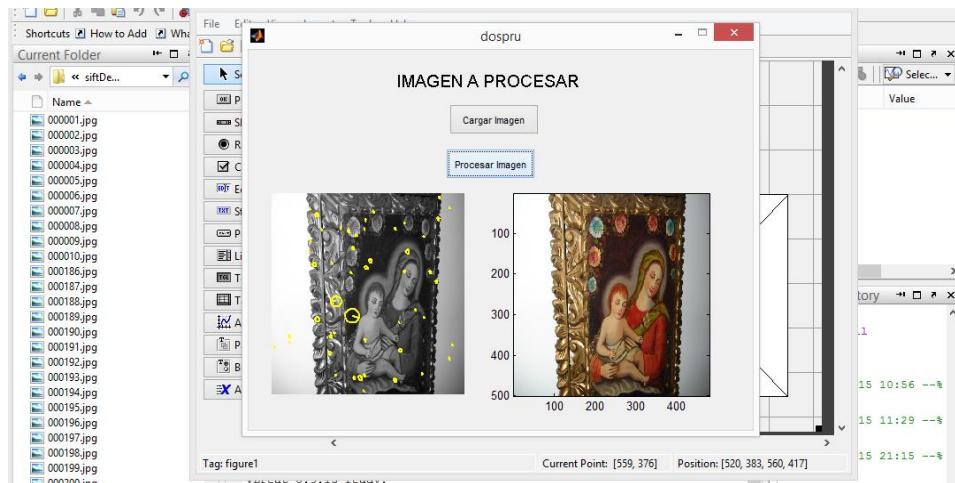
Figura 37. GUI que permite la selección de la imagen de la base de datos de imágenes a procesar.



Fuente: Autor

La Figura 38 muestra la imagen seleccionada y la representación de los puntos de interés encontrados.

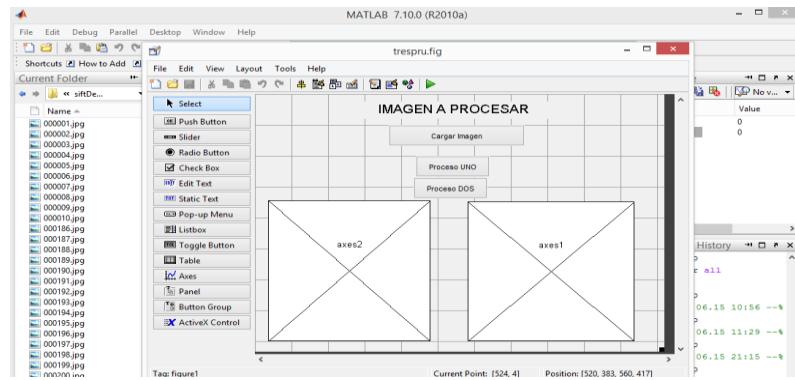
Figura 38. GUI con imagen original y con la representación de los puntos de interés



Fuente: Autor

La Figura 39 es una GUI que ejecuta el proceso UNO, el cual haya los puntos de interés para cualquier imagen de la base de datos de imágenes y el proceso DOS, el cual le haya los descriptores de los puntos de interés a cualquier imagen de la base de datos de imágenes.

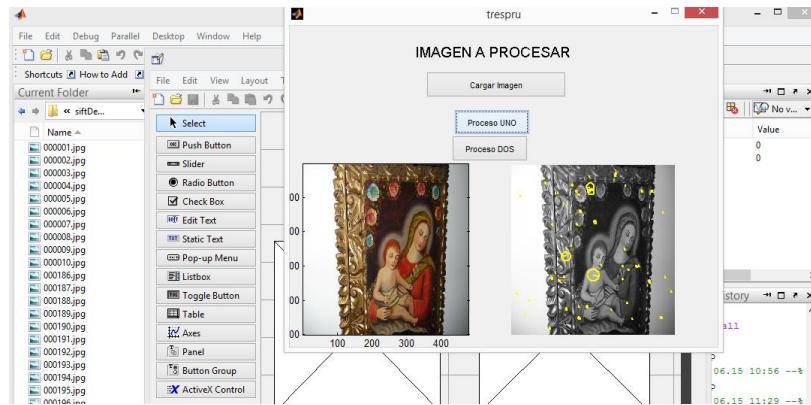
Figura 39. GUI que ejecuta el proceso UNO y el proceso DOS



Fuente: Autor

Al ejecutar el proceso UNO se le hallan los puntos de interés a la imagen seleccionada como se puede apreciar en la Figura 40.

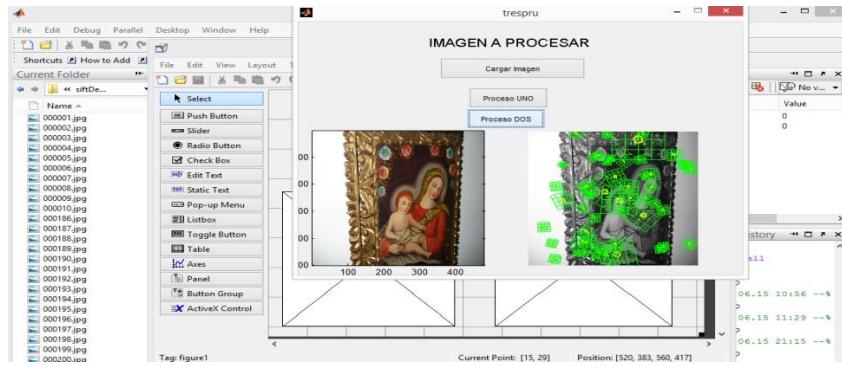
Figura 40. GUI que ejecuta el proceso UNO



Fuente: Autor

Al ejecutar el proceso DOS se le hallan los descriptores de los puntos de interés a la imagen seleccionada como se muestra en la Figura 41.

Figura 41. GUI que ejecuta el proceso DOS

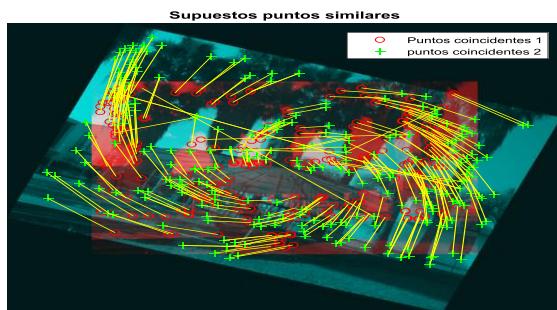


Fuente: Autor

6.1.2 Ejemplo de Correspondencia de Puntos Característicos. Este ejemplo muestra la correspondencia de puntos característicos para una imagen original cuando se compara ella misma con su imagen rotada.

La Figura 42, muestra como en una imagen, que se le han localizado sus puntos de interés y se ha rotado, al extraer nuevamente sus puntos de interés, estos se preservan en la imagen rotada. Con este ejemplo se concluye que los puntos de interés, son invariantes a la rotación. Esta es una de las características más importantes para los procesos de reconocimiento y clasificación de imágenes.

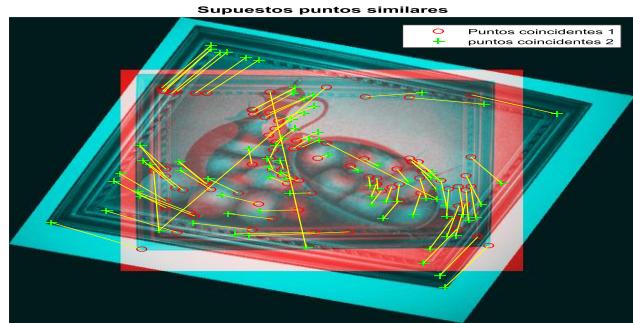
Figura 42. Figura original de un kiosco y figura del kiosco rotado y sobrepuerto



Fuente: Autor

Otro ejemplo de una imagen a la cual se le hayan sus puntos de interés y se hace la comparación de estos en la imagen original y en la imagen rotada se muestra en la Figura 43.

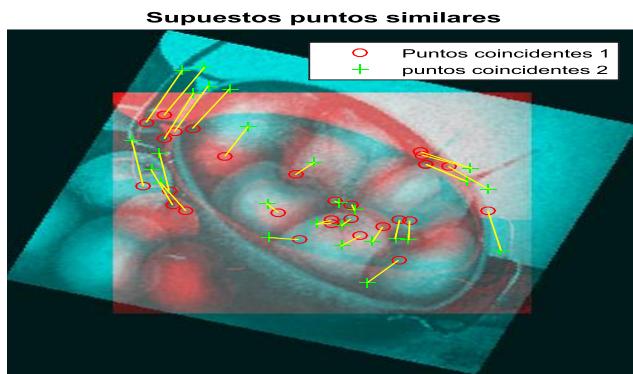
Figura 43. Figura original de un bodegón y figura del bodegón rotado y sobre puesto



Fuente: Autor

En la Figura 44 se hace otro ejemplo de una imagen a la cual se le localizan sus puntos de interés y se hace la comparación de estos en la imagen original y en la imagen rotada.

Figura 44. Figura original de un frutero y figura del frutero rotado y sobre puesto



Fuente: Autor

6.1.3 Ejemplo de Detección de Esquinas. Para este ejemplo se implementó una GUI que permite seleccionar cualquier tipo de imagen de la carpeta que contiene la **base de datos de imágenes**, aplicar el detector de esquinas de HARRIS a cualquier imagen de esta base y variar el número de bordes detectado modificando un factor que va desde 0,1 hasta 0,9.

La Figura 45 muestra la imagen seleccionada de la base de datos para realizar este ejemplo.

Figura 45. Imagen seleccionada de la base de datos de imágenes



Fuente: Autor

A la imagen mostrada en la Figura 45 se le aplica el detector de esquinas de HARRIS y se obtiene la imagen de la Figura 46.

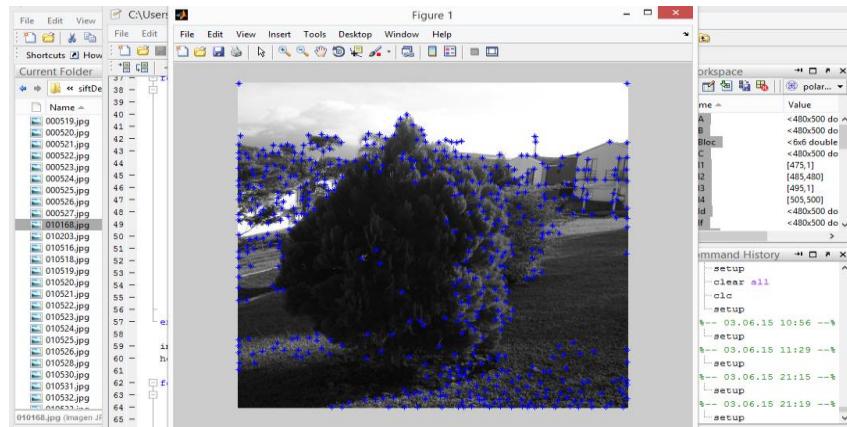
Figura 46. Imagen con detección de bordes



Fuente: Autor

En la Figura 47 se muestra otra forma de aplicar el detector de esquinas de HARRIS con el número de bordes modificado.

Figura 47. Detector de esquinas de HARRIS



Fuente: Autor

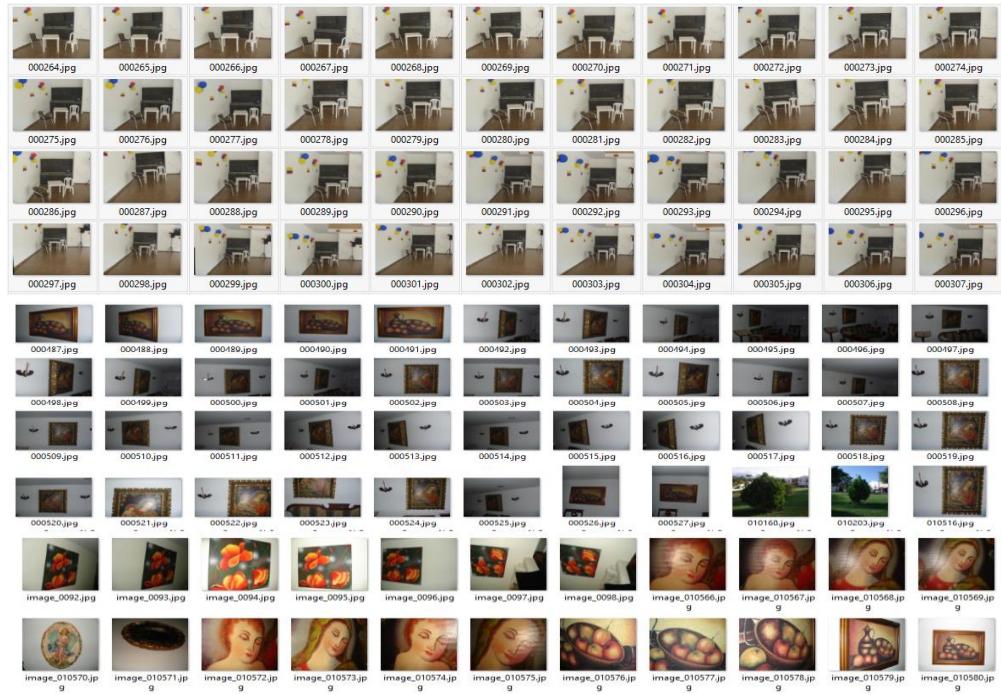
6.2 ESQUEMA DE DETECCIÓN DE PUNTOS DE INTERÉS Y BÚSQUEDA DE COINCIDENCIAS ENTRE IMÁGENES

6.2.1 Ejemplos de comparación de imágenes a partir de la coincidencia de sus puntos de interés. Para el esquema de reconocimiento por comparación de puntos de interés utilizando el descriptor SIFT, se hicieron pruebas con las siguientes imágenes, para lo cual se utilizó otra de las GUI implementadas.

Para la comparación de los puntos de interés entre las imágenes, se hizo el cálculo del producto punto entre vectores unitarios. Es importante aclarar que la relación de ángulos (ACOS de productos escalares de vectores unitarios) es una aproximación cercana a la relación de las distancias euclídeas para ángulos pequeños, esta relación fue denominada como el parámetro `distRatio`, el cual es una relación que define el valor de los ángulos de los vectores entre puntos vecinos de una imagen con otra a comparar. Este parámetro sólo mantiene aquellas comparaciones en las cuales la relación de los ángulos de vectores entre vecinos de la imagen uno, con la imagen dos está a menos del valor `distRatio` definido para dicha comparación. Para el caso de los ejemplos dicha relación está en el rango de 0,1 hasta 0,9.

En la Figura 48 se muestra la colección de imágenes de la base de datos a las cuales se les va a realizar dichas comparaciones.

Figura 48. Colección de imágenes para realizar las comparaciones



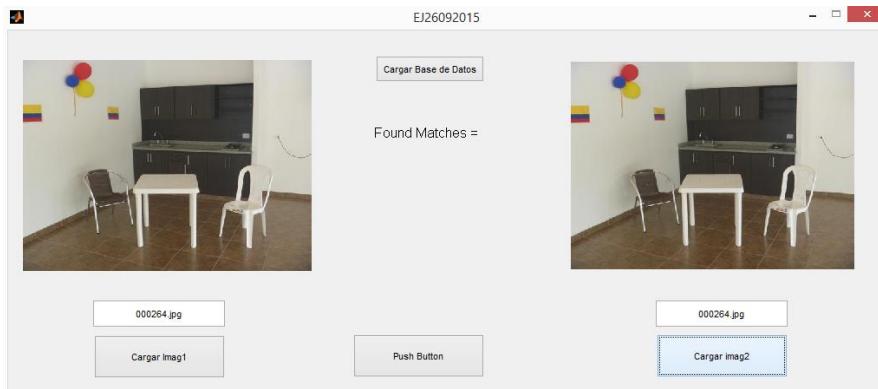
Fuente: Autor

En las GUI implementadas a partir de la Figura 49, realizan la comparación de imágenes utilizando la coincidencia de sus puntos de interés. En estas GUI se pueden escoger las imágenes de cualquier directorio que esté disponible con imágenes a través de la opción cargar imágenes.

En la ventana izquierda de las GUI se carga la imagen de prueba y en la ventana derecha la imagen de la base de datos con la cual se hará la comparación. Una vez las imágenes están cargadas, el programa realiza la extracción de los puntos de interés de las imágenes que se están comparando en ese momento y realiza la búsqueda de coincidencias de los puntos de interés de dichas imágenes, definiendo el número de puntos coincidentes.

6.2.1.1 Caso 1: Comparación de la misma imagen. La primera comparación se hace comparando la imagen con ella misma, para esto, se tomó una imagen de la base de datos, referenciada como: 000264.jpg como se puede apreciar en la Figura 49.

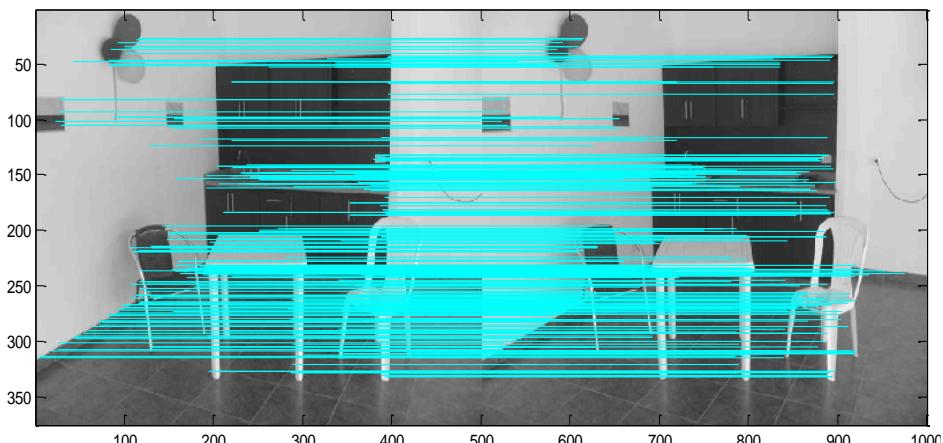
Figura 49. Comparación entre la misma imagen



Fuente: Autor

Al aplicar diferentes valores de distRatio = 0,9, 0,6 y 0,3 a la imagen, el esquema encuentra 309 puntos de interés como se muestra en la Figura 50.

Figura 50. Coincidencia para un distRatio = 0,9, 0,6, 0,3



Fuente: Autor

6.2.1.2 Caso 2: Comparación de dos imágenes con cambio de perspectiva. La segunda comparación se hace entre las imágenes 000264.jpg y 000265.jpg de la base de datos. Como se observa en la Figura 51, las imágenes tomadas corresponden a una misma escena vista desde dos puntos diferentes de perspectiva.

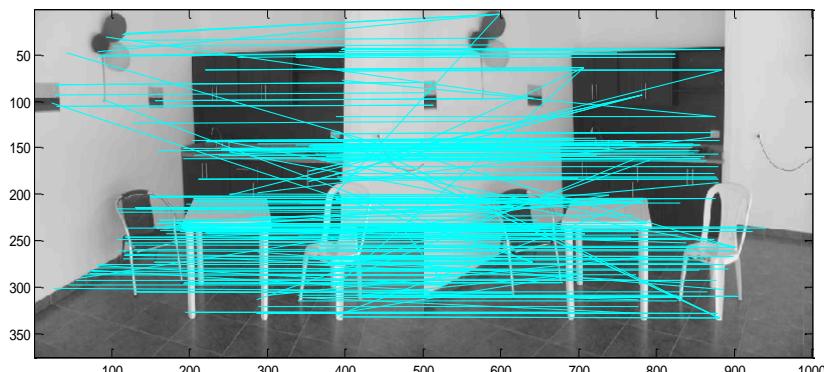
Figura 51. Comparación entre dos imágenes con diferente perspectiva



Fuente: Autor

Para la comparación mostrada en la Figura 51, se empleó un $\text{distRatio} = 0,9$ y el esquema encontró para la imagen 000264.jpg 309 puntos de interés, para la imagen 000265.jpg 302 puntos de interés y entre estas dos imágenes halló 230 puntos coincidentes los cuales se muestran en la Figura 52.

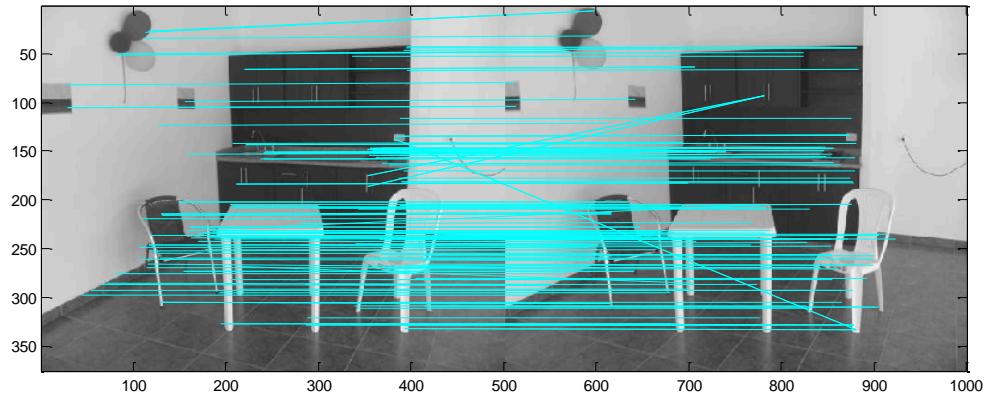
Figura 52. Coincidencia para un $\text{distRatio} = 0,9$



Fuente: Autor

Al hacer nuevamente la comparación de la Figura 51 con un distRatio = 0,6 el esquema halló que entre estas dos imágenes hay 158 puntos que son coincidentes como se puede apreciar en la Figura 53.

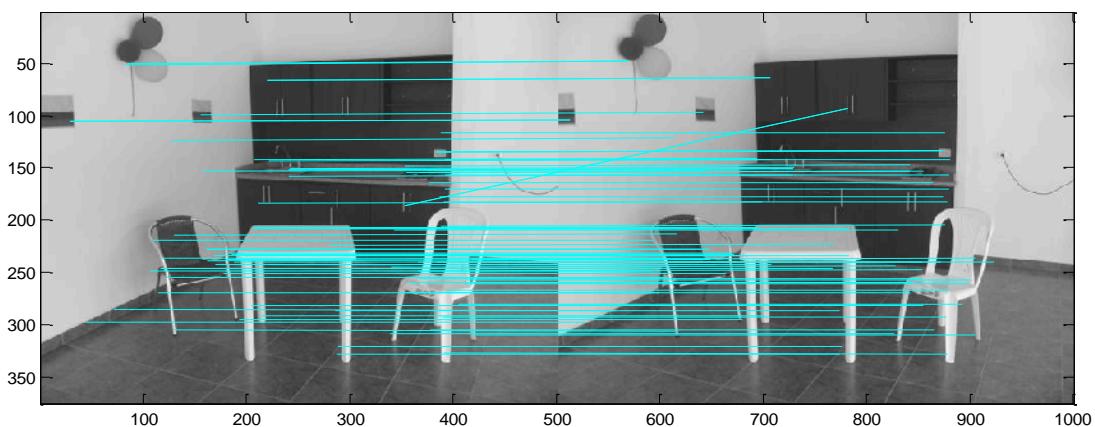
Figura 53. Coincidencia para un distRatio = 0,6



Fuente: Autor

Utilizando un distRatio = 0,3 en la comparación de la Figura 51, se encuentran 87 puntos que son coincidentes como se muestra en la Figura 54.

Figura 54. Coincidencia para un distRatio = 0,3

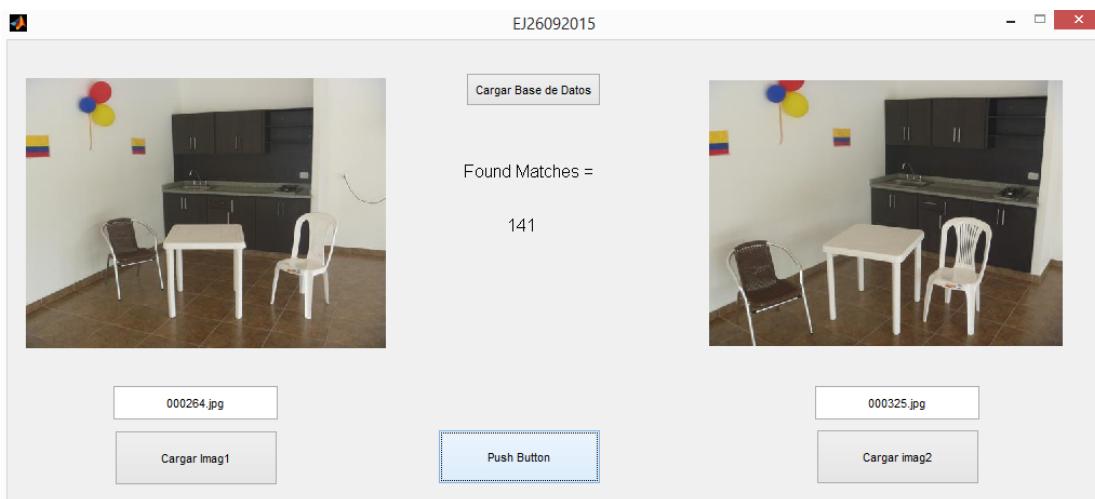


Fuente: Autor

Al analizar las comparaciones anteriores, se puede apreciar que para un menor valor del parámetro distRatio se genera un número de puntos coincidentes más exacto, lo cual permite obtener una mejor comparación y resultados más confiables.

6.2.1.3 Caso 3: Comparación de dos imágenes con cambio de escala. La tercera comparación se hace entre las imágenes 000264.jpg y 000325.jpg de la base de datos. Como se observa en la Figura 55, las imágenes tomadas corresponden a una misma escena con alta variación en la escala.

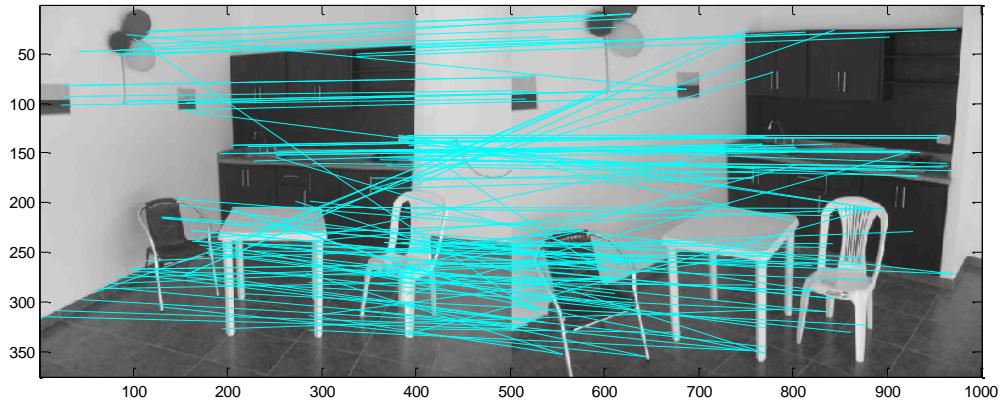
Figura 55. Comparación entre dos imágenes con cambio de escala



Fuente: Autor

Para la comparación mostrada en la Figura 55, se empleó un $\text{distRatio} = 0,9$ y el esquema encontró para la imagen 000264.jpg 309 puntos de interés, para la imagen 000325.jpg 352 puntos de interés y entre estas dos imágenes halló 141 puntos coincidentes los cuales se muestran en la Figura 56.

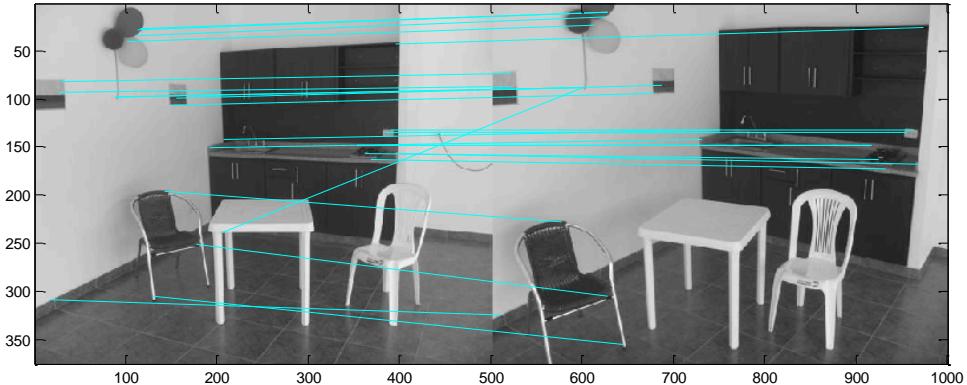
Figura 56. Coincidencia para un distRatio = 0,9



Fuente: Autor

Al hacer nuevamente la comparación de la Figura 55 con un distRatio = 0,6 el esquema halló que entre estas dos imágenes hay 25 puntos que son coincidentes como se puede apreciar en la Figura 57.

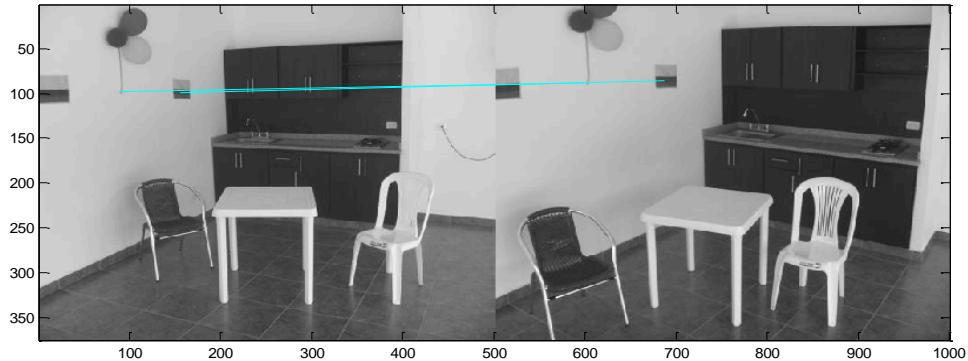
Figura 57. Coincidencia para un distRatio = 0,6



Fuente: Autor

Utilizando un $\text{distRatio} = 0,3$ en la comparación de la Figura 55, se encuentran 2 puntos que son coincidentes como se muestra en la Figura 58.

Figura 58. Coincidencia para un $\text{distRatio} = 0,3$



Fuente: Autor

6.2.1.4 Caso 4: Comparación de dos imágenes con cambio de escala y cambio de perspectiva. La cuarta comparación se hace entre las imágenes 000264.jpg y 000345.jpg de la base de datos. Como se observa en la Figura 59, las imágenes tomadas corresponden a una misma escena con alta variación en la escala y la perspectiva.

Figura 59. Comparación de dos imágenes con cambio de escala y cambio de perspectiva

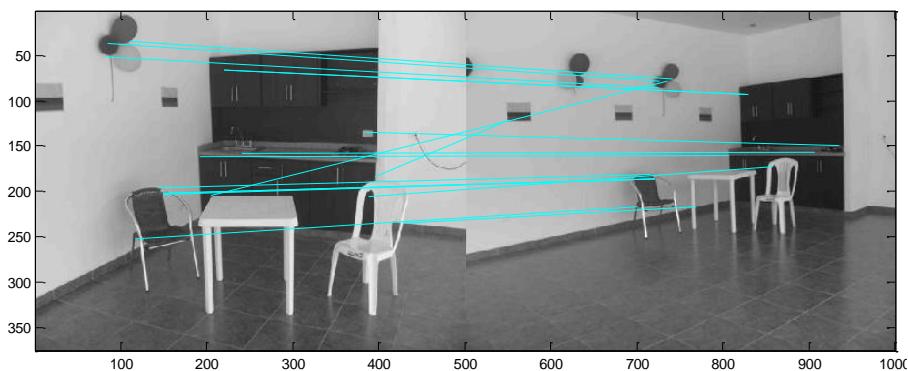


Fuente: Autor

Para la comparación mostrada en la Figura 59, se empleó un distRatio = 0,9 y el esquema encontró que entre estas dos imágenes existen 137 puntos coincidentes.

Al hacer nuevamente la comparación y utilizando un distRatio = 0,6, el esquema halló que entre estas dos imágenes hay 17 puntos que son coincidentes como se muestra en la Figura 60.

Figura 60. Coincidencia para un distRatio = 0,6



Fuente: Autor

Utilizando un distRatio = 0,3 en la comparación de la Figura 59, el esquema NO halló puntos coincidentes entre estas dos imágenes.

6.2.1.5 Caso 5. Comparación de dos imágenes con un porcentaje alto de cambio de perspectiva. Esta comparación se hace entre las imágenes 000264.jpg y 000354.jpg de la base de datos. Como se observa en la Figura 61, las imágenes tomadas corresponden a una misma escena con un porcentaje alto de cambio de perspectiva.

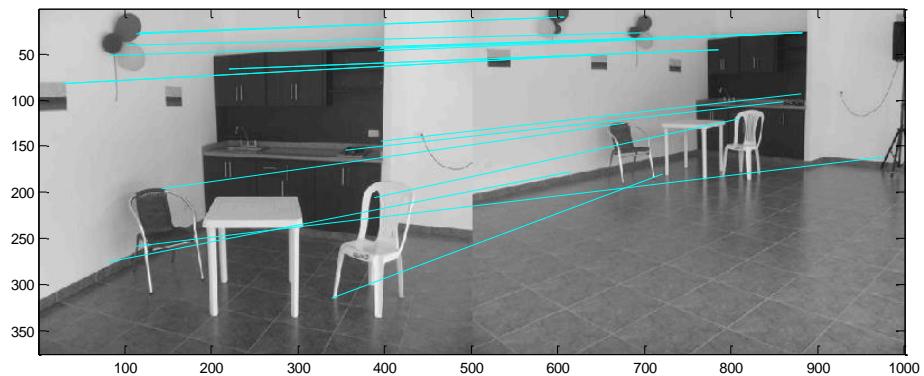
Figura 61. Comparación entre dos imágenes con un porcentaje alto de cambio de perspectiva



Fuente: Autor

Para la comparación mostrada en la Figura 61, se empleó un $\text{distRatio} = 0,9$ y el esquema encontró que entre estas dos imágenes existen 144 puntos coincidentes. Al hacer nuevamente la comparación y utilizando un $\text{distRatio} = 0,6$, el esquema halló que entre estas dos imágenes hay 18 puntos que son coincidentes como se muestra en la Figura 62.

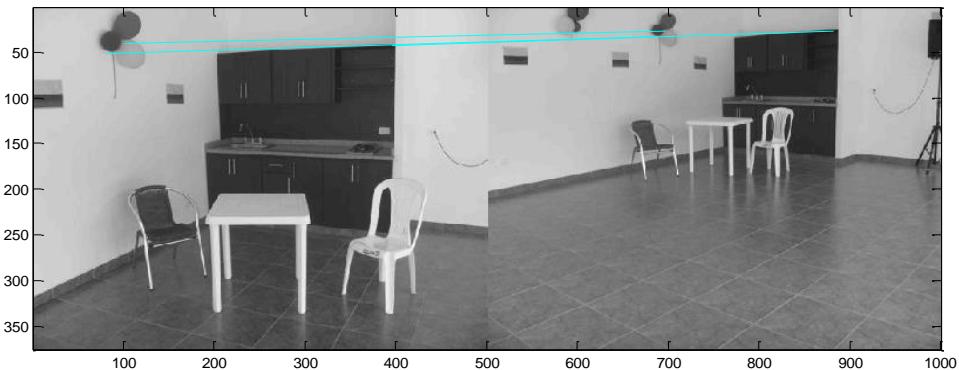
Figura 62. Coincidencia para un $\text{distRatio} = 0,6$



Fuente: Autor

Utilizando un distRatio = 0,3 en la comparación de la Figura 61, el esquema halló 3 puntos coincidentes entre estas dos imágenes.

Figura 63. Coincidencia para un distRatio = 0,3



Fuente: Autor

6.2.1.6 Caso 6: Localización de un objeto que hace parte de una imagen. Para este caso se utilizaron dos métodos de extracción de puntos de interés, el método de extracción de características SIFT y el método de extracción de características SURF.

❖ **Método de Extracción de Características SIFT:**

Ejemplo 1: En este ejemplo se toma la imagen 010544.jpg de la base de datos que corresponde a un cuadro de la virgen y utilizando el editor de imágenes PAINT, se extrae un objeto que pertenece a la imagen, en este caso la cara de la virgen correspondiente a la imagen 010544_1 como se puede apreciar en la Figura 64.

Figura 64. Comparación entre una imagen y un objeto de ella



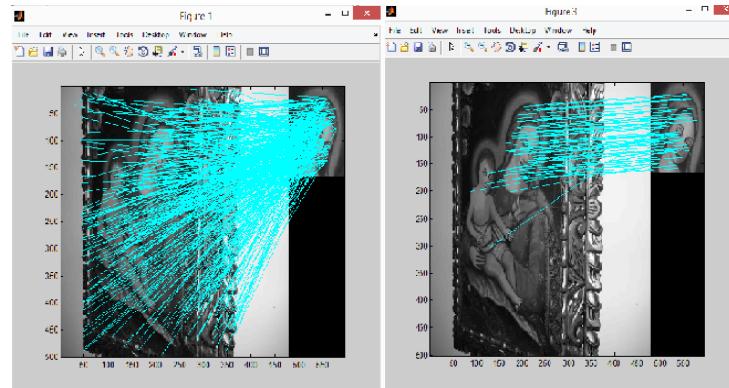
Fuente: Autor

Al hacer la comparación de las dos imágenes utilizando valores de `distRatio = 0,6` y `0,9` se observa que se detectan demasiados puntos que aparentemente el esquema los toma como “coincidentes”, pero este resultado no es confiable para un proceso de reconocimiento porque presenta un número elevado de coincidencias que no corresponden a la situación real.

En la parte izquierda de la Figura 65, se muestra el resultado obtenido al utilizar un `distRatio = 0,9` donde el esquema detecta 433 puntos como coincidentes, lo cual muestra la ineficiencia de la comparación pues señala como coincidencias algunos puntos de la cara de la virgen con los puntos del borde del marco del cuadro.

En la parte derecha de la Figura 65 se puede observar el resultado obtenido al emplear un `distRatio = 0,6`, en este caso, el esquema muestra 71 puntos coincidentes presentando una disminución de los puntos que aparentemente son coincidentes, sin embargo todavía se puede notar que el esquema presenta falencias porque hay puntos del cuadro como parte de la cara del niño y de las manos que no coinciden con la cara de la virgen de la otra imagen.

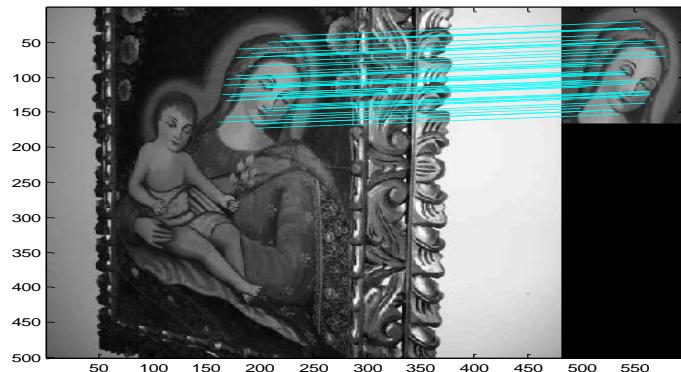
Figura 65. Coincidencia para diferentes valores de distRatio



Fuente: Autor

En la Figura 66 se muestran los resultados con un $\text{distRatio} = 0,3$ donde el esquema detectó 44 puntos coincidentes; al observar la Figura, se puede notar que se identifica completamente el rostro de la virgen.

Figura 66. Coincidencia para valores de distRatio = 0,3

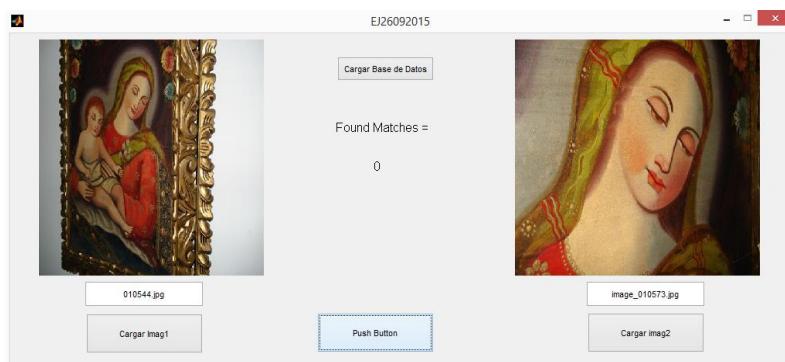


Fuente: Autor

De los resultados anteriores se puede deducir que previamente a la implementación de un procedimiento de reconocimiento o recuperación de imágenes, el hacer una correcta selección del valor de distRatio , haría más eficiente dicho procedimiento.

Ejemplo 2: Para este ejemplo se toma la imagen 010544.jpg, correspondiente al cuadro de la virgen, y como objeto, se utiliza otra imagen: la image_010573.jpg tomada de forma independiente sobre la cara de la virgen del mismo cuadro, pero a mayor escala, con diferente grado de iluminación y diferente perspectiva. En la Figura 67 se muestran las imágenes a comparar en la GUI.

Figura 67. Comparación entre una imagen y un objeto de ella obtenido de manera independiente



Fuente: Autor

En la Figura 68 se pueden apreciar los resultados obtenidos al emplear un $\text{distRatio} = 0,4$ en donde el esquema detectó 3 puntos coincidentes correctamente hallados.

Figura 68. Coincidencia para valores de $\text{distRatio} = 0,4$

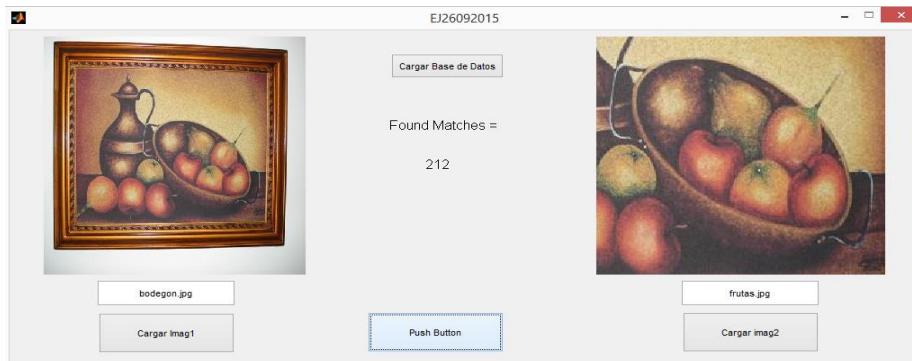


Fuente: Autor

Para el caso mostrado en la Figura 67, se hicieron pruebas para un distRatio = 0,9 se obtuvieron 363 puntos mal llamados coincidentes pues hacen comparaciones aleatorias e incoherentes entre las dos imágenes. Al hacer una prueba para un distRatio = 0,6 se obtuvieron 15 puntos coincidentes con resultados más acertados y para pruebas con un distRatio = 0,3 no se obtuvo ningún punto coincidente.

Ejemplo 3: En este ejemplo se hizo un procedimiento similar al Ejemplo 1, pero empleando la imagen de un bodegón con una jarra, vasija y frutas denominada bodegon.jpg y como objeto una imagen de vasija con frutas llamada frutas.jpg como se ilustra en la Figura 69.

Figura 69. Comparación entre una imagen y un objeto de ella

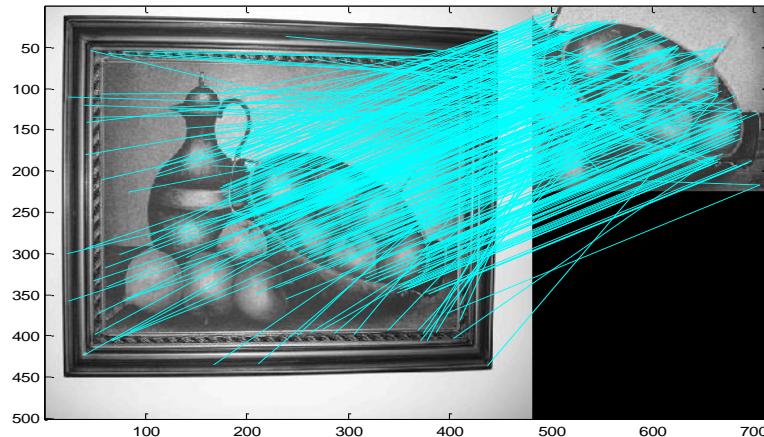


Fuente: Autor

El esquema detectó en la imagen bodegon.jpg 913 puntos de interés y en la imagen frutas.jpg 278 puntos de interés, al comparar dichos puntos, se obtienen 362 puntos coincidentes, lo cual muestra la ineficiencia de la comparación pues señala como coincidencias algunos puntos de la vasija con frutas con el borde del bodegón.

En la Figura 70 se muestran los resultados de la comparación obtenida al utilizar un distRatio = 0,9.

Figura 70. Puntos coincidentes para un distRatio = 0,9

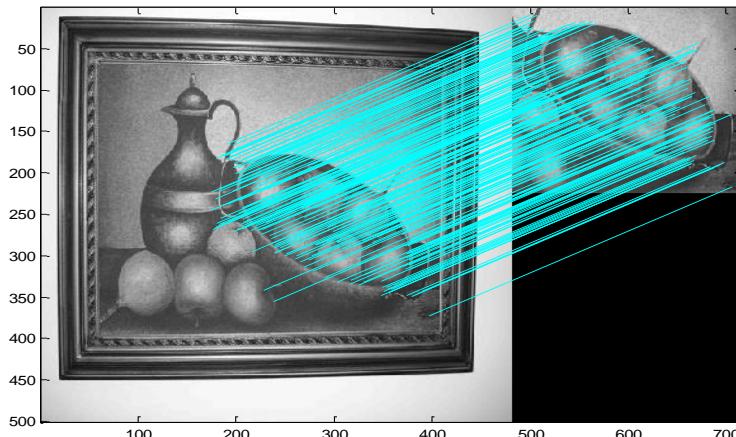


Fuente: Autor

El resultado obtenido al emplear un $\text{distRatio} = 0,6$ fue de 231 puntos coincidentes presentando una disminución de los puntos que aparentemente son coincidentes, sin embargo todavía se puede notar que el esquema sigue presentando falencias al realizar la comparación.

En la Figura 71 se muestran los resultados con un $\text{distRatio} = 0,3$ donde el esquema detectó 212 puntos coincidentes; al observar la Figura, se puede notar la eficiencia del esquema al identificar completamente la vasija con frutas.

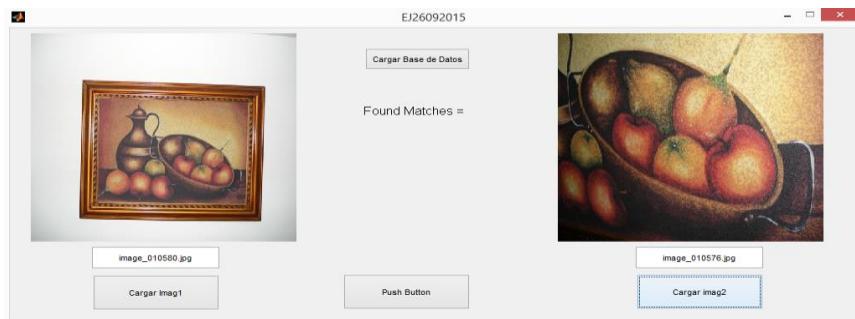
Figura 71. Puntos coincidentes para un distRatio = 0,3



Fuente: Autor

Ejemplo 4: En este ejemplo se realiza un procedimiento similar al del Ejemplo 2 utilizando las imágenes `image_010580.jpg` y `Image_ 010576.jpg` como se muestra en la Figura 72.

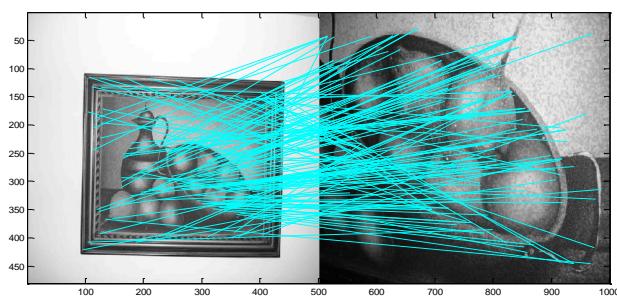
Figura 72. Comparación entre una imagen y un objeto de ella obtenido de manera independiente



Fuente: Autor

El esquema detectó en la imagen `image_010580.jpg` 682 puntos de interés y en la imagen `image_010576.jpg` 1403 puntos de interés, al comparar dichos puntos, se obtienen 146 puntos coincidentes empleando un `distRatio = 0,9`. Estos resultados muestran la ineficiencia de la comparación pues señala como coincidencias algunos puntos de la vasija con frutas con el borde del bodegón como se observa en la Figura 73.

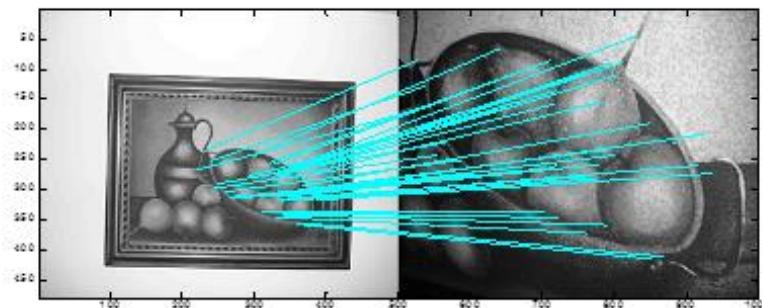
Figura 73. Puntos Coincidentes para un `distRatio = 0,9`



Fuente: Autor

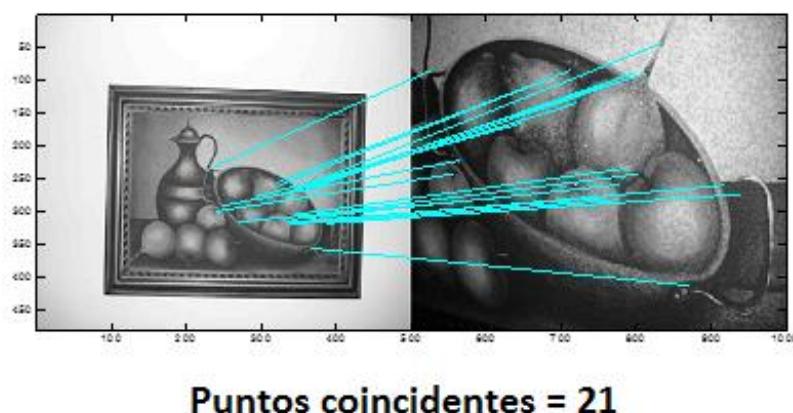
En las Figuras 74, 75 y 76 se pueden apreciar los resultados obtenidos de la comparación al emplear diferentes valores de distRatio.

Figura 74. Puntos coincidentes para un distRatio = 0,7



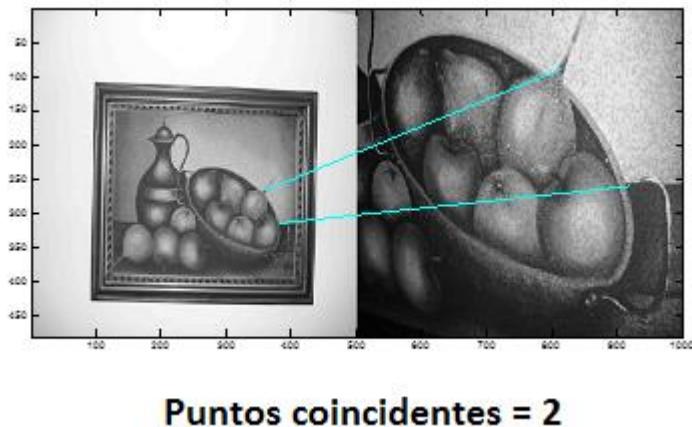
Fuente: Autor

Figura 75. Puntos coincidentes para un distRatio = 0,6



Fuente: Autor

Figura 76. Puntos coincidentes para un distRatio = 0,3



Fuente: Autor

Al analizar los resultados obtenidos en las pruebas anteriores, se puede deducir que el mejor resultado para efectos de reconocimiento se obtuvo al emplear un valor de distRatio = 0,7.

❖ **Método de Extracción de Características SURF:**

Ejemplo 1: Para este ejemplo se realizó un procedimiento similar empleando las mismas imágenes del *Ejemplo 1 del Método de Extracción de Características SIFT* descrito anteriormente. En este ejemplo se utilizó el Método de Extracción de Características SURF para comparar los resultados obtenidos por ambos métodos sobre el mismo conjunto de imágenes y bajo las mismas condiciones.

En este ejemplo se toma la imagen 010544.jpg de la base de datos y la imagen 010544_1 como se puede apreciar en la Figura 77.

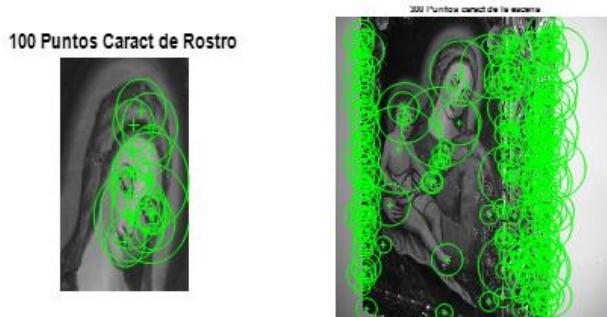
Figura 77. Imagen del rostro e imagen de la escena



Fuente: Autor

En la Figura 76 se muestra la extracción de un número determinado de puntos característicos de la imagen 010544_1, la cual contiene el objeto, que en este caso es la cara de la virgen y también se extraen un número determinado de puntos característicos de la imagen 010544.jpg que corresponde al cuadro de la virgen.

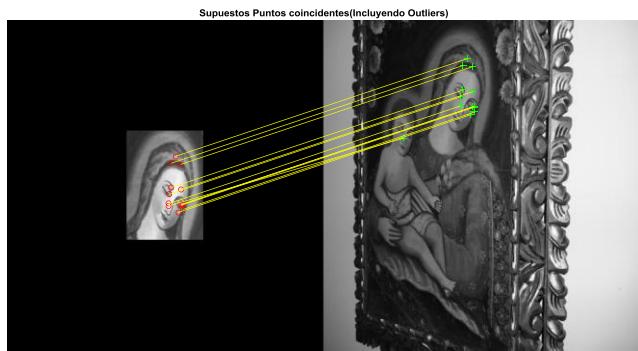
Figura 78. Puntos característicos obtenidos con el método de extracción de características SURF



Fuente: Autor

En la Figura 79 se puede observar el resultado de la comparación para valores altos del parámetro distRatio.

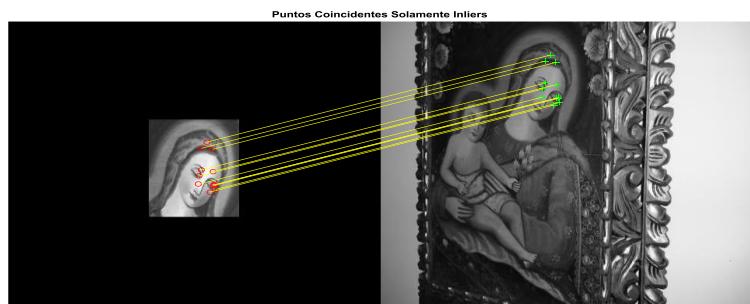
Figura 79. Puntos coincidentes en las 2 imágenes para valores altos del parámetro distRatio.



Fuente: Autor

En la Figura 80 se puede observar el resultado de la comparación para valores bajos del parámetro distRatio.

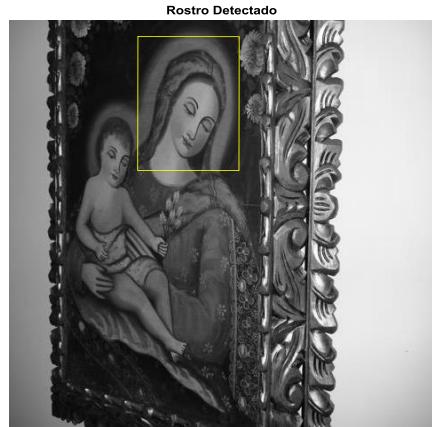
Figura 80. Puntos coincidentes en las imágenes para valores bajos del parámetro distRatio.



Fuente: Autor

Al aplicar un algoritmo de demarcación de la zona de mayor coincidencia entre las imágenes a comparar, se muestra que el esquema detectó de manera eficiente la zona de mayor coincidencia, en este caso detectó la cara de la virgen en el cuadro que contiene a la virgen como se ilustra en la Figura 81.

Figura 81. Rostro detectado en la imagen de la escena



Fuente: Autor

Ejemplo 2: Para este ejemplo se realizó un procedimiento similar empleando las mismas imágenes del *Ejemplo 2 del Método de Extracción de Características SIFT* descrito anteriormente. En este ejemplo se utilizó el Método de Extracción de Características SURF para comparar los resultados obtenidos por ambos métodos.

En este ejemplo se toma la imagen 010544.jpg de la base de datos y la imagen image_010573.jpg como se puede apreciar en la Figura 82.

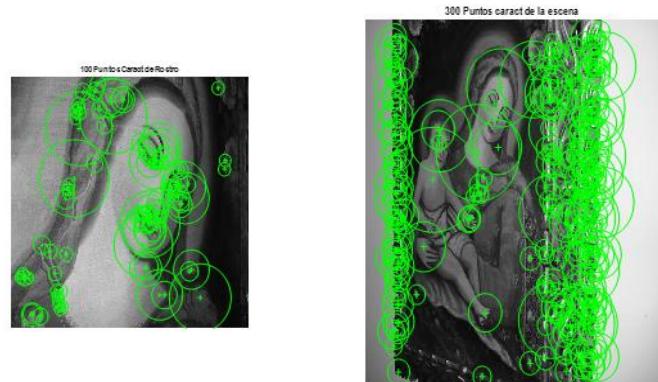
Figura 82. Imagen del objeto a reconocer e imagen del cuadro de la virgen



Fuente: Autor

En la Figura 83 se muestra la extracción de un número determinado de puntos característicos de la imagen `imagen_010573.jpg`, la cual contiene el objeto a reconocer y también se extraen un número determinado de puntos característicos de la imagen `010544.jpg`.

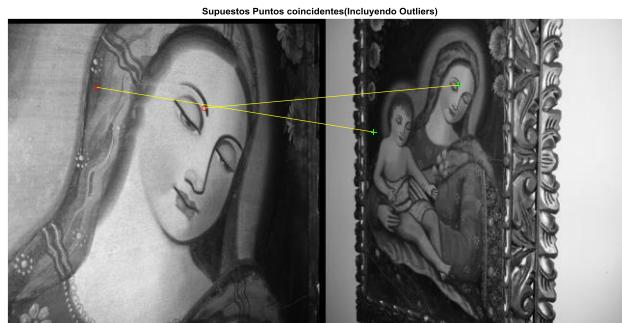
Figura 83. Puntos característicos obtenidos con el método de extracción de características SURF



Fuente: Autor

En la Figura 84 se puede observar el resultado de la comparación para valores bajos del parámetro `distRatio`.

Figura 84. Puntos coincidentes en las 2 imágenes para valores bajos del parámetro `distRatio`.

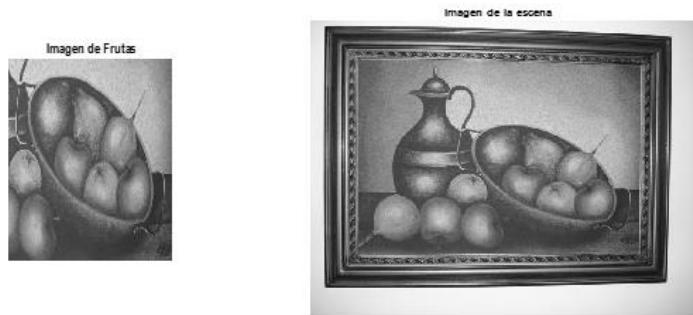


Fuente: Autor

Al aplicar un algoritmo de demarcación de la zona de mayor coincidencia entre las imágenes a comparar, se muestra que ***el esquema no fue capaz de detectar el rostro en la imagen.***

Ejemplo 3: En este ejemplo se hizo un procedimiento similar al *Ejemplo 1 del Método de Extracción de Características SURF* pero empleando la imagen de una jarra, vasija y frutas denominada bodegón.jpg y como objeto una imagen de vasija con frutas llamada frutas.jpg como se ilustra en la Figura 85.

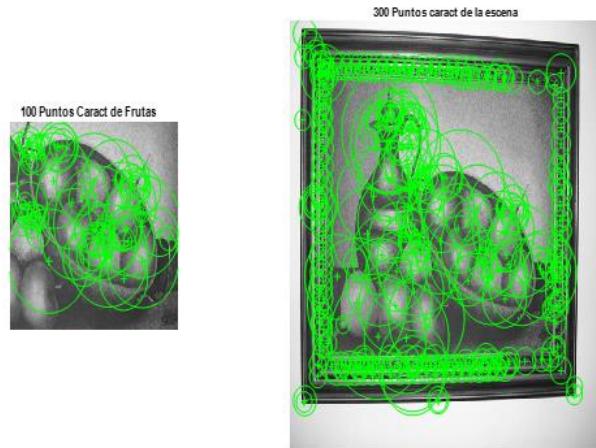
Figura 85. Imagen del objeto a reconocer en este caso Frutas e Imagen de la escena, llamada bodegon.jpg.



Fuente: Autor

En la Figura 86 se muestra la extracción de un número determinado de puntos característicos de la imagen frutas.jpg, la cual contiene el objeto a reconocer, en este caso una vasija con frutas y también se extraen un número determinado de puntos característicos de la imagen bodegon.jpg que corresponde al cuadro con jarra, vasija y frutas.

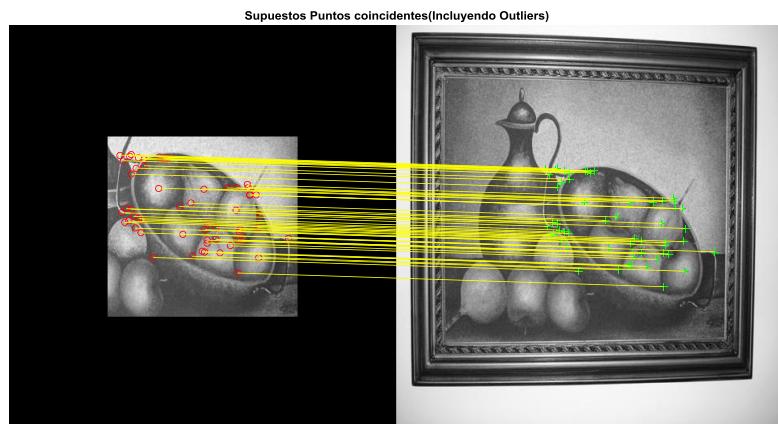
Figura 86. Puntos característicos obtenidos con el método de extracción de características SURF.



Fuente: Autor

En la Figura 87 se puede observar el resultado de la comparación para valores altos del parámetro distRatio.

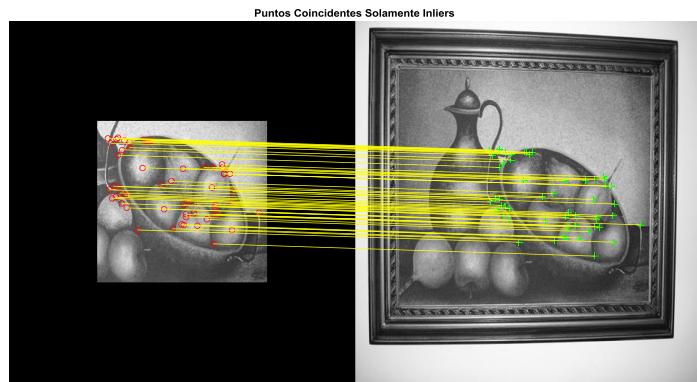
Figura 87. Puntos coincidentes en las imágenes para valores altos del parámetro distRatio.



Fuente: Autor

En la Figura 88 se puede observar el resultado de la comparación para valores bajos del parámetro distRatio.

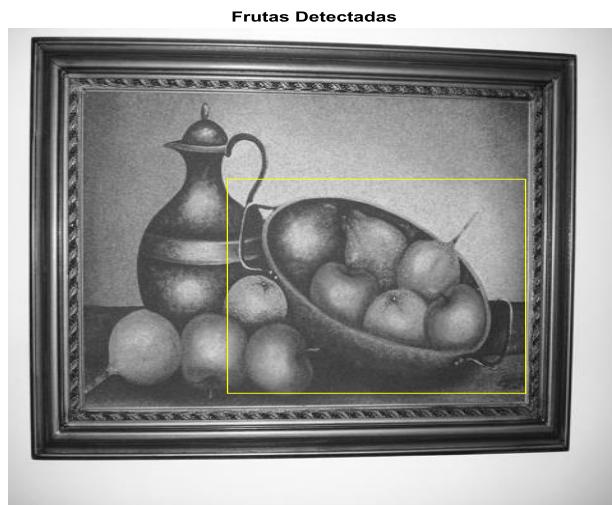
Figura 88. Puntos coincidentes en las imágenes para valores bajos del parámetro distRatio.



Fuente: Autor

Al aplicar un algoritmo de demarcación de la zona de mayor coincidencia entre las imágenes a comparar, se muestra que el esquema detectó de manera eficiente la zona de mayor coincidencia, en este caso detectó la vasija con frutas en el cuadro que contiene la jarra la vasija y las frutas como se puede apreciar en la Figura 89.

Figura 89. Frutas detectadas en la imagen bodegón

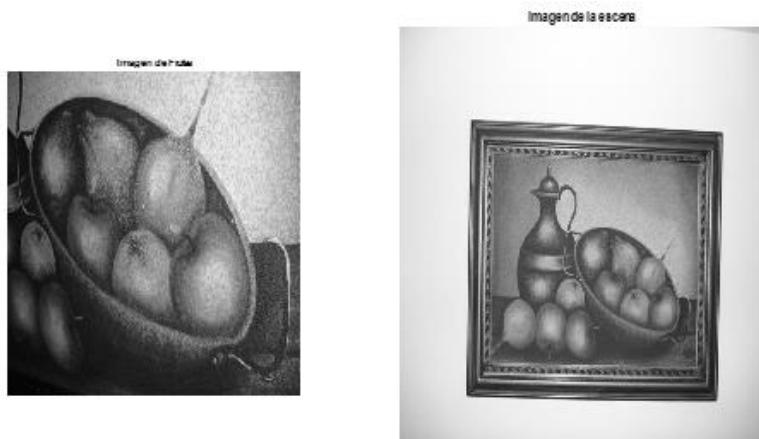


Fuente: Autor

Ejemplo 4: Para este ejemplo se realizó un procedimiento similar al del *Ejemplo 2* empleando las mismas imágenes utilizadas en el *Método de Extracción de Características SURF* descrito anteriormente. En este ejemplo se utilizó el Método de Extracción de Características SURF para comparar los resultados obtenidos por ambos métodos.

En este ejemplo se toma la imagen *image_010580.jpg* de la base de datos y la imagen *image_010576.jpg*, la cual, contiene un objeto de la imagen en este caso una vasija que contiene frutas como se puede apreciar en la Figura 90.

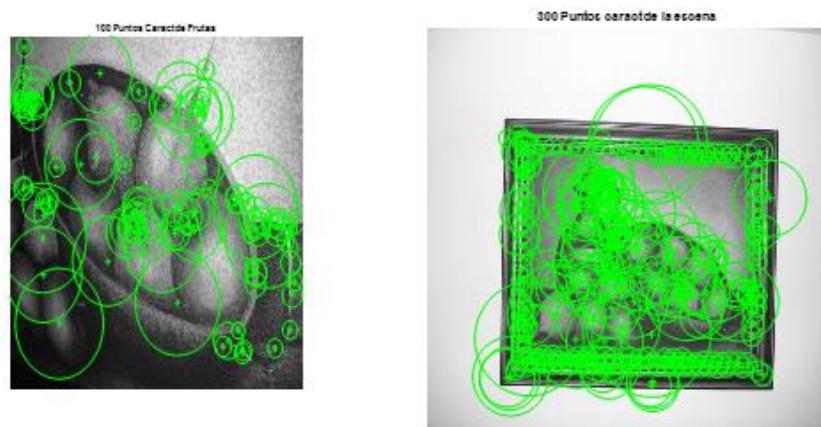
Figura 90. Imagen de una vasija con frutas e Imagen bodegón



Fuente: Autor

En la Figura 91 se muestran los resultados obtenidos mediante la extracción de un número determinado de puntos característicos de la imagen *frutas.jpg*, la cual contiene el objeto a reconocer, que en este caso es una jarra con frutas y también se extraen un número determinado de puntos característicos de la imagen *bodegon.jpg* que corresponde al cuadro con jarra, vasija y frutas.

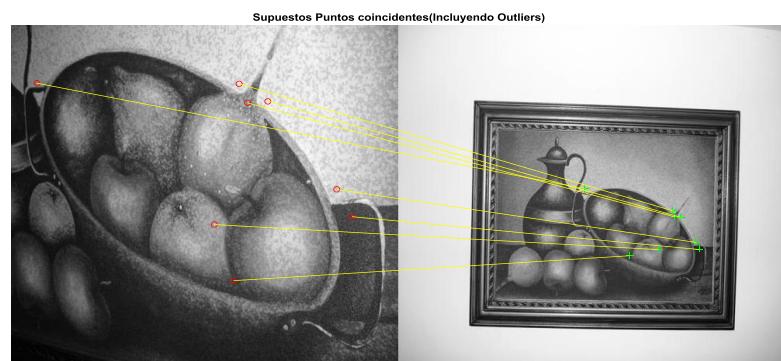
Figura 91. Puntos característicos obtenidos con el método de extracción de características SURF



Fuente: Autor

En la Figura 92 se puede observar el resultado de la comparación para valores altos del parámetro distRatio.

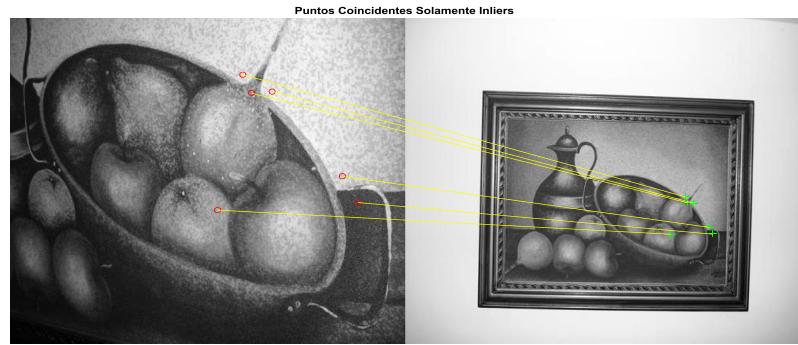
Figura 92. Puntos coincidentes en las imágenes para valores altos del parámetro distRatio



Fuente: Autor

En la Figura 93 se puede observar el resultado de la comparación para valores bajos del parámetro distRatio.

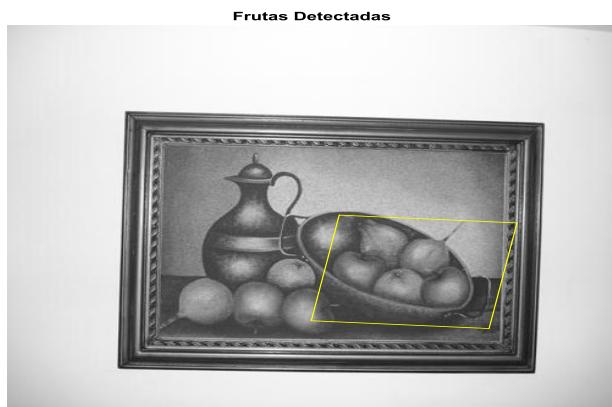
Figura 93. Puntos coincidentes en las imágenes para valores bajos del parámetro distRatio



Fuente: Autor

Al aplicar un algoritmo de demarcación de la zona de mayor coincidencia entre las imágenes a comparar, se muestra que el esquema **detectó solamente una parte de la zona de mayor coincidencia, lo cual demuestra que el esquema no funcionó correctamente** como se puede apreciar en la Figura 94.

Figura 94. Frutas detectadas en la imagen de la escena



Fuente: Autor

Ejemplo 5: En este ejemplo se toma la imagen llamada image_010580.jpg de la base de datos que contiene varios objetos entre ellos una jarra, vasija y frutas y las imágenes denominadas image_010576.jpg y jarrita.jpg. La primera imagen contiene un objeto que en este caso es una vasija con frutas y la segunda imagen contiene como objeto la parte superior de la jarra como se puede apreciar en la Figura 95.

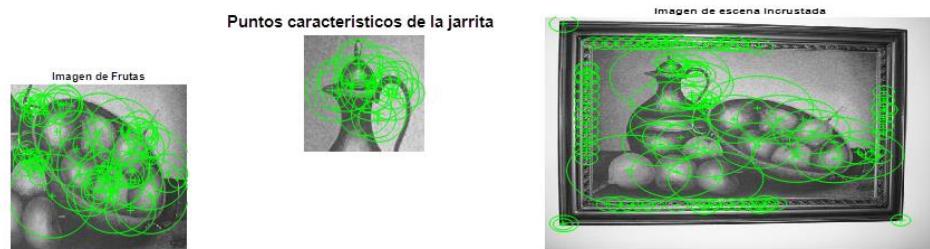
Figura 95. Imágenes que contienen los objetos a reconocer: Vasija con frutas e Imagen que contiene una jarra y la imagen que contiene toda la escena



Fuente: Autor

En la Figura 96 se muestran los puntos característicos obtenidos de las imágenes de la Figura 95 empleando el método de extracción de características SURF.

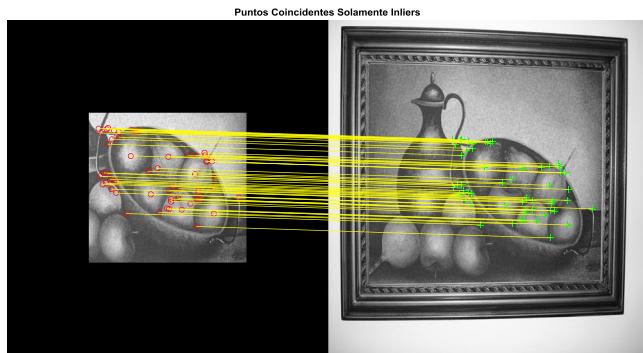
Figura 96. Puntos característicos de las imágenes de la Figura 95



Fuente: Autor

En la Figura 97 se muestran los puntos coincidentes entre la imagen que contiene toda la escena con los objetos y la imagen que contiene el objeto de la vasija de frutas.

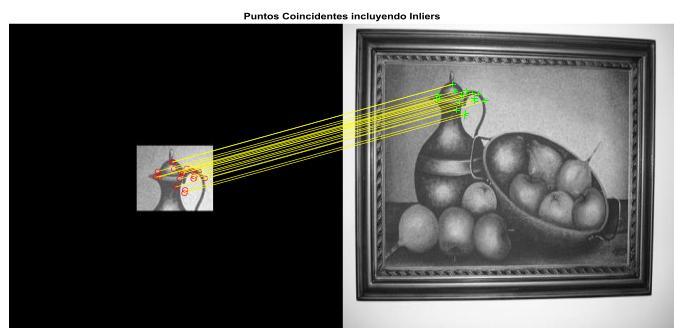
Figura 97. Puntos coincidentes entre la imagen que contiene toda la escena y la imagen que contiene el objeto vasija con frutas



Fuente: Autor

En la Figura 98 se muestran los puntos coincidentes entre la imagen que contiene toda la escena con los objetos y la imagen que contiene el objeto de la jarra.

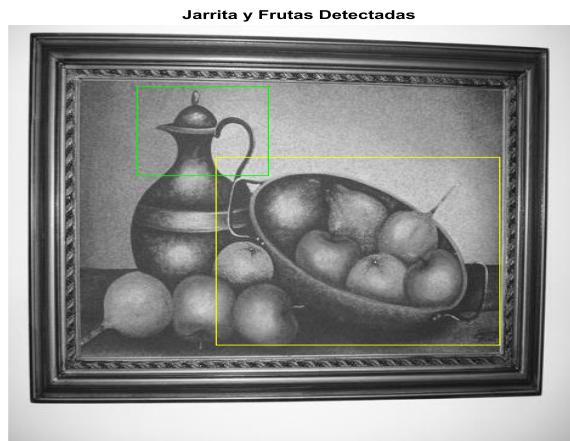
Figura 98. Puntos coincidentes entre la imagen que contiene toda la escena y la imagen que contiene el objeto de la jarra.



Fuente: Autor

Al aplicar un algoritmo de demarcación de la zona de mayor coincidencia entre las imágenes a comparar, se muestra que el esquema detectó de manera eficiente las zonas de mayor coincidencia, en la imagen de la escena, es decir la vasija con frutas y la jarra. Los resultados se muestran en la Figura 99.

Figura 99. Visualización de ambos objetos en la imagen de la escena



Fuente: Autor

6.2.2 Conclusiones del esquema de detección de objetos. De las pruebas realizadas con el esquema de detección de objetos se concluye que la comparación de imágenes a partir de la coincidencia de puntos de interés, utilizando el descriptor SIFT, no es suficiente para reconocer imágenes en general.

Si las imágenes son similares, es decir tienen muchas características en común con respecto a la posición, rotación, iluminación y perspectiva pueden tener un número considerable de puntos coincidentes, los cuales pueden de cierta manera contribuir al proceso de reconocimiento.

Como la comparación entre puntos de interés se hace utilizando el producto punto entre vectores (distRatio), dicho parámetro es crítico al momento de realizar la comparación de dichos puntos y su respectiva coincidencia. De las pruebas se concluye que para un valor mayor de distRatio, es menor el filtro de depuración de puntos coincidentes, con lo cual el esquema tiene menos restricciones para elegir los puntos que se consideran similares, dando como resultado un mayor número de puntos coincidentes, pero un peor resultado de clasificación de puntos coincidentes, que luego podrían ser fundamentales para tareas de reconocimiento de imágenes.

6.3 ESQUEMA DE BÚSQUEDA

En el esquema descrito anteriormente, se hallaron objetos específicos que pertenecían a una sola imagen. En este esquema se realizó la búsqueda de objetos específicos en una base de datos con muchas imágenes.

6.3.1 Esquema de búsqueda de objetos específicos, en un conjunto de imágenes a partir de su grado de coincidencia. Uno de los sistemas utilizados en visión por computador es el denominado esquema de búsqueda de objetos específicos o esquema de recuperación de imágenes basado en su contenido. Estos esquemas se utilizan para recuperar imágenes de una colección de imágenes que son similares a la imagen de referencia o de consulta. La aplicación de esquemas de este tipo se puede encontrar en muchas áreas, tales como: búsqueda de productos en la web, sistemas de vigilancia, identificación visual de lugares específicos, etc.

El esquema de recuperación de imágenes basado en contenido se fundamenta en el grado de coincidencia de los puntos de interés de los objetos de las imágenes, con una base de datos de una colección de imágenes, a partir de una imagen de consulta, la cual contiene el objeto a reconocer.

En este esquema se utiliza un índice de aparición de puntos de interés coincidentes, el cual define el número de características de vecinos más cercanos de cada imagen en la colección de imágenes. La comparación entre la imagen de consulta y el índice de apariciones, proporciona aquellas imágenes más similares a la imagen de consulta.

6.3.1.1 Ejemplo 1. Este ejemplo muestra cómo buscar una escena en una colección de imágenes, dada una imagen que defina dicha escena. Se muestra cómo encontrar de manera eficiente los puntos vecinos más cercanos por comparación de puntos de características utilizando el descriptor SURF (SURF es un método de correspondencia robusto a rotaciones y cambios de escala) aplicado a la imagen que define el objeto de la escena y a la colección de imágenes.

❖ **Paso 1: Preparar la colección de imágenes para la búsqueda:**

El primer paso del esquema de búsqueda consiste en leer en un conjunto de imágenes la escena que se quiere buscar. Este esquema se implementó de tal forma que la primera imagen de la colección tuviera la escena que se deseaba buscar, la cual, para este ejemplo es un quiosco como se puede apreciar en la Figura 100.

Figura 100. Colección de imágenes del quiosco tomada desde diferentes puntos de vista

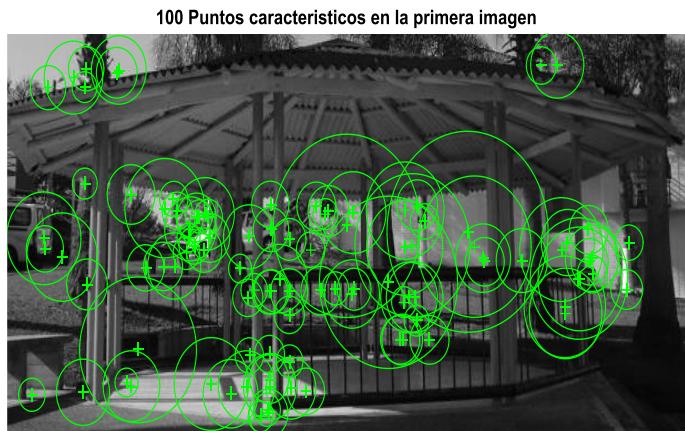


Fuente: Autor

❖ **Paso 2: Detectar puntos característicos en la colección de imágenes:**

Con este paso se detectan y se muestran los puntos característicos de la primera imagen de la colección de imágenes como se ilustra en la Figura 101.

Figura 101. Puntos característicos en la imagen que contiene el objeto a buscar



Fuente: Autor

Paso seguido, se hace la detección y extracción de puntos característicos de todas las imágenes de la colección.

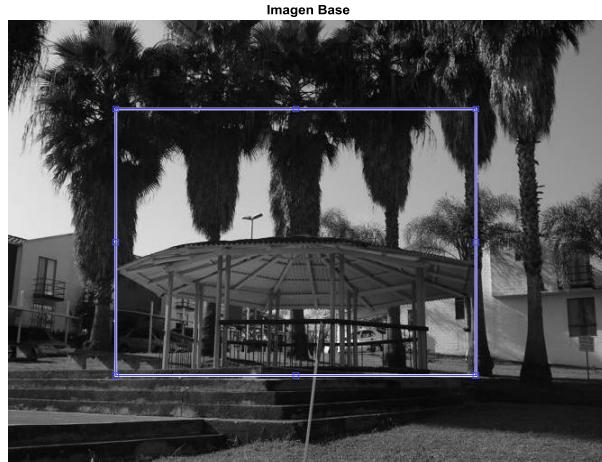
❖ **Paso 3: Construir la base de datos de las características:**

Por medio de este paso se combinan todas las características de cada imagen en una base de datos de características y se implementa el procedimiento de coincidencia de características de los objetos de las imágenes.

❖ **Paso 4: Seleccionar la imagen que contiene el objeto de consulta:**

En este paso se carga la imagen que contiene el objeto a buscar y se selecciona el objeto especificándolo en un cuadro delimitador. Para este caso, la imagen contiene un quiosco tomado desde una perspectiva diferente como se puede apreciar en la Figura 102. Esta imagen no hace parte del conjunto de imágenes.

Figura 102. Imagen con el recuadro que delimita el objeto a identificar

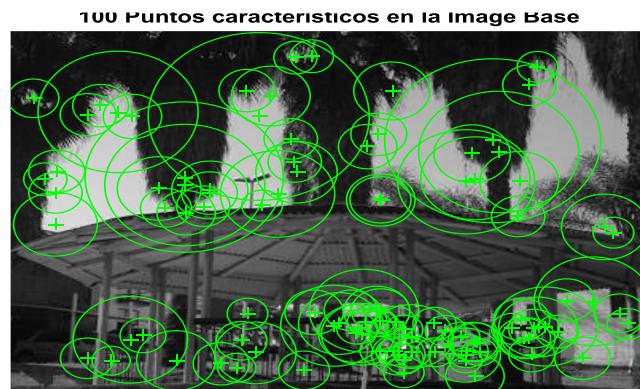


Fuente: Autor

❖ **Paso 5: Detectar los puntos característicos en la imagen de consulta:**

Este paso detecta y visualiza los puntos característicos en la imagen de consulta como se muestra en la Figura 103.

Figura 103. Puntos característicos en la imagen a buscar

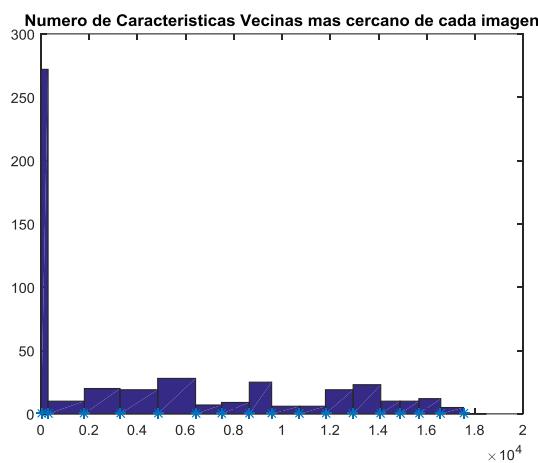


Fuente: Autor

❖ **Paso 6: Buscar la imagen que tiene el objeto a identificar en las imágenes de la colección:**

En este paso, para todos los puntos característicos de la imagen que contiene el objeto a buscar, se le hallan los puntos característicos vecinos más cercanos en el conjunto de imágenes de la colección. La Figura 104 muestra la gráfica que define el grado de coincidencia entre la imagen que contiene el objeto a buscar (primera imagen de la colección) y el resto de las imágenes del conjunto de imágenes.

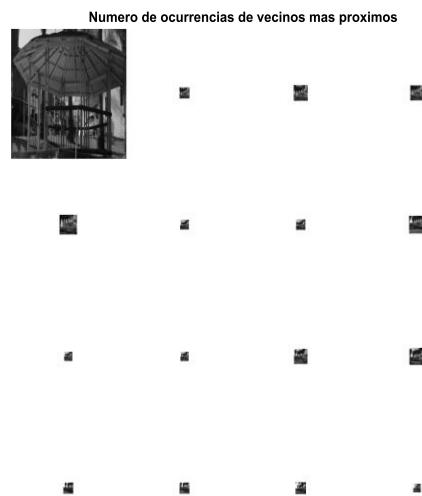
Figura 104. Gráfica que define el grado de coincidencia entre las imágenes de la colección



Fuente: Autor

La Figura 105 muestra como el esquema define el grado de coincidencia de las imágenes de la colección a través del tamaño y de la ubicación en la figura de la colección de imágenes. Como se puede observar en esta Figura, el tamaño de cada imagen es proporcional al número de puntos característicos que coinciden. Para este ejemplo, la imagen que tiene mayor coincidencia es la primera.

Figura 105. Grado de coincidencia de las imágenes de la colección a través del tamaño



Fuente: Autor

❖ **Paso 7. Eliminar las falsas coincidencias mediante test de distancia:**

En este paso, se eliminan las falsas coincidencias detectadas en la colección de imágenes mediante la comparación de las distancias del primero y segundo vecino más cercano. En la Figura 106 se muestra como al eliminar las falsas coincidencias, la imagen cuyo grado de coincidencia es mayor es la que presenta mayor tamaño.

Figura 106. Número de características que coinciden



Fuente: Autor

6.3.1.2 Ejemplo 2. Este ejemplo muestra cómo buscar en una colección de imágenes, aquella que contiene el objeto de la imagen de búsqueda. Se muestra cómo encontrar de manera eficiente el vecino más cercano por coincidencia de puntos vecinos utilizando el descriptor SURF.

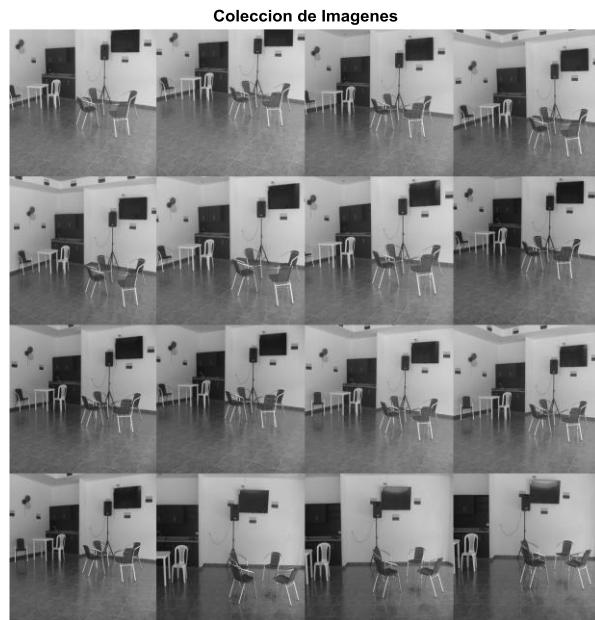
Los pasos a realizar en la ejecución de este procedimiento son:

❖ **Paso 1: Preparar la colección de imágenes para la búsqueda:**

El primer paso del esquema de búsqueda consiste en leer en un conjunto de imágenes la escena que se quiere buscar, para este caso, se buscó un objeto dentro de una colección de imágenes que contenían el mismo objeto visto desde diferentes perspectivas, no se excluyó la posibilidad de capturar zonas ocultas u ocluidas.

En la Figura 107 se muestra la colección de imágenes para este ejemplo.

Figura 107. Colección de imágenes del Ejemplo 2

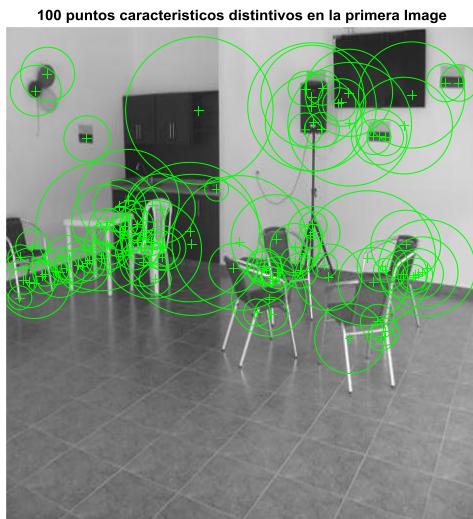


Fuente: Autor

❖ **Paso 2: Detectar puntos característicos en la colección de imágenes:**

Con este paso se detectan y se muestran los puntos característicos de la imagen de la colección que contiene el objeto a buscar como se ilustra en la Figura 108.

Figura 108. Puntos característicos en la imagen que contiene el objeto a buscar



Fuente: Autor

En este caso es la primera imagen de la colección de imágenes, es decir la imagen 0010733.jpg. De dicha imagen se extraen 100 puntos característicos.

Paso seguido, se hace la detección y extracción de puntos característicos de todas las imágenes de la colección.

❖ **Paso 3: Construir la base de datos de las características:**

Por medio de este paso se combinan todas las características de cada imagen en una base de datos de características y se implementa el procedimiento de coincidencia de características de los objetos de las imágenes.

❖ **Paso 4: Seleccionar la imagen que contiene el objeto de consulta:**

En este paso se carga la imagen que contiene el objeto a buscar y se selecciona el objeto especificándolo en un cuadro delimitador. Para este caso, la imagen contiene una sala de televisión tomada desde una perspectiva diferente como se puede apreciar en la Figura 109. Esta imagen no hace parte del conjunto de imágenes.

Figura 109. Imagen con el recuadro que delimita el objeto a identificar

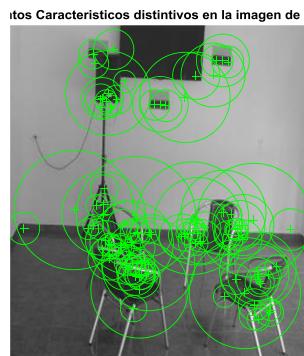


Fuente: Autor

❖ **Paso 5: Detectar los puntos característicos en la imagen de consulta:**

Este paso detecta y visualiza los puntos característicos en la imagen de consulta como se muestra en la Figura 110.

Figura 110. Puntos característicos en la imagen a buscar

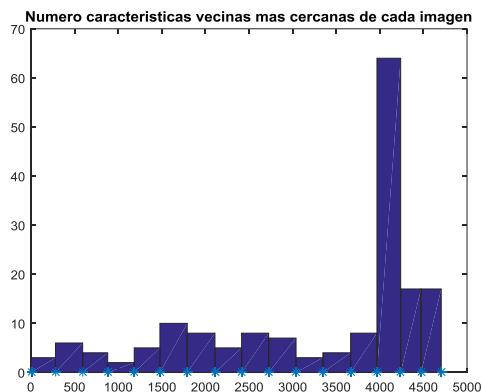


Fuente: Autor

❖ **Paso 6: Buscar la imagen que tiene el objeto a identificar en las imágenes de la colección:**

En este paso, para todos los puntos característicos de la imagen que contiene el objeto a buscar, se le hallan los puntos característicos vecinos más cercanos en el conjunto de imágenes de la colección. La Figura 111 muestra la gráfica que define el grado de coincidencia entre la imagen que contiene el objeto a buscar (primera imagen de la colección) y el resto de las imágenes del conjunto de imágenes.

Figura 111. Gráfica que define el grado de coincidencia entre las imágenes de la colección

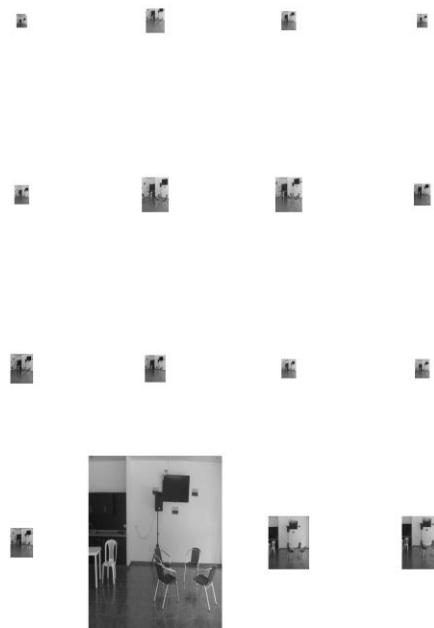


Fuente: Autor

La Figura 112 muestra como el esquema define el grado de coincidencia de las imágenes de la colección a través del tamaño y de la ubicación en la figura de la colección de imágenes. Como se puede observar en esta Figura, el tamaño de cada imagen es proporcional al número de puntos característicos que coinciden. Para este ejemplo, la imagen que tiene mayor coincidencia es la imagen ubicada en la posición catorce de la colección de imágenes.

Figura 112. Grado de coincidencia de las imágenes de la colección a través del tamaño.

Tamaño de la imagen y numero de ocurrencias de vecinos cercanos



Fuente: Autor

❖ **Paso 7. Eliminar las falsas coincidencias mediante test de distancia:**

En este paso, se eliminan las falsas coincidencias detectadas en la colección de imágenes mediante la comparación de las distancias del primero y segundo vecino más cercano. En la Figura 113 se muestra como al eliminar las falsas coincidencias, la imagen cuyo grado de coincidencia es mayor es la que presenta mayor tamaño.

Figura 113. Grafica que muestra el tamaño de la imagen en relación con el grado de coincidencia. Mayor grado de coincidencia, mayor tamaño.

Tamano de la imagen y numero de caracteristicas coincidentes



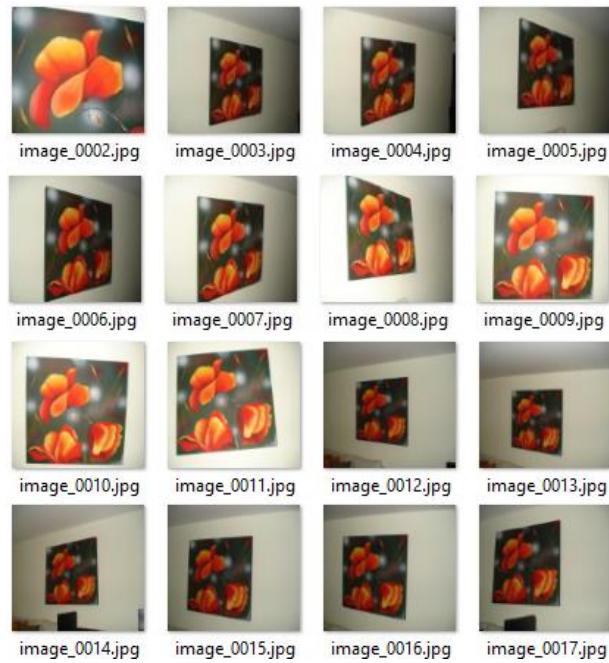
Fuente: Autor

La Figura 113 muestra la relación del tamaño de la imagen con relación al valor de la coincidencia. Se notan mejores resultados después de retirados los valores atípicos.

6.3.1.3 Ejemplo 3. En este ejemplo se toma una flor que hace parte de un cuadro de varias flores y la colección de imágenes es el cuadro visto desde diferentes puntos de vista.

En la Figura 114 se muestra la colección de imágenes sobre las cuales se hizo la búsqueda.

Figura 114. Colección de imágenes



Fuente: Autor

En la Figura 115 se muestran algunas de las imágenes de prueba. Es importante resaltar que dichas imágenes son de la misma escena pero no son parte de la colección de imágenes.

Figura 115. Algunas imágenes que se utilizarán de prueba



Fuente: Autor

En la Figura 116 se muestran las imágenes en escala de grises, como paso previo al procesamiento.

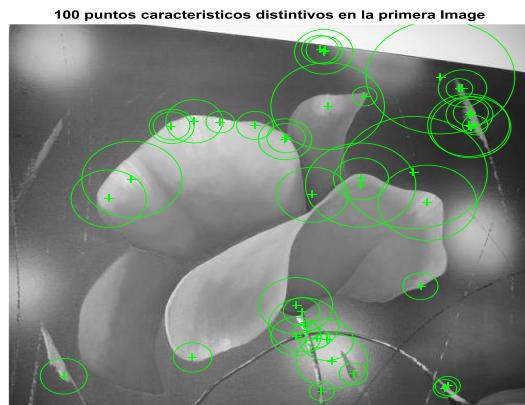
Figura 116. Colección de imágenes en escala de grises



Fuente: Autor

En la Figura 117 se muestran los puntos característicos de una de las imágenes de la colección de imágenes, en este caso se tomó la primera imagen.

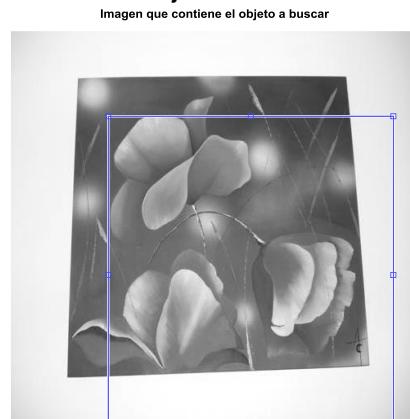
Figura 117. Primera imagen de la colección en escala de grises



Fuente: Autor

En la Figura 118 se muestra la imagen que delimita a través de un recuadro la zona de puntos característicos que se busca en todas las imágenes de la colección.

Figura 118. Imagen que contiene el objeto a identificar delimitado por un recuadro



Fuente: Autor

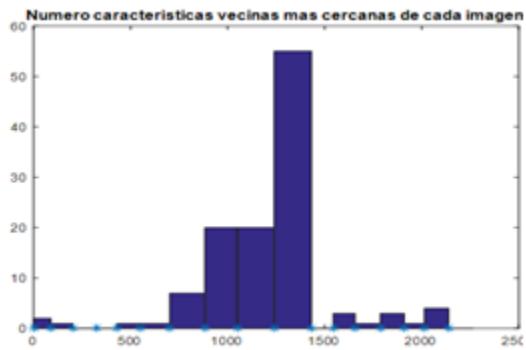
En las Figuras 119 y 120 respectivamente se muestra los puntos característicos de la imagen de consulta y la gráfica que define el grado de coincidencia entre las imágenes de la colección.

Figura 119. Puntos característicos más distintivos en la imagen de consulta



Fuente: Autor

Figura 120. Gráfica que define el grado de coincidencia entre imágenes



Fuente: Autor

En la Figura 121 se muestra que para este ejemplo, la imagen que tiene mayor grado de coincidencia es la imagen ubicada en la posición diez de la colección de imágenes.

Figura 121. Grado de coincidencia de las imágenes de la colección a través del tamaño

Tamaño de la imagen y numero de ocurrencias de vecinos cercano

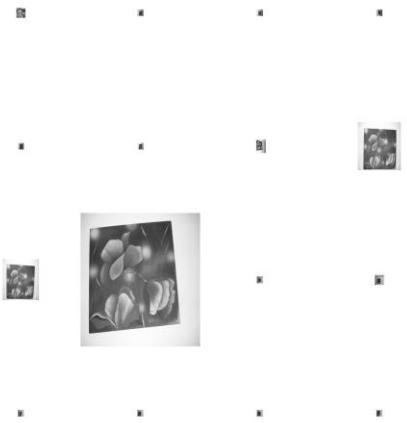


Fuente: Autor

En la Figura 122 se muestra como al eliminar las falsas coincidencias, la imagen cuyo grado de coincidencia es mayor, es la que presenta mayor tamaño.

Figura 122. Gráfica que muestra el tamaño de la imagen en relación con el grado de coincidencia. Mayor grado de coincidencia, mayor tamaño

Tamano de la imagen y numero de características coincidentes



Fuente: Autor

6.3.1.4 Ejemplo 4. En este caso se utilizan como imágenes de la colección, varias imágenes que contienen un cuadro de la virgen visto desde diferentes perspectivas y la escena a buscar es uno de estos cuadros.

La Figura 123 muestra la colección de cuadros sobre los cuales se van a buscar los puntos característicos del objeto de la imagen de consulta.

Figura 123. Colección de imágenes



Fuente: Autor

En la Figura 124 se muestran algunas de las imágenes de prueba, es decir, estas imágenes tienen el objeto de interés que se va a buscar en la colección de imágenes.

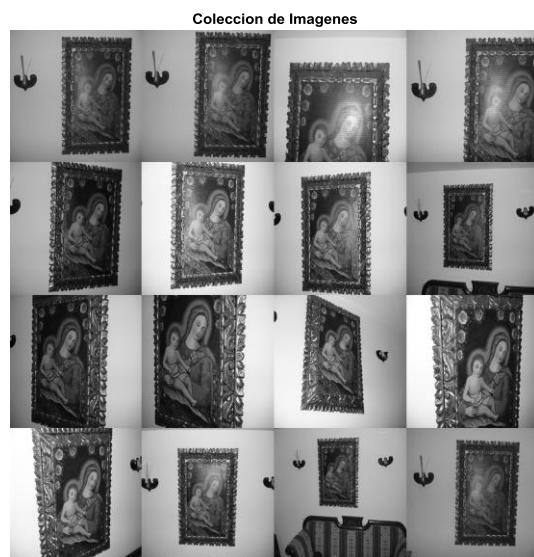
Figura 124. Imágenes de prueba



Fuente: Autor

En la Figura 125 se muestran las imágenes en escala de grises, como paso previo al procesamiento.

Figura 125. Colección de imágenes de vírgenes en escala de grises



Fuente: Autor

En la Figura 126 se muestran los puntos característicos de una de las imágenes de la colección de imágenes, en este caso se tomó la primera imagen.

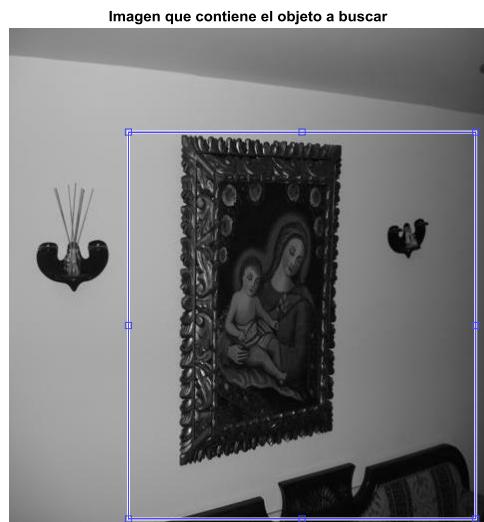
Figura 126. Puntos característicos en la primera imagen de las imágenes de colección



Fuente: Autor

En la Figura 127 se muestra la imagen que delimita a través de un recuadro la zona de puntos característicos que se busca en todas las imágenes de la colección.

Figura 127. Imagen que contiene el objeto a buscar



Fuente: Autor

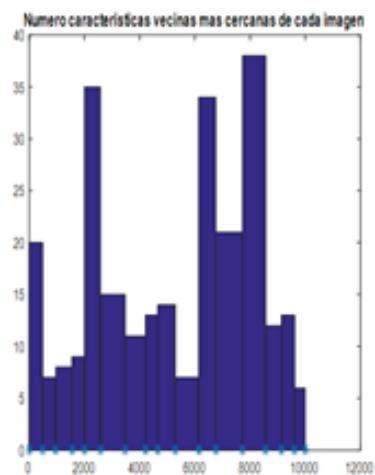
En las Figuras 128 y 129 respectivamente se muestran los puntos característicos de la imagen de consulta y la gráfica que define el grado de coincidencia entre las imágenes de la colección

Figura 128. Puntos característicos más distintivos en la imagen de consulta



Fuente: Autor

Figura 129. Gráfica que define el grado de coincidencia entre imágenes



Fuente: Autor

En la Figura 130 se muestra que para este ejemplo, la imagen que tiene mayor grado de coincidencia es la imagen ubicada en la posición trece de la colección de imágenes.

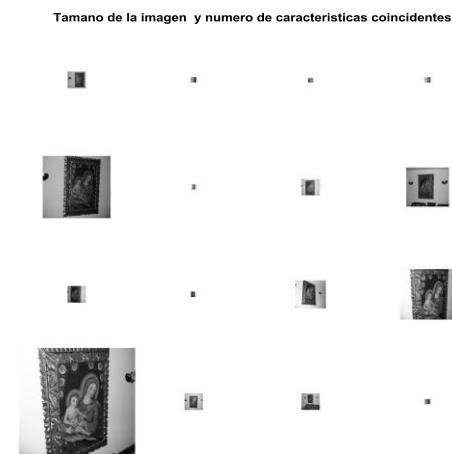
Figura 130. Grado de coincidencia de las imágenes de la colección a través del tamaño



Fuente: Autor

En la Figura 131 se muestra como al eliminar las falsas coincidencias, la imagen cuyo grado de coincidencia es mayor, es la que presenta mayor tamaño.

Figura 131. Gráfica que muestra el tamaño de la imagen en relación con el grado de coincidencia. Mayor grado de coincidencia, mayor tamaño



Fuente: Autor

6.3.2 Conclusiones del esquema de búsqueda. Se hicieron pruebas con varias imágenes obteniéndose muy buenos resultados. De estos ejemplos se concluye que el esquema recupera satisfactoriamente, cualquier objeto que esté en las imágenes de la colección.

6.4 SISTEMA DE CLASIFICACIÓN Y RECONOCIMIENTO DE IMÁGENES

Entre los sistemas de clasificación y reconocimiento de imágenes más utilizados en visión por computador se pueden mencionar los Sistemas de clasificación y reconocimiento de imágenes utilizando Bolsa de Palabras Visuales (BoVW).

Entre los Sistema de clasificación y reconocimiento de imágenes utilizando Bolsa de Palabras visuales (BoVW) se pueden mencionar:

- Sistema de clasificación de imágenes utilizando bolsa de características personalizada. Sistemas CBIR (CBIR- Sistemas basados en contenido para la recuperación de imágenes).
- Sistema de clasificación de imágenes utilizando bolsa de palabras visuales.

A continuación se mostrarán los resultados obtenidos al implementar estos sistemas.

6.4.1 Sistema de clasificación de imágenes utilizando bolsa de características personalizada. Los sistemas CBIR (CBIR- Sistemas basados en contenido para la recuperación de imágenes) se utilizan para encontrar imágenes que son visualmente similares a la imagen de consulta. La aplicación de sistemas CBIR se puede encontrar en muchas áreas, tales como búsqueda de productos en la web, sistema de vigilancia, identificación visual de sitios. Una técnica común utilizada para implementar un sistema CBIR es la Bolsa de Palabras Visuales (BoVW), también llamada bolsa de características [1,2]. Bolsa de características es una técnica que utiliza características de la imagen como palabras visuales para describir la imagen.

Las características de la imagen son una parte importante de los sistemas CBIR. Estas características de la imagen se utilizan para medir la similitud entre las

imágenes y puede incluir características globales de la imagen, como color, textura y forma. Las características de la imagen se pueden extraer de alguna región de interés en la imagen. Para la extracción de dichas características se utilizan técnicas como (SURF - Speeded Up Robust Features), (HOG- Histograma de Gradiente Orientado) o (LBP- Patrones Binarios Locales). El beneficio del método de Bolsa de Características es que el tipo de características que se utilizan para crear el vocabulario visual de palabras puede ser personalizado para adaptarse a la aplicación.

La velocidad y la eficiencia de la búsqueda de imágenes es un factor importante en los sistemas CBIR. Por ejemplo, puede ser aceptable para realizar la búsqueda de una característica específica en una pequeña colección de imágenes (Menos de 100 imágenes), comparar las características de la imagen de consulta con cada una de las características de cada imagen en la colección. Para colecciones más grandes, una búsqueda de este tipo no es factible y técnicas de búsqueda más eficientes deben ser utilizadas.

La bolsa de características proporciona un esquema de codificación concisa para representar una gran colección de imágenes utilizando un conjunto disperso de histogramas de palabras visuales; esto permite un almacenamiento compacto y una búsqueda eficiente a través de la utilización de índices en la base de datos.

Los pasos que describen el procedimiento del sistema de recuperación de imágenes utilizando bolsa de características personalizada son los siguientes:

- Seleccionar las características que se van a recuperar de la imagen
- Crear una Bolsa de características
- Generar el índice de las Imágenes
- Buscar imágenes similares

6.4.1.1 Ejemplo 1. El siguiente ejemplo, muestra el procedimiento para crear un sistema de recuperación de imágenes que busca en una base de imágenes de carros, aquellos carros de características similares a la imagen de consulta [45].

Este conjunto de datos contiene cerca de 347 imágenes de diferentes tipos de carros.

❖ **Paso 1** - Seleccionar la imagen característica para recuperación:

El tipo de función utilizado para la recuperación depende del tipo de imágenes dentro de la colección. Si se busca una colección de imágenes de escenas donde se muestren detalles muy generales, por ejemplo encontrar playas, ciudades, carreteras, etc., es preferible utilizar características globales de la imagen, como histogramas de color los cuales capturan el contenido de color de toda la escena.

Si el objetivo es encontrar objetos específicos dentro de las colecciones de imágenes, es preferible utilizar características locales para extraer puntos significativos de interés alrededor de los objetos de la imagen.

Inicialmente se mostró un conjunto de imágenes al azar que muestra el tipo de imágenes de la colección. En este caso una colección de carros.

Figura 132. Imágenes al azar de carros de la colección de imágenes



Fuente: Modificado por Autor de [46]

En este ejemplo, el objetivo es buscar carros similares en el conjunto de imágenes utilizando la información de color de la imagen de consulta. Las imágenes en el conjunto de imágenes contienen un tipo de carro en cada imagen. Por lo tanto, una característica simple de la imagen puede ser la distribución espacial del color.

❖ **Paso 2.** Crear la bolsa de características:

Con el tipo de característica definida, en este caso la información de color (Distribución espacial de color), el siguiente paso es aprender el vocabulario visual utilizando un conjunto de imágenes de entrenamiento.

❖ **Paso 3.** Generar índices a las imágenes:

Ahora que se ha creado la bolsa de características, todo el conjunto de imágenes de carros puede ser indexado para la búsqueda. El procedimiento de indexación extrae características de cada imagen utilizando la función extractora personalizada desde el paso 1. Las características extraídas se codifican en un histograma de palabras visuales y se agrega en el índice de imágenes.

❖ **Paso 4.** Buscar imágenes similares:

El paso final es buscar imágenes similares. Se define una imagen de consulta, la cual se busca por el índice de ubicación en la base de la colección de imágenes.

En la Figura 133 se muestra la imagen de búsqueda que se quiere encontrar en la base de datos de la colección de imágenes de carros. Para este ejemplo, se escoge como imagen de consulta un carro de color rojo.

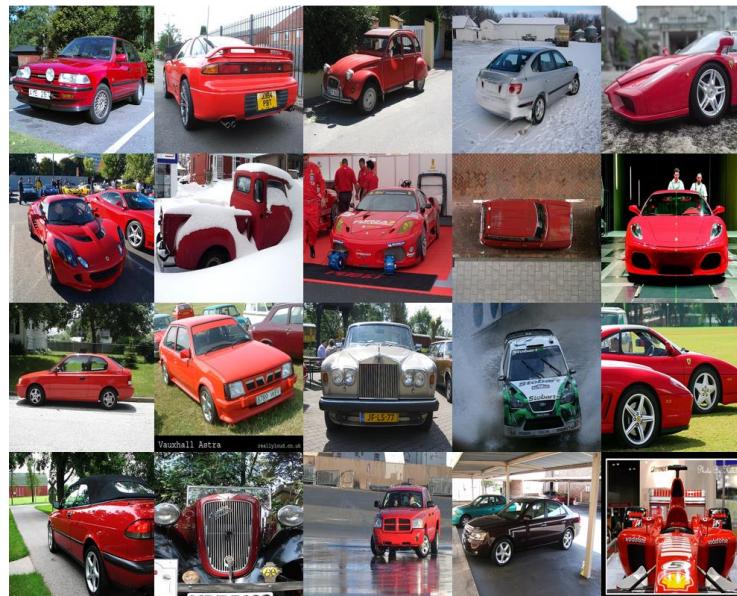
Figura 133. Imagen de consulta



Fuente: Modificado por Autor de [46]

En la Figura 134 se muestran las mejores 20 imágenes con contenido de color parecido. Como se observa, el sistema recupero de toda la base de datos de carros las 20 imágenes cuyo contenido por color se parecen más a la imagen de consulta.

Figura 134. Las 20 mejores imágenes de la colección con contenido de color parecido



Fuente: Modificado por Autor de [46]

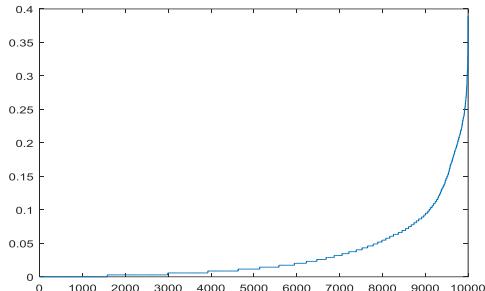
A continuación se define el rango de similitud de cada imagen. Este rango se define en orden de mejor a peor.

scores =

0.8847
0.2620
0.2174
0.2156
0.1985
0.1768
0.1757
0.1651
0.1623
0.1598
0.1538
0.1501
0.1470
0.1461
0.1460
0.1446
0.1404
0.1367
0.1356
0.1345

Se definen varios índices estadísticos que son relevantes para la búsqueda. Uno de ellos es el porcentaje de imágenes en las que se produce cada palabra visual. Esto muestra que palabras son más comunes, así como cuáles son raras en todo el conjunto de datos. Esta información a menudo es útil para suprimir las palabras más comunes con lo cual se reduce el conjunto de búsqueda. También es útil para suprimir aquellas palabras que pueden generar valores atípicos en el conjunto de la imagen. La Figura 135 muestra el gráfico de distribución de palabras visuales.

Figura 135. Gráfico de distribución de palabras visuales



Fuente: Modificado por Autor de [46]

El gráfico muestra que la distribución de las palabras visuales alcanza su pico en torno al 35%; esto significa que sólo unas pocas palabras visuales están en el 35% de las imágenes.

Para mostrar los efectos en los resultados de búsqueda, se puede bajar el rango superior al 20%, esto significa que sólo unas pocas palabras visuales están en el 20% de las imágenes. En la Figura 136 se observa el resultado.

Figura 136. Las veinte mejores imágenes de la colección con contenido de color parecido, con una distribución de palabras visuales para un valor igual al 20%



Fuente: Modificado por Autor de [46]

En este caso, ya que el rango superior es cerca de 35%, un ajuste de 20% limita que una gran cantidad de similitudes relevantes no hagan parte de los resultados de búsqueda. Esto confirma los pobres resultados de búsqueda.

6.4.1.2 Ejemplo 2. El siguiente es otro ejemplo utilizando la técnica de Bolsa de Palabras Personalizada. En este caso se utilizará una base de datos de flores ornamentales, y se buscará en dicha base, aquellas flores de características similares a la imagen de consulta [45]. Para este ejemplo se utilizó una base de datos conformada por 172 imágenes de diferentes tipos de flores ornamentales.

❖ **Paso 1** - Seleccionar la imagen característica para recuperación:

Inicialmente se mostró un conjunto de imágenes al azar que muestra el tipo de imágenes de la colección. En este caso una colección de flores ornamentales.

La Figura 137 muestra algunas de las 172 imágenes de la colección que serán objeto de la búsqueda.

Figura 137. Imágenes al azar de flores ornamentales de la colección de imágenes



Fuente: Autor

En este ejemplo, el objetivo es buscar flores ornamentales similares en el conjunto de imágenes utilizando la información de color de la imagen de consulta. Las imágenes en el conjunto de imágenes contienen un tipo de flor ornamental en cada imagen. Por lo tanto, una característica simple de la imagen puede ser la distribución espacial del color.

❖ **Paso 2.** Crear la bolsa de características:

Con el tipo de característica definida, en este caso la información de color (Distribución espacial de color), el siguiente paso es aprender el vocabulario visual utilizando un conjunto de imágenes de entrenamiento.

❖ **Paso 3.** Generar índices a las imágenes

Una vez creada la bolsa de características, todo el conjunto de imágenes de flores puede ser indexado para la búsqueda. El procedimiento de indexación extrae características de cada imagen utilizando la función extractora personalizada desde el paso 1. Las características extraídas se codifican en un histograma de palabras visuales y agregado en el índice de imágenes.

❖ **Paso 4.** Buscar imágenes similares:

El paso final es buscar imágenes similares. Se define una imagen de consulta, la cual se busca por el índice de ubicación en la base de la colección de imágenes.

En la Figura 138 se muestra la imagen de búsqueda que se quiere encontrar en la base de datos de la colección de imágenes de flores ornamentales. Para este ejemplo se escoge como imagen de consulta una flor exótica cuyo color predominante es el rojo.

Figura 138. Imagen de consulta



Fuente: Autor

En la Figura 139 se muestran las mejores 20 imágenes con contenido de color parecido. Como se observa el sistema recuperó de toda la base de datos de flores ornamentales las 20 imágenes cuyo contenido por color se parecen más a la imagen de consulta.

Figura 139. Las 20 mejores imágenes de la colección con contenido de color parecido



Fuente: Autor

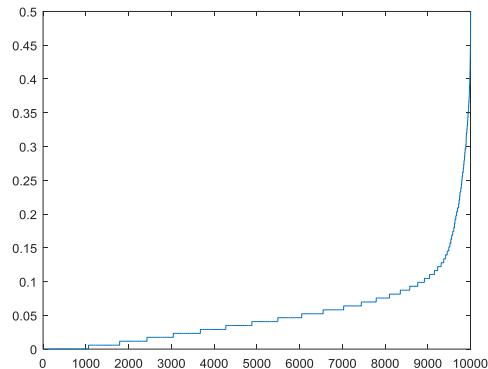
A continuación se define el rango de similitud de cada imagen. Este rango se define en orden de mejor a peor.

scores =

0.9983
0.4756
0.4305
0.4229
0.3985
0.3861
0.3774
0.3523
0.3469
0.3223
0.3163
0.3126
0.2829
0.2815
0.2727
0.2653
0.2609
0.2595
0.2529
0.2484

Se definen varios índices estadísticos que son relevantes para la búsqueda. Uno de ellos es el porcentaje de imágenes en las que se produce cada palabra visual. Esto muestra que palabras son más comunes, así como cuáles son raras en todo el conjunto de datos. Esta información a menudo es útil para suprimir las palabras más comunes con lo cual se reduce el conjunto de búsqueda. También es útil para suprimir aquellas palabras que pueden generar valores atípicos en el conjunto de la imagen. La Figura 140 muestra el gráfico de distribución de palabras visuales.

Figura 140. Gráfico de distribución de palabras visuales



Fuente: [45]

El gráfico muestra que la distribución de las palabras visuales alcanza su pico en torno al 35%. Esto significa que sólo unas pocas palabras visuales están en el 35% de las imágenes.

Al colocar el rango superior al 40% se muestran los efectos en los resultados de búsqueda. La siguiente figura muestra el resultado. En la Figura 141 se observa el resultado.

Figura 141. Las 20 mejores imágenes de la colección con contenido de color parecido con una distribución de palabras visuales para un valor igual al 40%



Fuente: Autor

Al colocar el rango superior al 5% se muestran los efectos en los resultados de búsqueda. La Figura 142 muestra el resultado.

Figura 142. Las 20 mejores imágenes de la colección con contenido de color parecido, con una distribución de palabras visuales para un valor igual al 5%



Fuente: Autor

En este caso, ya que el rango es cercano al 5%, este ajuste del 5% limita que una gran cantidad de similitudes relevantes no hagan parte de los resultados de búsqueda, con lo cual se confirman los pobres resultados de la búsqueda.

6.4.1.3 Conclusiones del sistema de clasificación y reconocimiento de imágenes. Estos ejemplos mostraron cómo personalizar las bolsas de características y cómo utilizar la indexación de las imágenes para crear un sistema de recuperación de imágenes basada en características de color.

6.4.2 Sistema de clasificación de imágenes utilizando Bolsa de Palabras Visuales (BoVW). El sistema de clasificación de imágenes utilizando Bolsa de Palabras Visuales (BoVW), crea un mapeo de palabras visuales y a través de un

índice define el número de apariciones de cada palabra visual en la colección de imágenes. A continuación se describe el procedimiento implementado.

6.4.2.1 Procedimiento del Sistema de recuperación de imágenes utilizando bolsa de palabras visuales. Dicho procedimiento consta de los siguientes pasos:

- ❖ **Crear un conjunto de imágenes que represente las características de la imagen a recuperar:**

En este paso se almacenan los datos de la imagen. Se utilizan un gran número de imágenes que representan varios puntos de vista del objeto a recuperar en la imagen. Un gran número de imágenes de diversa forma, ayuda a entrenar la bolsa de palabras visuales y aumenta la precisión de búsqueda del objeto a recuperar en la imagen.

- ❖ **Generar tipo de características:**

Se crea la bolsa de palabras visuales utilizando las características del descriptor, este puede ser cualquiera de los descriptores estudiados anteriormente, SURF, SIFT, etc. También se puede utilizar un descriptor particular de características para otros tipos de características y luego crear la bolsa de palabras visuales.

Para obtener un buen conjunto de entrenamiento, se puede utilizar una colección de imágenes de diferentes tipos. Luego se crea la bolsa de palabras visuales antes de crear el índice de imágenes. La ventaja de utilizar el mismo conjunto de imágenes es que el vocabulario visual se adapta rápidamente al conjunto de búsqueda. La desventaja de este enfoque es que el sistema de recuperación debe volver a aprender el vocabulario visual para utilizar en un conjunto de imágenes drásticamente diferentes.

- ❖ **Generar índices a las imágenes:**

Se crea un índice de búsqueda que se asigna a las palabras visuales por su aparición en la colección de imágenes.

Los datos de búsqueda establecidos para las imágenes similares, buscan un conjunto de imágenes que son similares a la imagen de consulta. Por ejemplo, para devolver las mejores imágenes similares, se utiliza una región más pequeña de la imagen de consulta. Una región más pequeña es útil para aislar un objeto en particular, en una imagen que se desea buscar.

❖ **Evaluar las imágenes recuperadas:**

Para evaluar la recuperación de imágenes mediante el uso de una imagen de consulta, se utiliza un conjunto conocido de resultados. Si los resultados no son lo que se esperan, se puede modificar o aumentar las características de la imagen en la bolsa de palabras visuales y examinar nuevamente el tipo de características recuperado. El tipo de función utilizada para la recuperación de imágenes depende del tipo de imágenes dentro de la colección.

6.4.2.2 Sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales, máquinas de vector de soporte (SVM) y descriptores SURF y SIFT. A continuación se describe el sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales (BoVW), máquinas de vector de soporte (SVM) y descriptores SURF y SIFT.

Inicialmente se hará la descripción detallada del sistema utilizando máquinas de vector de soporte (SVM) y el descriptor SURF.

❖ **Sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales (BoVW), máquinas de vector de soporte (SVM) y descriptores SURF:**

La clasificación de imágenes utilizando bolsa de palabras visuales se hace mediante la creación de una bolsa de palabras visuales. El proceso genera un histograma de apariciones de palabras visuales para representar una imagen. Estos histogramas se utilizan para entrenar un sistema clasificador compuesto por máquinas de vector de soporte que define las imágenes por categorías.

Los siguientes pasos describen cómo configurar las imágenes, crear la bolsa de palabras visuales, entrenar y clasificar en el sistema clasificador diferentes categorías de imágenes.

❖ **Paso 1.** Configurar la imagen según su categoría

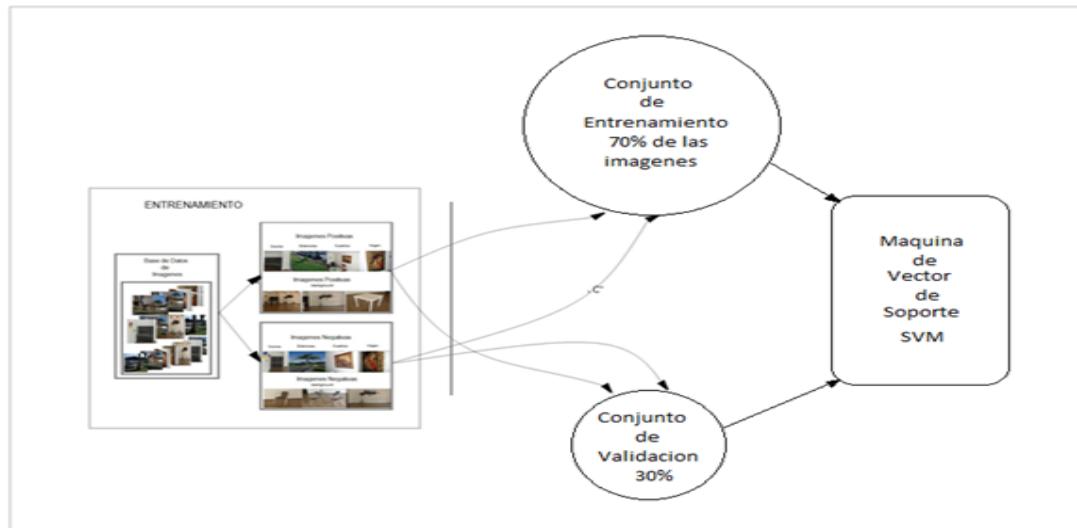
Se organizan las imágenes a entrenar por categorías para ingresarlas al sistema clasificador de imágenes. La organización de imágenes en categorías hace mucho más fácil el manejo de grandes conjuntos de imágenes.

Lee las imágenes por categorías y crea los conjuntos de imágenes. Separa los conjuntos de imágenes de cada categoría en subconjuntos de imágenes de entrenamiento y prueba.

En este ejemplo, el 70% de las imágenes se utilizó para entrenamiento y el resto para la prueba.

En la Figura 143 se muestra el sistema clasificador de imágenes.

Figura 143. Sistema clasificador de imágenes



Fuente: Autor

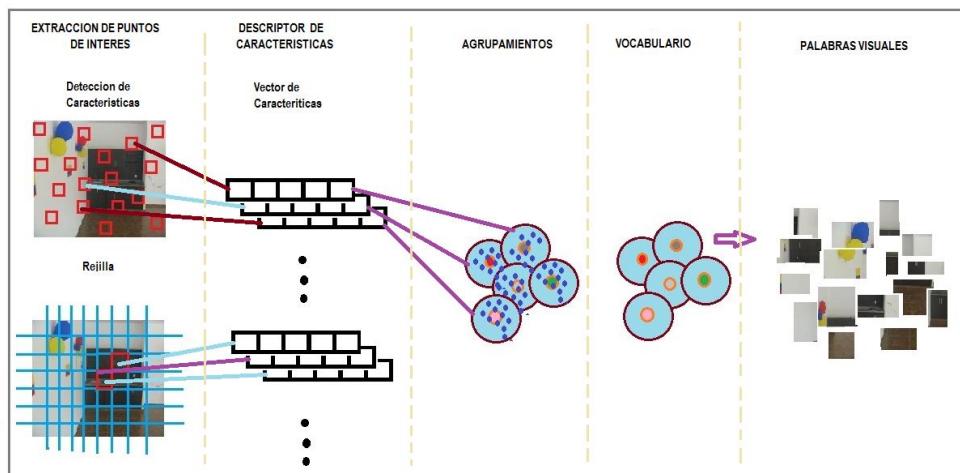
❖ **Paso 2.** Crear la bolsa de características:

Se crea un vocabulario visual o una bolsa de características mediante la extracción de descriptores de características representativas de cada una de las categorías de las imágenes.

Se definen las características o palabras visuales, mediante el uso de algoritmos de agrupamiento k-means en los descriptores de las características extraídas. El algoritmo iterativamente agrupa los descriptores en k grupos mutuamente excluyentes. Las agrupaciones resultantes son compactas y separadas por características similares. Cada centro de clúster representa una característica o una palabra visual.

Se puede extraer características, basados en un detector de características y definir una rejilla para extraer los descriptores de características. Utilizando un descriptor de características como el SURF, se puede lograr una mayor invarianza de escala. Para este caso, el algoritmo se ejecuta utilizando el método de rejilla para definir los descriptores de características, realizar los agrupamientos y generar los vocabularios de las palabras visuales. En la Figura 144 se muestra el procedimiento empleado para la generación de las palabras visuales.

Figura 144. Procedimiento de generación de palabras visuales



Fuente: Autor

El algoritmo analiza las imágenes en su totalidad. Las imágenes deben tener las etiquetas apropiadas que describen la clase que representan. Por ejemplo, un conjunto de imágenes de paisajes pueden ser etiquetadas como paisajes. El algoritmo no se basa en información espacial ni marca objetos particulares de una imagen. La técnica de bolsa de palabras visuales se basa en una detección sin localización.

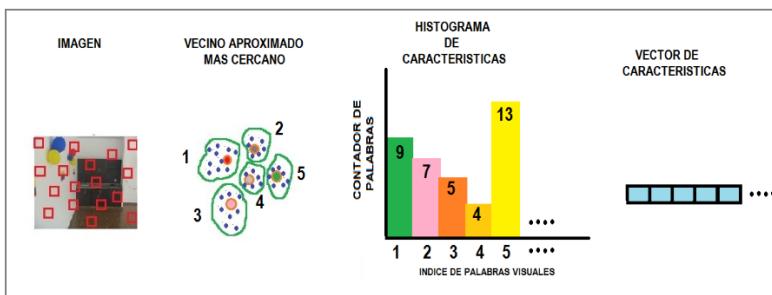
❖ **Paso 3.** Entrenar el sistema clasificador de imágenes con la bolsa de palabras visuales:

Se entrena un sistema clasificador multiclase utilizando una máquina de vector de soporte (SVM), utilizando la bolsa de palabras visuales de los objetos para codificar imágenes del conjunto de imágenes por medio de histogramas de palabras visuales.

El histograma de palabras visuales se utiliza luego como muestras positivas y negativas para entrenar el sistema clasificador.

1. Se utiliza un método de codificación para codificar cada imagen del conjunto de entrenamiento. Se detectan y extraen características de la imagen y luego se utiliza el algoritmo del vecino próximo más cercano para construir un histograma de características para cada imagen. El ancho del histograma se define por la proximidad del descriptor a un centro de clúster particular. La longitud del histograma corresponde al número de palabras visuales con que cada objeto es construido. El histograma se convierte en un vector de características para la imagen. La Figura 145 muestra la generación de vectores característicos para cada palabra visual.

Figura 145. Generación de vectores característicos para cada palabra visual.

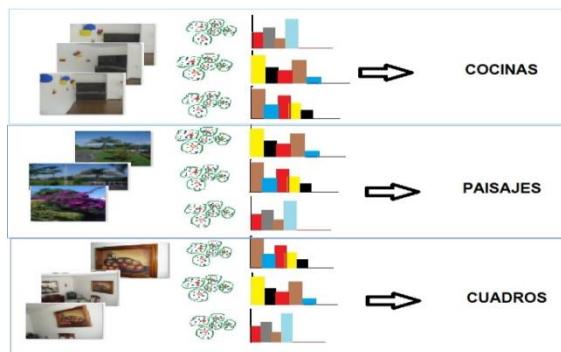


Fuente: Autor

2. Repita el paso 1 para cada imagen en el conjunto de entrenamiento para crear los datos de entrenamiento.

La Figura 146 muestra diferentes tipos de imágenes y los respectivos histogramas para cada clase.

Figura 146. Diferentes tipos de imágenes a clasificar



Fuente: Autor

3. Se evalúa la calidad del clasificador utilizando el método de matriz de confusión. Este método sirve para probar el conjunto de imágenes clasificadas contra el conjunto de imágenes validadas. Una matriz de confusión de salida representa el análisis de la predicción. Una clasificación perfecta se obtiene si en la matriz normalizada se obtienen 1s en la diagonal. Una clasificación errónea se obtiene con resultados de valores fraccionarios.

❖ **Paso 4.** Clasificar una imagen o un conjunto de imágenes:

Se utiliza de nuevo el método para determinarle a una nueva imagen su categoría.

Ejemplo 1. El siguiente ejemplo muestra la aplicación del método para tres categorías de imágenes. En este ejemplo se tomaron 254 imágenes de una cocina vista desde diferentes puntos de vista, 189 imágenes de quioscos vistas desde diferentes puntos de vista y 99 imágenes de un cuadro tomadas desde diferentes puntos de vista.

Inicialmente se iguala el número de imágenes por cada categoría para mejorar las condiciones iniciales del sistema clasificador. En este caso se igualan todas las clases al menor número de imágenes de una de las categorías, en este caso a la clase cuadro, que es la que tiene el menor número de imágenes, pues sólo tiene 99.

Cocinas = 'Cocinas'. 99 imágenes de cocinas.

Kioskos = 'Kiosko'. 99 imágenes de Kioskos.

Paisajes = 'PaisajesSURF'. 99 imágenes de Paisajes.

ans =

99 99 99

Creación de la Bolsa de palabras visuales para los tres conjuntos de imágenes.

- * La categoría 1: Imágenes de Cocinas.
- * La categoría 2: Imágenes de Kioskos.
- * La categoría 3: Imágenes de Paisajes (PaisajesSURF).

* Se hace la extracción de características con el descriptor SURF, utilizando el método de selección de rejilla.

Para el entrenamiento de las respectivas máquinas de soporte vectorial se utilizó el método de validación cruzada, Holdout method (Método de Retención).

Para el ejemplo se utilizaron 30 imágenes de cada clase para entrenar las respectivas máquinas de soporte y 69 imágenes de cada clase para realizar la validación de cada clase en la respectiva máquina de soporte.

Para la fase de entrenamiento se trajeron las características a treinta imágenes de la categoría 1, obteniéndose 453,600 características, a las treinta imágenes de la categoría 2, obteniéndose 627,480 características y a las treinta imágenes de la categoría 3, obteniéndose 745,680 características.

Se obtiene el 80% de las características más significativas de cada conjunto de imágenes, resultando que el conjunto de imágenes de la categoría 1 tiene el menor

número de características más significativas: 362.880; entonces se fuerza para que las otras categorías de imágenes tengan el mismo número de características significantes. El procedimiento de igualar el número de características en todas las categorías de las imágenes mejora el proceso de agrupación. Una vez igualado el número de características en todas las categorías de las imágenes, se procede a realizar el proceso de agrupación, para lo cual se utiliza la agrupación K-mean para crear un vocabulario de 500 palabras visuales.

Número de características: 1088640

Número de agrupaciones (K): 500

Una vez terminado de crear la Bolsa de palabras visuales, se procede al entrenamiento del sistema clasificador para las 3 categorías de imágenes.

- Categoría 1: Cocinas.
- Categoría 2: Kiosko.
- Categoría 3: PaisajesSURF.

Se realiza la codificación de todas las características, de todas las categorías de imágenes.

Codificación de características para la categoría 1.

Codificación de características para la categoría 2.

Codificación de características para la categoría 3.

Terminado el entrenamiento del sistema clasificador de categorías, se utiliza un sistema que pueda evaluar el sistema clasificador para los conjuntos de entrenamiento de cada categoría.

Evaluación del sistema clasificador para las 3 categorías de imágenes de entrenamiento.

Se evaluó el sistema clasificador para las tres categorías de imágenes, utilizando las imágenes de entrenamiento.

Se evaluaron 30 imágenes de entrenamiento de la categoría 1.

Se evaluaron 30 imágenes de entrenamiento de la categoría 2.

Se evaluaron 30 imágenes de entrenamiento de la categoría 3.

Terminada la evaluación de todos los conjuntos de entrenamiento. La matriz de confusión para este conjunto de entrenamiento fue:

PRONOSTICADO

CONOCIDO	Cocinas	Kiosko	PaisajesSURF
Cocinas	1.0	0.0	0.0
Kiosko	0.0	1.0	0.0
PaisajesSURF	0.0	0.03	0.97

Promedio de Precisión es 0,99.

Lo que se deduce del resultado de la matriz de confusión es que para las treinta imágenes de cocinas, la máquina de soporte de cocinas clasificó todas las treinta imágenes de cocinas.

Para las treinta imágenes de quioscos, la máquina de soporte de quioscos clasificó todas las treinta imágenes de quioscos.

Para las treinta imágenes de Paisajes (PaisajesSURF), la máquina de soporte de Paisajes clasificó veinte y nueve imágenes como imágenes de paisajes y una imagen de quiosco la clasificó como paisaje.

Evaluación del sistema clasificador para las 3 categorías de imágenes de prueba.

Se evaluó el sistema clasificador para las tres categorías de imágenes, utilizando las imágenes de prueba de las tres categorías.

Categoría 1: Cocinas

Categoría 2: Kiosko

Categoría 3: PaisajesSURF

Se evaluaron 69 imágenes de prueba de la categoría 1.

Se evaluaron 69 imágenes de prueba de la categoría 2.

Se evaluaron 69 imágenes de prueba de la categoría 3.

Terminada la evaluación de todos los conjuntos de prueba. La matriz de confusión para este conjunto de prueba fue:

CONOCIDO	Cocinas	Kiosko	PaisajesSURF
Cocinas	1.0	0.0	0.0
Kiosko	0.0	1.0	0.0
PaisajesSURF	0.0	0.03	0.97

Promedio de Precisión es 0,99.

Lo que se deduce del resultado de la matriz de confusión es que para las sesenta y nueve imágenes de cocinas, la máquina de soporte de cocinas clasificó todas las sesenta y nueve imágenes de cocinas.

Para las sesenta y nueve imágenes de quioscos, la máquina de soporte de quioscos clasificó todas las sesenta y nueve imágenes de quioscos.

Para las sesenta y nueve imágenes de Paisajes (PaisajesSURF), la máquina de soporte de Paisajes clasificó sesenta y nueve imágenes como imágenes de paisajes y una imagen de quiosco la clasificó como paisaje.

La Figura 147 muestra una imagen de cada categoría después de la clasificación realizada por las respectivas máquinas de vector de soporte. Como se observa en la Figura se clasifican de manera eficiente las diferentes categorías de imágenes.

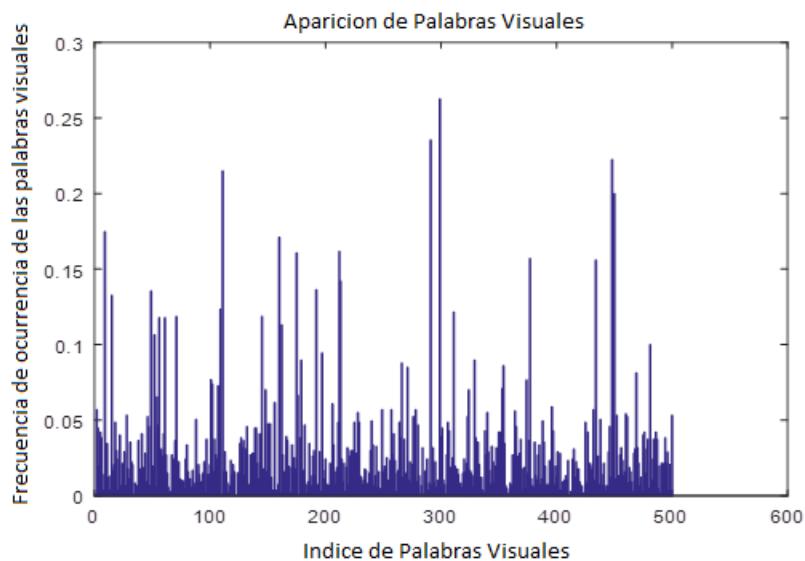
Figura 147. Tipo de imágenes clasificadas



Fuente: Autor

La Figura 148 muestra el índice de aparición de palabras visuales en una de las categorías de imágenes clasificadas.

Figura 148. Índice de aparición de palabras visuales



Fuente: Modificado por Autor [46]

❖ **Sistema de clasificación de imágenes y reconocimiento de imágenes utilizando bolsa de palabras visuales y descriptores SIFT:**

Se describe el método propuesto de clasificación y reconocimiento de imágenes, utilizando la técnica de bolsa de palabras visuales (BoVW), máquinas de vector de soporte (SVM) y descriptores SIFT a través de tareas de reconocimiento de objetos en una imagen dada usando sus vectores de características. La estructura del vector de característica consta del cálculo de características SIFT sobre una rejilla regular a través de la imagen ('densa SIFT') y la cuantificación del vector en palabras visuales. El clasificador es una máquina lineal de vector de soporte (SVM). En clasificación de imágenes, una imagen es clasificada de acuerdo con su contenido visual. Una aplicación importante es el reconocimiento de imágenes, es buscar a través de una base de datos de imágenes para obtener (o reconocer) aquellas imágenes con un contenido visual particular.

Es importante resaltar que en este sistema se utilizó la clasificación de una categoría de imágenes utilizando la estrategia uno contra todos, es decir, para validar la efectividad del sistema de clasificación se validó la máquina de soporte

con datos de otras categorías, es decir, se entrenó con una categoría y se validó con otra cuya información correspondía a otras categorías.

Los pasos a seguir para la ejecución de este procedimiento, son los siguientes:

❖ **Realizar pruebas de clasificación de imágenes:**

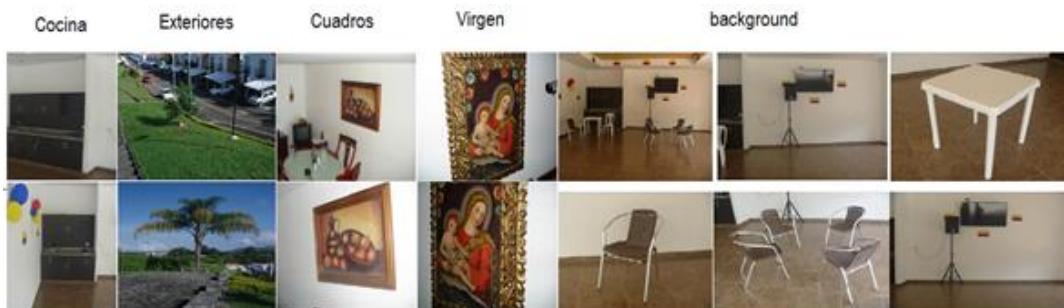
Las pruebas para hacer la clasificación y el reconocimiento parten de un conjunto de imágenes tomadas de diferentes entornos, las cuales serán procesadas para su posterior clasificación. El clasificador utiliza para su entrenamiento de validación el método de Retención (holdout method).

Estas pruebas incluyen conjuntos de imágenes de diversas categorías. Algunas de estas imágenes son: Imágenes de mesas, salas, sillas, cocinas, paisajes, cuadros etc. En estas pruebas se utilizan sólo las imágenes de nuestra propia base de datos.

A continuación se muestran los resultados obtenidos de algunas de las pruebas realizadas con la base de datos de imágenes tomadas a paisajes, cocinas, salas y cuadros.

En la Figura 149 se muestran algunas de las categorías de imágenes utilizadas.

Figura 149. Conjunto de imágenes de diferente tipo



Fuente: Autor

❖ **Ejemplo 1.** En este ejemplo se utilizan las imágenes de la base de datos del autor. Para mostrar la efectividad del sistema de clasificación y reconocimiento de imágenes se toman como imágenes a reconocer imágenes de la clase cocina,

donde se entrena la máquina de soporte con imágenes de cocina e imágenes cuadros y exteriores (paisajes) utilizando el método de validación de retención. Estrategia Uno contra Todos.

Esta prueba incluye 2 conjuntos de imágenes de 2 clases. La primera clase incluye imágenes de fotos de cocina, llamada cocinaUNO y la segunda clase se denomina imágenes de backCuadrosExter, las cuales son imágenes de Cuadros y Exteriores solamente.

La Figura 150 muestra algunas imágenes del conjunto de imágenes cocinaUNO.

Figura 150. Imágenes del conjunto de imágenes cocinaUNO



Fuente: Autor

El conjunto de imágenes de cocinaUNO, se divide a su vez en 2 subconjuntos, el primero se denomina cocinaUNO_train, el cual contiene un total de 63 imágenes, para entrenar el sistema y el conjunto cocinaUNO_val el cual contiene un total de 82 imágenes para validar el sistema. Las imágenes de cocinaUNO son imágenes de una misma cocina, tomada desde diferentes puntos de vista, es decir tanto las imágenes de entrenamiento, como las imágenes de validación son de semántica similar.

El conjunto de imágenes de backCuadrosExter, se divide a su vez en 2 subconjuntos el primero se denomina backCuadrosExter _train_hist.mat, el cual contiene un total de 262 imágenes y el conjunto backCuadrosExter _val_hist.mat el cual contiene un total de 261 imágenes.

Es importante resaltar que las imágenes contenidas en backCuadrosExter, no contienen ninguna imagen de las contenidas en los archivos de cocinaUNO y su composición semántica es muy diferente.

Número de imágenes de entrenamiento: 63 positivas, 262 negativas

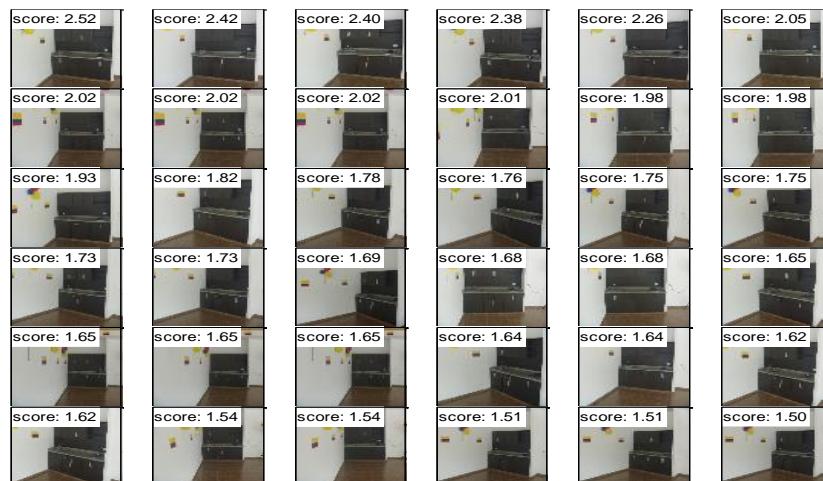
Número de imágenes de prueba: 82 positivas, 261 negativas

Índice de Prueba: 0.99

Imágenes correctamente reconocidas de 36, reconoció correctamente 35.

La Figura 151 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría.

Figura 151.Ranking de Imágenes de entrenamiento (subset)



Fuente: Autor

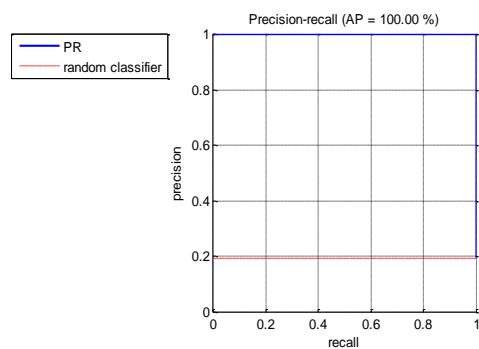
La Figura 152 mide el rendimiento del clasificador en la recuperación cuantitativa a través del cálculo de la curva de Precision-Recall. Donde Precisión define la

proporción de imágenes retornadas que son positivas y Recall define la proporción de imágenes positivas que son retornadas.

Tanto el ranking como el cálculo de la curva de Precision-Recall se hallan tanto para las imágenes de entrenamiento como para las imágenes de prueba o validación.

La Figura 152 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de entrenamiento.

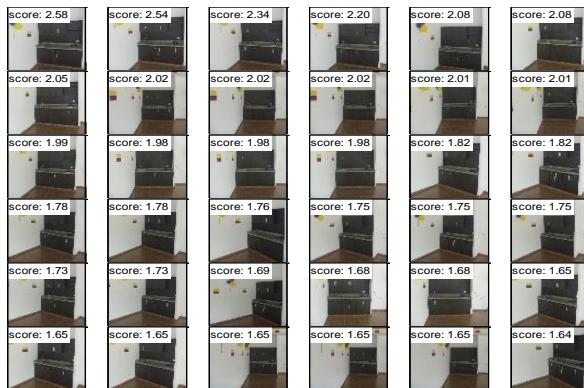
Figura 152. Precision-Recall en datos de entrenamiento



Fuente: Autor

La Figura 153 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría.

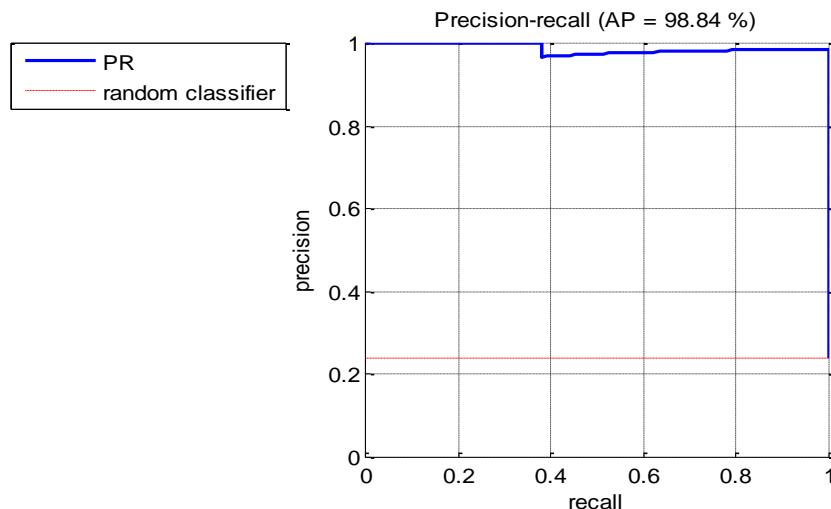
Figura 153. Ranking de imágenes de pruebas (subset)



Fuente: Autor

La Figura 154 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de prueba.

Figura 154. Precision-Recall en datos de prueba



Fuente: Autor

❖ **Ejemplo 2.** Esta prueba incluye 2 conjuntos de imágenes de 2 clases. La primera clase incluye imágenes de fotos de exteriores y la segunda clase se denomina imágenes de backCociCuadros.

El conjunto de imágenes de Exteriores, se divide a su vez en 2 subconjuntos, el primero se denomina ExterioresUNO_train_hist, el cual contiene un total de 36 imágenes, para entrenar el sistema y el conjunto ExterioresUNO_val_hist el cual contiene un total de 116 imágenes para validar el sistema.

El conjunto de imágenes de backCociCuadros, se divide a su vez en 2 subconjuntos el primero se denomina backCociCuadros _train_hist, el cual contiene un total de 78 imágenes y el conjunto backCociCuadros _val_hist el cual contiene un total de 336 imágenes.

Es importante resaltar que las imágenes contenidas en backCociCuadros, no contienen ninguna imagen de las contenidas en los archivos de ExterioresUNO **y su composición semántica es completamente diferente. Ademas al obtener el vector de características sólo se tomaron en cuenta los archivos que contenían las imágenes de Exteriores y las imágenes de backCociCuadros, es decir sólo se procesaron exteriores y en el backCociCuadros imágenes de cocinas y cuadros.**

Número de imágenes de entrenamiento: 36 positivas, 78 negativas

Número de imágenes de prueba: 116 positivas, 336 negativas

Índice de Prueba: 0.99

Imágenes correctamente reconocidas de 36, reconoció correctamente 36.

La Figura 155 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría.

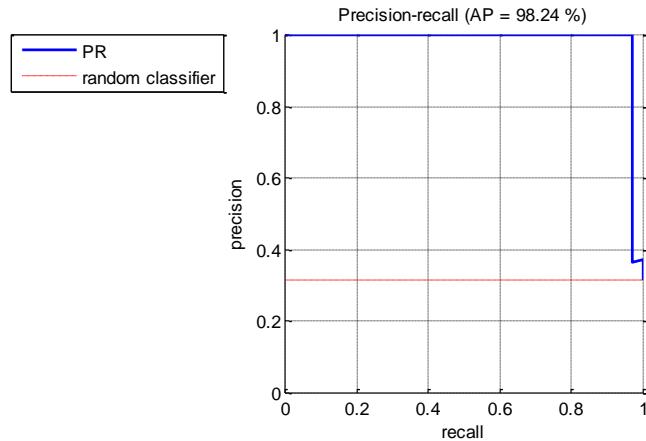
Figura 155. Ranking de imágenes de entrenamiento (subset)



Fuente: Autor

La Figura 156 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de entrenamiento.

Figura 156. Precision-Recall en datos de entrenamiento



Fuente: Autor

La Figura 157 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría para los datos de prueba.

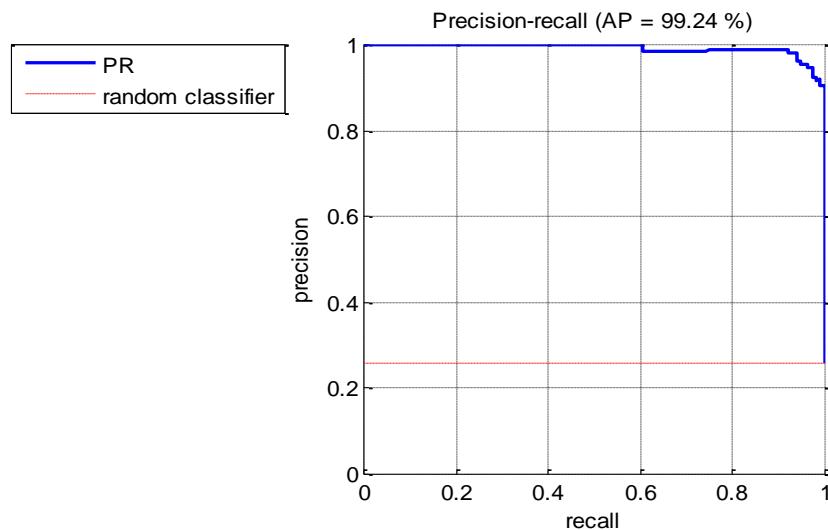
Figura 157. Ranking en imágenes de prueba (subset)



Fuente: Autor

La Figura 158 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de prueba.

Figura 158. Precision-Recall en datos de prueba



Fuente: Autor

- ❖ **Ejemplo 3.** Esta prueba incluye 2 conjuntos de imágenes de 2 clases. La primera clase incluye imágenes de fotos de cuadros, llamada cuadrosUNO y la segunda clase se denomina imágenes de backCociExter. Es importante resaltar que en las imágenes de backCociExter, se utilizaron sólo las imágenes de nuestra propia base de datos.

El conjunto de imágenes de backCociExter, se divide a su vez en 2 subconjuntos el primero se denomina backCociExter _train_hist.mat, el cual contiene un total de 250 imágenes y el conjunto backCociExter _val_hist.mat el cual contiene un total de 269 imágenes.

Número de imágenes de entrenamiento: 63 positivas, 250 negativas

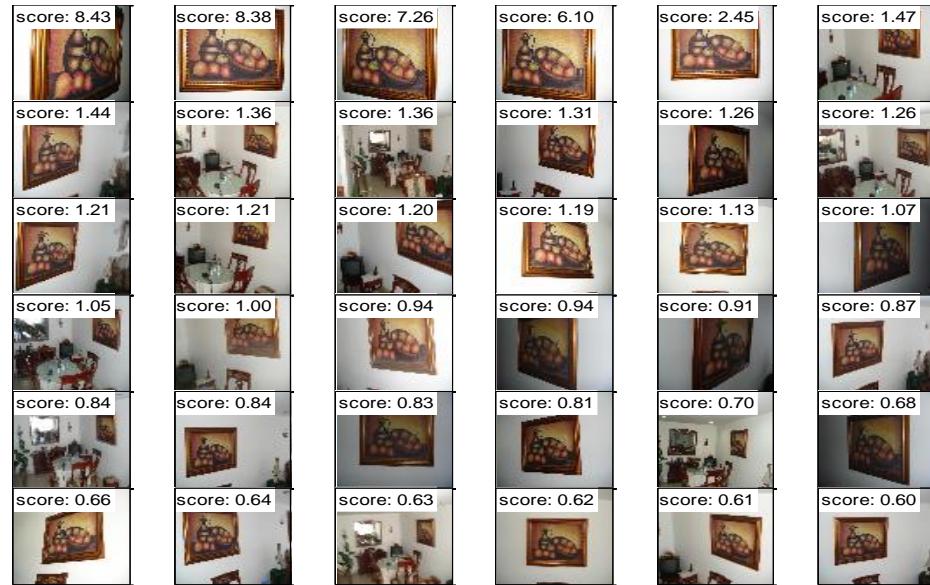
Número de imágenes de prueba: 84 positivas, 269 negativas

Índice de Prueba: 0.97

Imágenes correctamente reconocidas de 36, reconoció correctamente 36.

La Figura 159 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría.

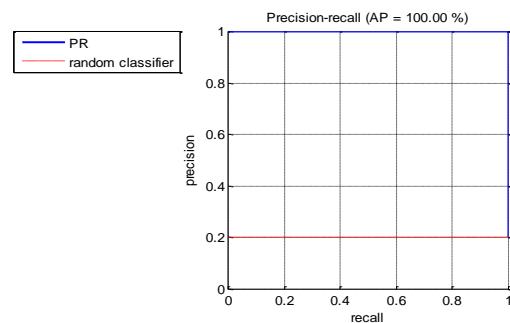
Figura 159. Ranking de imágenes de entrenamiento (subset)



Fuente: Autor

La Figura 160 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de entrenamiento.

Figura 160. Precision-Recall en datos de entrenamiento



Fuente: Autor

La Figura 161 muestra el ranking que clasifica las imágenes cuyas palabras visuales, el sistema clasificador considera tienen mayor relación con la categoría para los datos de prueba.

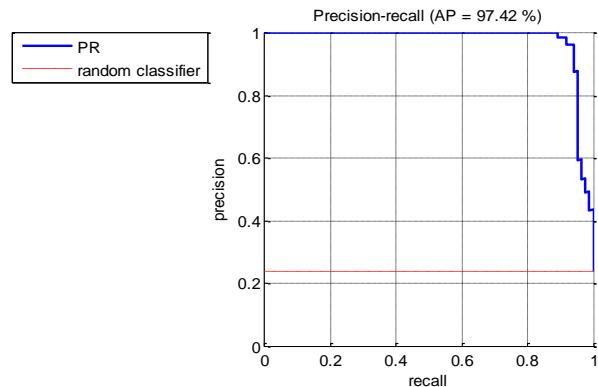
Figura 161. Ranking en imágenes de prueba (subset)



Fuente: Autor

La Figura 162 muestra la gráfica del cálculo de la curva de Precision-Recall para los datos de prueba.

Figura 162. Precision-Recall en datos de prueba



Fuente: Autor

7. CONCLUSIONES

El sistema de clasificación y reconocimiento de imágenes implementado, parte de la extracción de puntos de interés de las imágenes de la base de datos utilizando los descriptores SIFT y SURF y la respectiva comparación de estos puntos en las imágenes de estudio para determinar aquellas zonas cuyos puntos de interés son similares. Los resultados mostraron como el esquema pudo distinguir diferentes objetos de las imágenes incluso cuando en las imágenes dichos objetos tenían variación en escala, iluminación y perspectiva.

Al implementar el esquema de búsqueda de objetos específicos de una imagen de consulta en un conjunto de imágenes, el esquema reconoció con un alto grado de acierto aquellas imágenes donde los objetos específicos se encontraban.

De los esquemas implementados se concluye que ellos por sí mismos no son suficientes para realizar tareas de clasificación y reconocimiento de imágenes, sino que se hace necesario implementar técnicas más sofisticadas que incluyan mecanismos de aprendizaje, lo cual se logró con la implementación del sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales, máquina de vector de soporte y descriptores.

De los diferentes tipos de sistemas de clasificación y reconocimiento de imágenes utilizando la bolsa de palabras visuales implementados, se concluye que dependiendo del tipo de imágenes a clasificar y reconocer es el sistema a utilizar. Por ejemplo, si las imágenes a clasificar tienen en común características de color, forma o dimensión, el sistema de clasificación puede utilizar una bolsa de palabras visuales personalizada. En este proyecto, se implementó un sistema cuya característica personalizada es el color y se obtuvieron muy buenos resultados.

En los sistemas de clasificación y reconocimiento de imágenes utilizando la bolsa de palabras visuales, máquinas de vector de soporte y descriptores, se trabajaron dos sistemas, el primero utilizando descriptores SURF y varias máquinas de vector de soporte por cada categoría de imágenes concluyendo que dicho sistema presentó muy buenos resultados de clasificación para diferentes categorías de imágenes. En las pruebas realizadas se tomaron varios casos de estudio y el sistema respondió muy bien. Para el segundo caso, utilizando la bolsa de palabras visuales y descriptores SIFT, se utilizó la máquina de vector de soporte en el modo de clasificación uno contra todos y se obtuvieron resultados de clasificación y

reconocimiento de imágenes con alto grado de acierto, mostrando la robustez del sistema.

En general se concluye que los resultados del sistema de clasificación y reconocimiento utilizando la bolsa de palabras visuales, máquinas de soporte y descriptores funcionó muy bien para las tareas de clasificación y reconocimiento de imágenes.

8. RECOMENDACIONES

Una vez concluido el trabajo de la implementación del sistema de clasificación y reconocimiento de imágenes utilizando bolsa de palabras visuales, máquinas de vector de soporte y descriptores, el siguiente trabajo estaría orientado a:

- ❖ Implementar los diferentes sistemas de clasificación y reconocimiento en módulos embebidos, como Beagle Bone o tarjetas multi-procesadores para trabajo en paralelo.
- ❖ Trabajar dichos sistemas en un lenguaje de programación de software libre especializado en procesamiento de imágenes, como el openCV bajo el ambiente de Ubuntu.
- ❖ Estudiar e implementar un descriptor de puntos de interés que incluya las técnicas modernas de Big Data.

9. BIBLIOGRAFÍA

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60(2), pp. 91-110, 2004.
- [2] Stephens, Chris Harris & Mike. *A combined corner and edge detector*. s.l. : In Fourth Alvey Vision Conference, 1988.
- [3] Brady, S.M. Smith and J.M. *SUSAN- a new approach to low level image processing*. s.l. : International Journal of Computer Vision, (May 1997).
- [4] H. Zhang, A. Berg, M. Maire and J. Malik. *Svm-knn: discriminative nearest neighbor.classification for visual category recognition*. s.l. : CVPR, 2006.
- [5] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Y. Wu. *An efficient kmeans clustering algorithm: Analysis and implementation*. s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 2002.
- [6] Reynolds, Douglas. *Gaussian Mixture Models*. USA : MIT Lincoln Labaratory.
- [7] Triggs, F. Jurie and B. *Creating efficient codebooks for visual recognition*. s.l. : In ICCV, 2005.
- [8] A. Bosch, A. Zisserman and X. Munoz. *Image classification using random forests and ferns*. s.l. : ICCV, 2007.
- [9] A. Kumar, C. Sminchisescu. *Support kernel Machines for object recognition*. s.l. : In ICCV, 2007.
- [10] X. Munoz and A. Bosch, A. Zisserman. *Scene Classification via pLSA*. s.l. : In Proc. ECCV, 2006.
- [11] F. Monay, P. Quelhas, D. Gatica-Perez and J.-M. Odobez. *Constructing Visual Models with a Latent Space Approach*. s.l. : IDIAP , 2005.
- [12] G. Martínez-Muñoz, W. Zhang, N. Payet, S. Todorovic, N. Larios, A. Yamamuro, D. Lytle, A. Moldenke, E. Mortensen, R. Paasch, L. Shapiro and T. G. Dietterich.

Dictionary-Free Categorization of Very Similar Objects via Stacked Evidence Trees.
s.l. : In: CVPR, 2008.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf:Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.

[14] Philbin, J., Chum, O., Isard, M., A., J.S., Zisserman: Object retrieval with large vocabularies and fast spatial matching. In: CVPR. (2007)

[15] M. Turk and A. Pentland "Face recognition using eigenfaces". *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1991. pp. 586–591.

[16] Lowe, D. G., "Object recognition from local scale-invariant features", International Conference on Computer Vision, Corfu, Greece, September 1999.

[17] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, June 2006, vol. II, pp. 2169-2178.

[18] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. In Proc. ICCV, 2005.

[19] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric l_p -norm feature pooling for image classification, in: Conference on Computer Vision and Pattern Recognition, 2011, pp.2609–2704.

[20] Fernanda B. Silva, Siome Goldenstein, Salvatore Tabbone, and Ricardo da S. Torres. "Image Classification based on Bag of Visual Graphs". RECOD Lab, Institute of Computing, University of Campinas – UNICAMP 13083-852, Campinas, SP – Brazil.

[21] Svebor Karaman, Jenny Benois-Pineau, Rémi Mégret and Aurélie Bugeau. "Multi-Layer Local Graph Words for Object Recognition" LaBRI - University of Bordeaux, 351, Cours de la Libération, 33405 Talence Cedex, , IMS - University of Bordeaux, 351, Cours de la Libération 33405 Talence Cedex, France.

[22] H. Bay, A. Ess, T. Tuytelaars and Luc Van Gool. *Speeded-Up Robust Features (SURF)*. Zurich : in Computer Vision - ECCV , 2006.

- [23] A. Andreopoulos, J.K. Tsotsos, 50 years of object recognition: directions forward, *Computer Vision and Image Understanding* 117(8) (2013)827–891.
- [24] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, “Visual word ambiguity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271 –1283, 2010.
- [25] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Surf:Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.
- [26] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” *ICCV’2003*, vol. 2, pp. 1470-1477, 2003.
- [27] D. G. Lowe, Distinctive image features from scale-invariant key points, *International Journal of Computer Vision*60 (2) (2004) 91–110.
- [28] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial-bag-of-features, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 3352– 3359.
- [29] W. Zhou, Y. Lu, H. Li, Y. Song, Q. Tian, Spatial coding for large scale partial-duplicate web image search, in: International Conference on Multimedia, 2010,pp.511–520.
- [30] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large-scale image search, *International Journal of Computer Vision* 87 (2010) 316–336.
- [31] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric L_p -norm feature pooling for image classification, in: Conference on Computer Vision and Pattern Recognition, 2011, pp.2609–2704.
- [32] <https://es.wikipedia.org/Wikipedia>, la enciclopedia libre.
- [33] J.C. raina, A. Traina, C. Faloutsos, B. Seeger, Fast indexing and visualization of metric data sets using slim-trees, *Transactions on Knowledge and Data Engineering* 14(2)(2002)244–260.
- [34] O.A.B. Penatti, E. alle,R.d.S.Torres, Encoding spatial arrangement of visual words, in: Iberoamerican Congress on Pattern Recognition, vol.7042, 2011, pp. 240–247.

- [35] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large-scale image search, International Journal of Computer Vision 87 (2010) 316–336.
- [36] N.V. Hoàng, V. Gouet-Brunet, M. Rukoz, M. Manouvrier, Embedding spatial information into image content description for scene retrieval, Pattern Recognition 43(2010)3013–3024.
- [37] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp.2169–2178.
- [38] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial-bag-of-features, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 3352– 3359.
- [39] R. Weber, H. Schek, S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: International Conference on Very Large Data Bases, 1998, pp. 194–205.
- [40] H. Kang, M. Hebert, T. Kanade, Image matching with distinctive visual vocabulary, in: IEEE Workshop on Applications of Computer Vision, 2011, pp. 402–409.
- [41] J.C. raina, A. Traina, C. Faloutsos, B. Seeger, Fast indexing and visualization of metric data sets using slim-trees, Transactions on Knowledge and Data Engineering 14(2)(2002)244–260.
- [42] Wang, Y., Zhang Bin, Z., Ge, Y., “*The Invariant Relations of 3D to 2D Projection of Point Sets*”. Journal of Pattern Recognition Research, volumen 3, número 1 (2008)
- [43] J.E González, Detección y asociación automática de puntos característicos para diferentes aplicaciones.(2009).
- [44] R. Aracil López, Desarrollo de un sistema cognitivo de visión para la navegación robótica. (2112).
- [45] Nilsback, M-E. and Zisserman, A. A Visual Vocabulary for Flower Classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006).

[46] Toolbox Matlab version R2015a.

[47] Matlab version R2010a.