El tribunal d'avaluació d'aquest Projecte Fi de Carrera, reunit el dia _____, ha acordat concedir la següent qualificació:

President: _____

Vocal: _____

Secretari: _____

UAB

Els sotasignants, José A. López-Salcedo[1], Professor de l'Escola d'Enginyeria de la Universitat Autònoma de Barcelona (UAB), i David Marimon Sanjuan, tutor de l'alumne a l'empresa Telefónica I+D.

CERTIFIQUEN:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumne Arturo Bonnín Llofriu.

I, perquè consti a tots els efectes, signen el present certificat.

Bellaterra, 19 de febrer de 2010.

Signatura: José A. López-Salcedo David Marimon

1 En cas d'absència, el professor Josep Parrón serà el seu substitut.

Resum:

En aquest treball realitzem un estudi sobre la detecció y la descripció de punts característics, una tecnologia que permet extreure informació continguda en les imatges. Primerament presentem l'estat de l'art juntament amb una avaluació dels mètodes més rellevants. A continuació proposem els nous mètodes que hem creat de detecció i descripció, juntament amb l'algorisme òptim anomenat DART, el qual supera l'estat de l'art. Finalment mostrem algunes aplicacions on s'utilitzen els punts DART. Basant-se en l'aproximació de l'espai d'escales Gaussià, el detector proposat pot extreure punts de distint tamany invariants davant canvis en el punt de vista, la rotació i la iluminació. La reutilització de l'espai d'escales durant el procés de descripció, així com l'ús d'estructures simplificades i optimitzades, permeten realitzar tot el procediment en un temps computacional menor a l'obtingut fins al moment. Així s'aconsegueixen punts invariants i distingibles de forma ràpida, el qual permet la seva utilització en aplicacions com el seguiment d'objectes, la reconstrucció d'escenaris 3D i en motors de cerca visual.

Resumen:

En este trabajo realizamos un estudio sobre la detección y la descripción de puntos característicos, una tecnología que permite extraer información contenida en las imágenes. Primeramente presentamos el estado del arte junto con una evaluación de los métodos más relevantes. A continuación proponemos los nuevos métodos que hemos creado de detección y descripción, junto con el algoritmo óptimo llamado DART, el cual supera el estado del arte. Finalmente mostramos algunas aplicaciones donde se utilizan los puntos DART. Basándose en la aproximación del espacio de escalas Gaussiano, el detector propuesto es capaz de extraer puntos de distinto tamaño invariantes ante cambios de punto de vista, rotación e iluminación. La reutilización del espacio de escalas durante el proceso de descripción, así como el uso de estructuras simplificadas y optimizadas, permiten realizar todo el proceso en un tiempo computacional menor al obtenido hasta la fecha. Así se logran puntos invariantes y distinguibles de forma rápida, lo cual permite su utilización en aplicaciones como el seguimiento de objetos, la reconstrucción de escenarios 3D y en motores de búsqueda visual.

Summary:

An study about keypoint detection and description is performed in this work. This tecnology allows the extraction of information contained in images. Firstly, we show the state of the art and an evaluation of methods. Secondly, we propose new keypoint detection and description methods and an optimum algorithm socalled DART, which outperforms the state of the art. Finally some applications using DART are shown. Based on the Gaussian scale-space approximation, the proposed detector extracts scale, rotation and ilumination invariant keypoints. Re-using the scale-space, jointly with simplified and optimized structures, the whole process can be run in less computational time than that obtained to date. Invariant keypoints are extracted quickly and distinguishable, which allows its use in applications such as object tracking, 3D scene reconstruction and visual search engines.

Estudio sobre la extracción de puntos característicos en imágenes y sus aplicaciones.

Arturo Bonnín Llofriu

11 de febrero de 2010

Índice general

Capítulo 1

Introducción

El reconocimiento de los contenidos de una imagen es uno de los principales objetivos en el campo de la visión por computador. A partir de sus contenidos se puede encontrar la correspondencia entre imágenes, es decir, su similitud. Esta tarea puede ser realizada por los humanos, siendo capaces de reconocer los objetos, escenas o actividades que aparecen en una imagen sin mayor dificultad. Sin embargo, para los computadores este caso no es trivial, sobre todo si se trata del caso general en que las situaciones y los objetos son arbitrarios. En el momento en que un ordenador es capaz de realizar dicha tarea, aparecen numerosas aplicaciones que hubiesen resultado imposibles anteriormente, como la búsqueda de objetos a partir de sus contenido en una base de datos de imágenes, el seguimiento de un objeto en un vídeo o la detección de elementos en un determinado escenario.

Una de las formas de obtener los contenidos de una imagen es la extracción de puntos característicos, ya que con ellos se consigue representar la información más relevante de una escena. Durante los últimos años se han investigado numerosos métodos para conseguir puntos característicos invariantes ante las transformaciones geométricas más comunes, robustos frente a situaciones adversas, distinguibles y estables. Al mismo tiempo, las investigaciones han sido enfocadas a realizar esta tarea de forma rápida computacionalmente para poder adaptarse a los requerimientos de las aplicaciones.

El objetivo de este trabajo es crear un detector y descriptor de puntos característicos superior a los métodos actuales, siendo a la vez capaz de ejecutarse en un menor periodo de tiempo. Para ello se ha realizado un estudio del estado del arte actual, con el cual se ha concluido que los mejores algoritmos son los basados en la función Gaussiana para la obtención de un espacio de escalas que permita extraer puntos de distintos tamaños. Entre los métodos descriptivos existentes, los más efectivos han resultado ser los basados en la distribución de componentes característicos.

Concretamente, SIFT [14] y SURF [2] son los algoritmos con mejores prestaciones.

El primero obtiene los mejores resultados pero requiere de tiempos de cómputo que le impiden su utilización en aplicaciones con restricciones de tiempo. SURF, por otra parte, mediante la aproximación de métodos es capaz de ejecutarse rápidamente, no consiguiendo sin embargo los mismos resultados que SIFT en determinadas situaciones.

Las investigaciones llevadas a cabo en este trabajo se centran en el estudio de nuevos métodos de detección y descripción de puntos con el fin de lograr un algoritmo que obtenga puntos característicos invariantes ante cambios de escala, rotaciones e iluminación; así como un método de descripción capaz de formar puntos distinguibles en un tiempo de cómputo bajo. Debido a que los puntos extraídos por SIFT han demostrado tener las mejores características, este trabajo tiene en cuenta su estructura general. Asimismo, inspirándonos en los pasos utilizados por SURF, hemos estudiado la forma de reducir la complejidad del algoritmo con una mínima repercusión en los resultados.

Este trabajo está dividido en siete capítulos. El primero es el presente, mostrando una introducción a la problemática estudiada. El segundo contiene los conceptos generales en la detección y descripción de puntos característicos. En el tercero se explican los conceptos básicos de la detección de puntos característicos, el estado del arte actual y finalmente se muestra una evaluación de los detectores más representativos en la actualidad. El cuarto capítulo tiene la misma estructura que el capítulo anterior, aplicado a la descripción de puntos característicos. Es en el quinto donde se propone un nuevo método de detección y descripción de puntos, así como un algoritmo óptimo llamado DART [18]. En este apartado también se muestran los experimentos realizados, así como una evaluación que muestra los buenos resultados de DART frente a los mejores algoritmos existentes en el momento. En el sexto capítulo aparecen las aplicaciones donde se ha probado DART y por último en el séptimo se describen las conclusiones a las que se ha llegado tras la realización de este trabajo.

Capítulo 2

Conceptos generales en la detección y descripción de puntos característico en imágenes

En este capítulo se muestran los conceptos generales de la detección y descripción de puntos característicos para así poder entrar en detalle en los capítulos posteriores. Primeramente se hace una breve introducción y a continuación se detallan los conceptos mencionados.

2.1. Introducción

La extracción de puntos característicos es un paso usado frecuentemente para llegar a definir el contenido de una imagen. El primer paso consiste en detectar la posición de cada punto. Posteriormente se define su contenido mediante una o más características.

- **Detección:** La detección consiste en encontrar el lugar de la imagen donde existen puntos representativos.
- **Descripción** La descripción se realiza tras la detección. Una vez localizado el lugar, se procede a describir mediante una o más características la región situada alrededor del punto característico detectado. Estas características se almacenan frecuentemente en un vector para su uso posterior.

La detección es el proceso en el cual se localizan las regiones de la imagen que contienen unas características determinadas, debido a su forma o textura. Actualmente la mayoría de detectores buscan regiones donde aparecen esquinas, bordes o formas específicas. Cada

punto representa una región característica de la escena, pudiéndose extraer miles de puntos de una sola imagen. Estos puntos no están distribuidos de forma uniforme en la imagen, sino que se concentran en las zonas donde existen contenidos con más información. De esta forma, cualquier imagen queda representada por sus puntos característicos.

Existen numerosas técnicas para detectar y describir puntos. Mostrándose las más representativas de ellas en este trabajo. La gran mayoría de algoritmos trabajan sobre imágenes en escala de grises (conocidas popularmente como blanco y negro), por lo tanto, al referirnos a imágenes en este documento, siempre nos estaremos refiriendo a las de este tipo.

Los puntos característicos se utilizan para determinar la similitud entre imágenes. A partir de sus descriptores, los puntos de una imagen se comparan con los de otra y así se establece la correspondencia entre ellos. En caso de aparecer un mismo escenario en ambas imágenes, los puntos característicos que los describen van a coincidir y por lo tanto dicho escenario va a ser identificado como el mismo en las dos imágenes.

2.2. Detección de puntos característicos

El objetivo de un detector de puntos característicos es encontrar los puntos de una imagen en los cuales están situadas las regiones más representativas. Existen numerosos detectores de puntos que se basan en la búsqueda de distintos elementos representativos tales como esquinas, bordes, regiones circulares, etc.

Los retos actuales de los detectores son encontrar regiones características de distintos tamaños y con cualquier orientación. De esta forma se consigue ser invariante a tales efectos, lo cual permite detectar las mismas regiones en escenas donde varía el tamaño de los objetos y donde puedan estar orientados de cualquier forma. Asimismo es importante que la detección de puntos no se vea afectada por cambios de iluminación en la escena ni por el grado de definición de ésta.

Los detectores se evalúan midiendo el número de regiones repetidas en distintas imágenes de una misma escena que ha sufrido transformaciones geométricas y fotométricas, es decir, midiendo la invariabilidad ante estos cambios. Esta medición se conoce como repetibilidad.

Otro factor importante al valorar un detector es el tiempo computacional que necesita para extraer los puntos. Existe un compromiso entre repetibilidad y tiempo de ejecución, ya que al intentar mejorar la primera puede que el tiempo requerido aumente llegando a tiempos demasiado altos para poder ser utilizado en algunas aplicaciones, como las que necesitan ser ejecutadas en tiempo real en un vídeo.

2.3. Descripción de puntos característicos

Tras localizar la posición y el tamaño de los puntos característicos, el siguiente paso es describir dichos puntos. Este proceso se lleva a cabo mediante una descripción de la región que rodea a cada punto, concepto conocido como descripción local. La similitud de los descriptores pertenecientes a distintas imágenes indica la similitud de éstas. Al encontrar descriptores similares en dos imágenes distintas, se detecta una coincidencia y con el número de éstas se obtiene el grado de similitud entre las dos imágenes.

Un descriptor debe definir de forma única las características de una región para así lograr ser distinguible. En caso contrario, no sería posible distinguir regiones que representan fenómenos distintos en una imagen. Asimismo, una región que varía en orientación, tamaño, definición o iluminación no debe modificar su descripción para así mantenerse invariante ante dichos cambios. Por otra parte, la descripción de puntos característicos debe realizarse de la forma más breve posible, sin que ello comprometa los resultados, para así evitar posteriores cargas computacionales.

Existen diversos métodos para describir las características de una región de la imagen. El método más sencillo es crear un vector con los valores de intensidad de los píxeles contenidos en la región. Sin embargo, de esta forma no se consigue invariabilidad ante los cambios anteriormente mencionados ya que un mínimo cambio en la imagen provoca grandes cambios en el descriptor. Asimismo esta descripción obtiene un vector de grandes dimensiones, lo cual no es eficiente ya que su posterior procesamiento precisará de altas cargas computacionales y de una alta utilización de espacio en memoria. Por este motivo los descriptores existentes en la actualidad utilizan métodos más complejos.

Capítulo 3

Detección de puntos característicos: conceptos básicos, estado del arte y evaluación

El primer paso para la obtención de puntos característicos consiste en localizar las regiones más representativas de una imagen. Éstas pueden ser esquinas, bordes, regiones circulares, etc. Primeramente se muestran los conceptos básicos, a continuación el estado del arte y por último una comparativa de técnicas.

3.1. Conceptos básicos

Al hablar de detectores de puntos característicos es imprescindible definir previamente algunos conceptos que tienen un papel muy importante. Éstos son la repetibilidad, el espacio de escalas, los puntos gota y la función Gaussiana.

Los detectores tienen como objetivo ser invariantes ante los cambios más frecuentes en una escena, suceso medido mediante la repetibilidad. Uno de estos cambios es la escala, la que define el tamaño de una región representativa de la imagen. Para conseguir invariabilidad en el tamaño o en la escala, los detectores utilizan el llamado espacio de escalas. En este trabajo también se estudia la función Gaussiana ya que tiene también mucha importancia debido a su utilización en la creación del espacio de escalas.

Por otro lado, un detector busca elementos representativos en las imágenes, los llamados "features" en inglés. Existen varios tipos de elementos, pero el más utilizado actualmente es el "blob", que se traduce en lengua castellana como "punto gota" debido a su forma aproximadamente circular.

3.1.1. Repetibilidad

La repetibilidad mide el grado en que una misma región característica puede localizarse sin verse afectada por características tales como su tamaño, orientación, definición o iluminación. De esta forma el punto va a repetirse en todas las imágenes de una escena que sufra dichas variaciones.

Para evaluar la repetibilidad de un detector primeramente se escogen dos imágenes donde aparezca una misma escena afectada por una transformación geométrica o fotométrica. A continuación se extraen los puntos característicos de cada una de las imágenes y se mide el número de puntos correspondientes. Para saber la correspondencia entre puntos debe saberse de antemano qué tipo de transformación ha sufrido la imagen para así conocer la posición y el tamaño que tendrá un punto en la imagen transformada. Para ello se utiliza una matriz de transformación conocida como homografía, definida en detalle en la sección 3.3.1.

La repetibilidad se muestra como un porcentaje entre el número de puntos extraídos con correspondencia y el número de puntos que deberían tener correspondencia según la transformación dada. Un detector ideal tiene una repetibilidad del 100 %, ya que detecta en ambas escenas todos los puntos comunes. Los puntos que, debido a la transformación, han dejado de existir en una de las escenas, no se tienen en cuenta en la evaluación.

3.1.2. Espacio de escalas

Una de las características que a lo largo de los últimos años ha cobrado más importancia es la invariancia ante cambios de escala. Un objeto puede aparecer en cualquier tamaño y un buen detector debe ser capaz de reconocer que se trata únicamente de un cambio en la escala de la escena (ver figura 3.1). Para conseguir este tipo de invariancia, la mayoría de detectores utilizan sistemas de filtrado en un espacio de escalas.

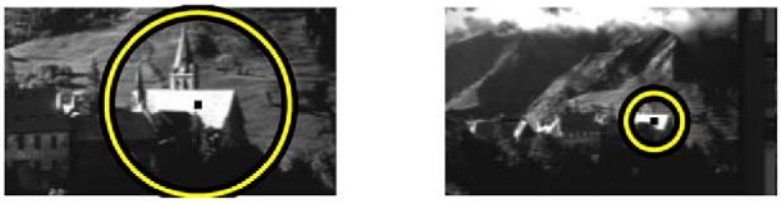


Figura 3.1: Dos imágenes mostrando una misma escena desde dos distancias. El círculo marca la región característica. Un detector invariable a la escala es capaz de reconocer que las dos regiones pertenecen al mismo objeto. Figura extraída de [22].

El espacio de escalas permite extraer estructuras características con su correspondiente escala en una imagen. Ésta es una tarea primordial, ya que muchas de las aplicaciones

existentes en el mundo de la visión por computador, tales como el reconocimiento y seguimiento de objetos, necesitan de este proceso en su fase preliminar para poder efectuarse.

La representación del espacio de escalas fue introducida por vez primera en el año 1983 por Witkin [33]. Gracias a su utilización se puede tratar con estructuras características en distintas escalas, es decir, a tamaños distintos respecto el tamaño total de la imagen.

El espacio de escalas es una representación de los datos donde éstos se muestran en forma de familia de señales progresivamente suavizadas para así poder ser representado en distintas escalas.

Dada una señal bidimensional continua $f : \Re^2 \rightarrow \Re$, el espacio de escalas $L : \Re^2 X \Re_+ \rightarrow \Re$ se define como la solución a la ecuación de difusión

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}(\partial_{x_1 x_1} + \partial_{x_2 x_2})L \tag{3.1}$$

con la condición inicial $L(:;0) = f$

$$L(:;t) = g(:;t) * f \tag{3.2}$$

donde

$$g(x;t) = \frac{1}{2\pi t} e^{\frac{-(x_1^2+x_2^2)}{2t}} \tag{3.3}$$

y $x = (x_1, x_2) \in \Re^2$. t, una coordenada en el espacio de escalas $\Re_+$, es el cuadrado de la desviación estándar del núcleo de la Gaussiana $t = \sigma^2$. Por lo tanto, la señal de entrada f consigue variar el nivel de escala convolucionándose con una señal Gaussiana de σ variable.

Al trabajar con imágenes, el espacio de escalas se aproxima mediante un conjunto de imágenes con diferentes niveles de filtrado a partir de la imagen original que se desea procesar. A medida que aumenta σ, se produce un aumento en el suavizado de la imagen. De esta forma los niveles más bajos (t cercano a 0) contienen más detalles de la imagen original, mientras que en los altos niveles ($t >> 0$) la imagen aparece difuminada y tan sólo pueden diferenciarse los objetos de gran tamaño.

A partir del espacio de escalas pueden extraerse las estructuras de tamaño distinto que aparecen en una imagen. Una de las principales estructuras que se hacen evidentes en el

espacio de escalas son los puntos gota. Éstos son regiones más claras o más oscuras que el espacio que los rodea. El objetivo al trabajar con espacio de escalas es determinar tales estructuras a distintos tamaños, poder compararlas y determinar cuál es el nivel de escala que las caracteriza, es decir, el tamaño real de la estructura en cuestión. Una explicación más detallada aparece en la sección 3.1.4

3.1.3. Filtro Gaussiano

La función Gaussiana ha sido ampliamente utilizada en el campo de la visión por computador gracias a sus características tales como la orientividad, simetría y separabilidad, entre otras. Un filtro Gaussiano puede aplicarse en un espacio multidimensional en distintas direcciones y, concretamente, aplicado en un espacio bidimensional presenta la ventaja de ser simétrico circularmente. Por otro lado, la separabilidad lineal es la característica que permite aplicar un filtro en un espacio n-dimensional con n filtros unidimensionales. Así, un filtro Gaussiano puede aplicarse sobre una imagen bidimensional con 2 cálculos unidimensionales. Es decir, el efecto de aplicar una matriz bidimensional puede conseguirse utilizando series de Gaussianas unidimensionales en dirección horizontal y, posteriormente, repetirlo en dirección vertical. De esta forma se consigue reducir el tiempo computacional.

Para la generación del espacio de escalas (sección 3.1.2), los filtros Gaussianos y sus transformaciones han demostrado durante los últimos años ser la mejor y prácticamente única opción. Para ello se debe utilizar una versión normalizada de los filtros en todas las escalas para que todas ellas sean comparables.

Función Gaussiana y derivadas

La función Gaussiana en una sola dimensión se define, tal como se muestra en [31], como

$$G(t) = e^{\frac{-t^2}{2\sigma^2}} \tag{3.4}$$

Sin embargo, debe utilizarse su versión normalizada para asegurar que el nivel medio de señal no va a ser modificado tras la aplicación del filtro

$$G(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}} \tag{3.5}$$

El área bajo la curva de la función es igual a 1 y el único efecto observable sobre la señal es un filtrado de altas frecuencias. En una imagen el efecto es parecido a un desenfoque.

La primera derivada de la función Gaussiana es

$$G'(t) = \frac{1}{\sqrt{2\pi}\sigma}\frac{-t}{\sigma^2}e^{-\frac{t^2}{2\sigma^2}} = -\frac{t}{\sqrt{2\pi}\sigma^3}e^{-\frac{t^2}{2\sigma^2}} = -\frac{t}{\sigma^2}G(t) \tag{3.6}$$

mientras que la segunda derivada responde a la expresión

$$G''(t) = \frac{1}{\sqrt{2\pi}\sigma}\frac{t^2-\sigma^2}{\sigma^4}e^{-\frac{t^2}{2\sigma^2}} = \frac{t^2-\sigma^2}{\sqrt{2\pi}\sigma^5}e^{-\frac{t^2}{2\sigma^2}} = \frac{t^2-\sigma^2}{\sigma^4}G(t) \tag{3.7}$$

En la figura 3.2 pueden observarse los gráficos de las 3 funciones definidas anteriormente.

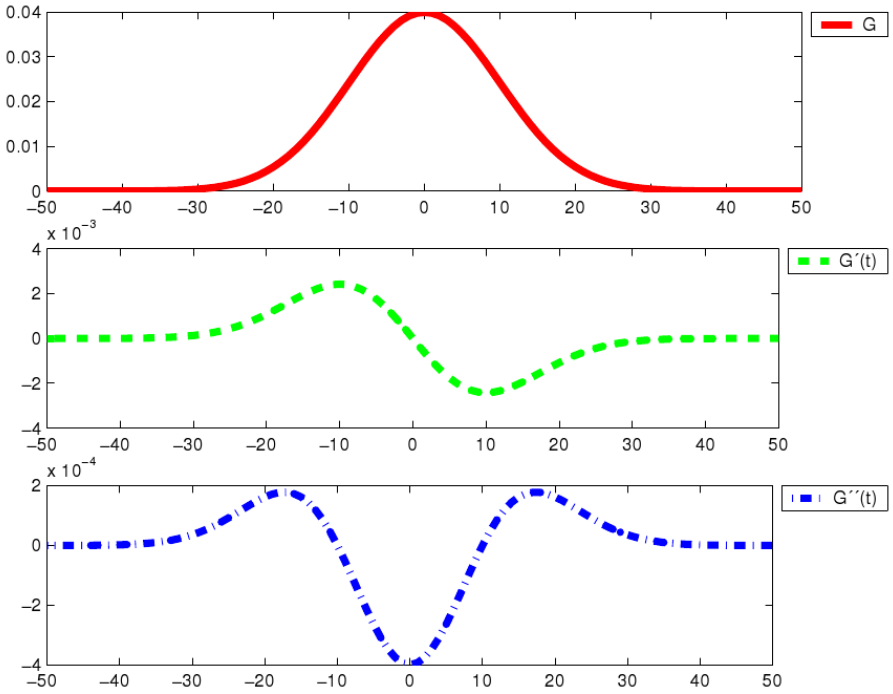


Figura 3.2: Función Gaussiana y sus derivadas. De arriba a abajo: Función Gaussiana normalizada, primera derivada de la función Gaussiana normalizada y segunda derivada de la función Gaussiana normalizada. Figura extraída de [31].

Normalización sobre escalas

Las derivadas Gaussianas son ampliamente utilizadas en el proceso de extracción de puntos característicos. El problema de dichas derivadas es que no están normalizadas ante cambios de escala del filtro. En este caso concreto y en los casos futuros que aparecen en este documento, la escala la determina la desviación estándar de la función Gaussiana (σ).

Al variar la escala, las respuestas convolucionables de los filtros también varían, debido a que la amplitud en las escalas altas es menor. De esta forma es imposible conseguir una comparativa justa de puntos máximos a través del espacio de escalas, por lo tanto es necesaria una normalización sobre escalas.

Las funciones de derivadas Gaussianas normalizadas son las siguientes, donde nG indica Gaussiana normalizada. Para seguir el procedimiento necesario para llegar a dichas expresiones ver [31].

$$nG'(t) = \sigma \frac{-t}{\sqrt{2\pi}\sigma^3} e^{-\frac{t^2}{2\sigma^2}} = -\frac{t}{\sqrt{2\pi}\sigma^2} e^{-\frac{t^2}{2\sigma^2}} \tag{3.8}$$

$$nG''(t) = \sigma^2 \frac{t^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{t^2}{2\sigma^2}} = \frac{t^2 - \sigma^2}{\sqrt{2\pi}\sigma^3} e^{-\frac{t^2}{2\sigma^2}} \tag{3.9}$$

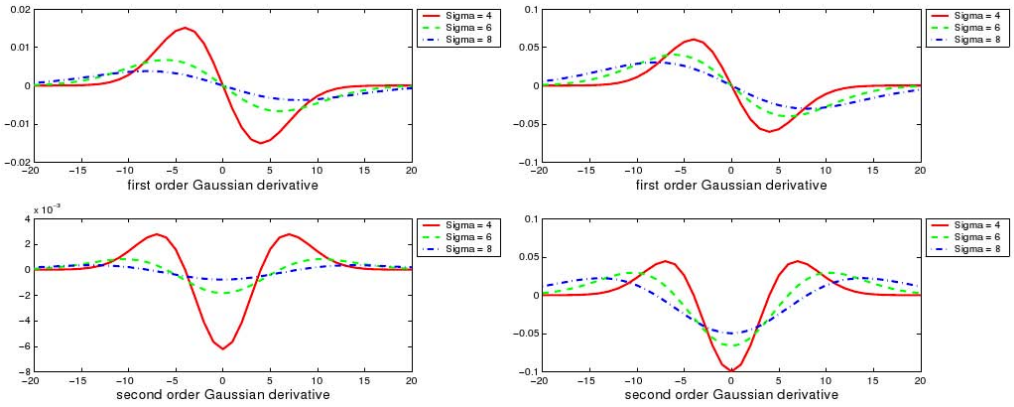


Figura 3.3: Izquierda: Primera y segunda derivada de la función Gaussiana sin normalizar. Derecha: Primera y segunda derivada de la función Gaussiana normalizadas. Las versiones normalizadas muestran como las amplitudes se igualan por distintos valores de sigma. Resultados mostrados para sigma igual a 4, 6 y 8. Figura extraída de [31].

Gracias a la normalización de las derivadas se consigue obtener los mismos niveles de amplitud para distintas escalas. De este modo se pueden realizar comparativas en el

espacio de escalas. La figura 3.3 muestra cómo actúa la normalización para Gaussianas con valores sigma de 4, 6 y 8.

3.1.4. Puntos gota

El punto gota es una de las estructuras más comunes a extraer en una imagen cuando se desea obtener un punto característico. Durante los últimos años otras estructuras como esquinas y bordes han sido utilizadas, pero son los puntos gota los que han obtenido mejores resultados.

Un punto gota es un máximo o un mínimo local en una función de una o varias dimensiones. En el caso de las imágenes se trata de un extremo en el valor de intensidad de una zona de la imagen. En [12] un punto gota se define como una región con valores de intensidad mayores o menores que su entorno, siendo, respectivamente, un punto gota claro o oscuro. Para definir el tamaño se considera que un punto gota debe extenderse hasta juntarse con otro.

En la figura 3.4 pueden verse algunos ejemplos de puntos gota oscuros en la camisa del hombre, marcados con círculos. Igualmente, en la figura 3.7 aparece la extracción de puntos gota oscuros a distintos niveles de escala para un espacio de escalas de 12 niveles. Puede apreciarse cómo al aumentar el nivel de escala los puntos gota aumentan su tamaño, pasando de extraer las teclas del teléfono en la tercera escala a todo el conjunto de teclas y la calculadora en la octava.

Como se ha comentado anteriormente, es importante determinar la relevancia de un punto gota. En la imagen 3.4 tan sólo aparecen los puntos gota más relevantes. Para ello existen diversas técnicas de extracción de puntos gota que permiten dar una puntuación a cada uno de ellos.

Tal como se explica en la sección 3.1.2, el espacio de escalas se crea mediante la convolución de imágenes con filtros Gaussianos con distinta σ. Con dicha convolución se obtiene un conjunto de imágenes como las que aparecen en la figura 3.7. A partir de ahí existen varios métodos para extraer puntos gota de las imágenes filtradas.

Las imágenes filtradas a una escala “t” responden a la expresión

$$L(x, y, t) = g(x, y, t) * f(x, y) \tag{3.10}$$

donde “g” es la función Gaussiana y “f” es la imagen de entrada.

- **Laplaciana:** Este método consiste en aplicar el operador Laplaciano sobre la imagen filtrada.

Figura 3.4: Ejemplo de puntos gota oscuros. Los puntos gota aparecen rodeados por un círculo en la camisa del hombre. Figura extraída de [3].

El operador Laplaciano se expresa como

$$\nabla^2 L(x, y; t) = L_{xx} + L_{yy} \tag{3.11}$$

Donde L_{xx} y L_{yy} son las segundas derivadas de la imagen filtrada a una escala t, en sentido horizontal y vertical, respectivamente.

Los puntos gota oscuros de tamaño alrededor de $\sqrt{t}$ aparecen como fuertes respuestas positivas, mientras que las respuestas negativas pertenecen a los puntos gota claros.

Para la extracción de puntos gota en escalas múltiples surgen problemas con la fórmula 3.11, ya que los puntos gota extraídos a escalas altas tienen valores menores. Para ello existe el operador Laplaciano normalizado a escalas:

$$\nabla^2_{norm} L(x, y; t) = t(L_{xx} + L_{yy}) \tag{3.12}$$

De esta forma los puntos extremos de $\nabla^2_{norm} L \nabla^2_{norm} L$, es decir, los puntos máximos o mínimos en $(x, y; t)$, serán puntos gota estables.

- **Determinante:** El determinante de la matriz Hessian se aplica sobre el conjunto de imágenes del espacio de escalas para obtener puntos gota.

El operador Hessian, matriz mostrada en la ecuación 3.13, se computa mediante las derivadas parciales de segundo orden, obtenidas en un entorno local.

$$H = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} \tag{3.13}$$

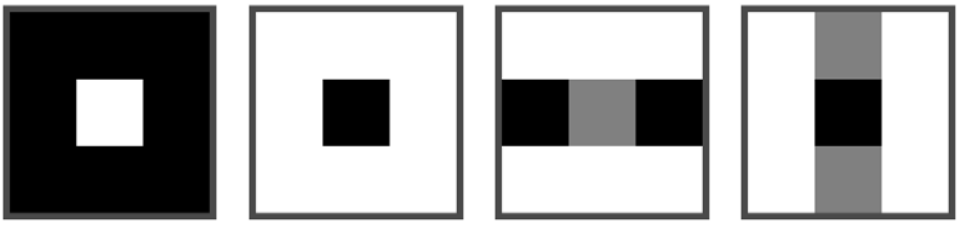
Donde L_{xy} es la segunda derivada cruzada de la función filtrada a una escala t.

A continuación se aplica el determinante de dicha matriz.

$$detH = t^2(L_{xx}L_{yy} - L_{xy}^2) \tag{3.14}$$

Cabe destacar que la ecuación 3.14 está normalizada para todas las escalas, de forma que los valores obtenidos son comparables independientemente del valor de escala.

En el caso del determinante de la matriz Hessian, tan sólo los máximos corresponden a puntos gota en la imagen, ya que un mínimo significaría que L_{xx} y L_{yy} tienen signos contrarios y dicha estructura no es en absoluto un punto gota. Ejemplos de estructuras con valores máximos y mínimos en el determinante Hessian pueden apreciarse en la figura 3.5



(a) (b) (c) (d)

Figura 3.5: Las figuras a) y b) son puntos gota que representan estructuras con valores máximos en el determinante Hessian. Las figuras c) y d) son dos ejemplos de estructuras con valores mínimos en el determinante Hessian.

El componente L_{xy}, correspondiente a la derivada de segundo orden cruzada, al obtener altos valores indica que se trata de una estructura diagonal alargada. Estas formas no se corresponden con puntos gota y, por lo tanto, el componente L_{xy} se utiliza para penalizar una estructura. En la figura 3.6 se pueden ver dos ejemplos gráficos.

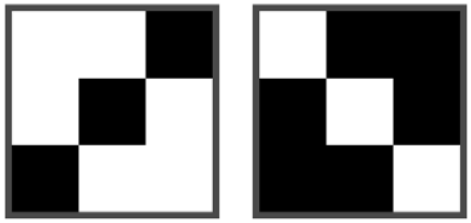


Figura 3.6: Dos estructuras con un alto valor en su componente L_{xy} de la matriz determinante de Hessian, así como en el producto $L_{xx}L_{yy}$. El valor alto de L_{xy} penaliza la estructura por no corresponderse con un punto gota.

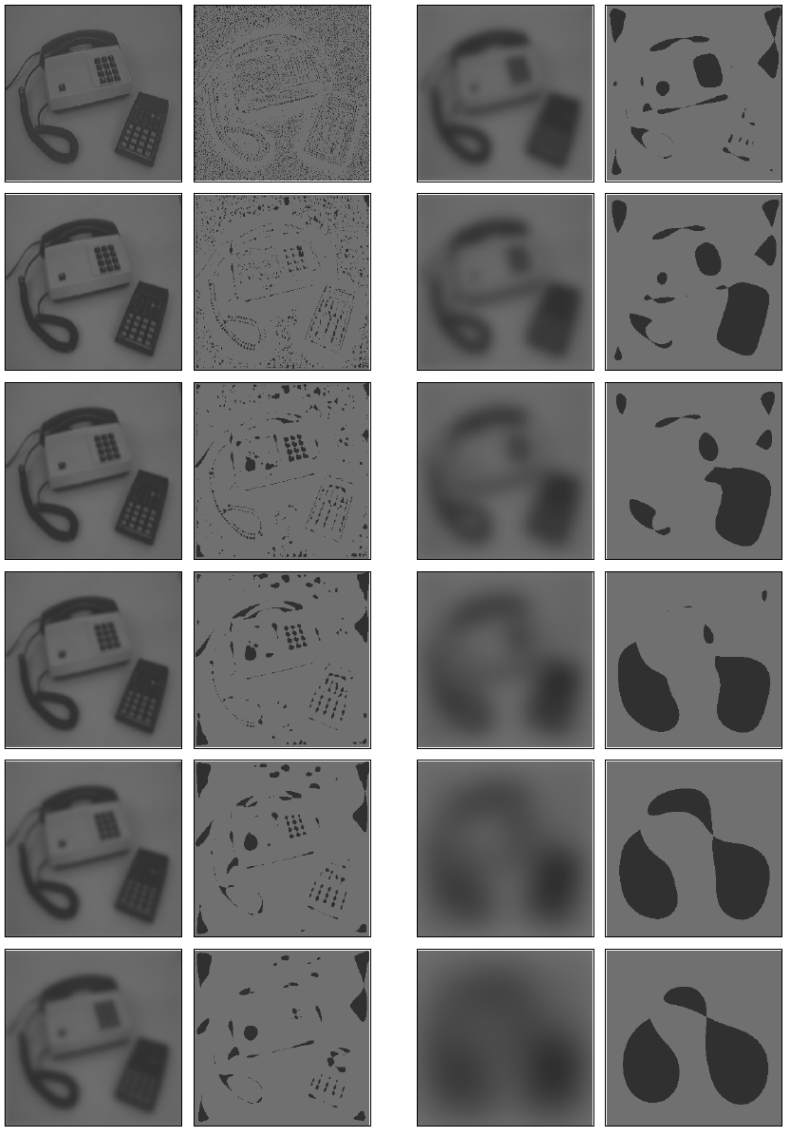


Figura 3.7: Espacio de escalas de una imagen y extracción de puntos gota. Cada una de las dos columnas muestra en la izquierda la imagen filtrada a una escala determinada y en la derecha una imagen binaria mostrando los puntos gota oscuros obtenidos. Los niveles de escala, mostrados de arriba a abajo en cada una de las 2 columnas, son t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 y 1024. Figura extraída de [12].

3.2. Estado del arte de los detectores de puntos característicos

3.2.1. Introducción

Los numerosos detectores de puntos característicos existentes se basan en la búsqueda de unas características determinadas en las imágenes para así conseguir puntos que sean repetibles. Existen detectores que buscan esquinas, otros que obtienen crestas y bordes y finalmente los más comunes actualmente, los basados en la búsqueda de puntos gota, tal como se explica en la sección 3.1.4.

Para conseguir un buen porcentaje de repetibilidad un detector debe ser invariante ante los típicos cambios que pueden sufrir las imágenes. A la hora de evaluar los detectores se deben someter a test ante tales cambios para comprobar su robustez.

Muchos de los detectores mostrados a continuación tienen la característica de ser invariantes ante cambios del punto de vista de la escena, los llamados detectores afines. Esto es debido a que la comparativa más completa de detectores [22] se hizo en un momento en que esta característica se tenía muy en cuenta a la hora de escoger un buen detector. Actualmente los detectores han tomado un rumbo distinto ya que se vio que la capacidad de invariancia ante cambios afines en la escena tan sólo era útil en situaciones muy pronunciadas, a partir de ángulos de 50 grados.

Los detectores SIFT (sección 3.2.7) y SURF (sección 3.2.8) son los más utilizados actualmente. Éstos no son invariantes ante cambios en el punto de vista debido a que esto conlleva un alto coste computacional sin aportar suficientes mejoras. Ambos son detectores de puntos gota basados en aproximaciones del espacio de escalas Gaussiano.

SURF presenta resultados similares a SIFT y es actualmente el algoritmo que obtiene una mejor relación de repetibilidad ante coste computacional. Estos dos algoritmos están explicados con más detalle ya que el método propio propuesto en la sección 5 utiliza esquemas similares.

SIFT y SURF también describen los puntos característicos detectados. La descripción de puntos mediante estos algoritmos se muestra en la sección 4.

3.2.2. Detector Harris

Harris es un detector que obtiene puntos característicos en el espacio de escalas. El máximo sobre escalas se obtiene utilizando el operador Laplaciano. Éste demostró ser el mejor en los estudios realizados en [20]. Para obtener la escala característica de una región, se busca un extremo entre escalas, tal como se muestra en [13].

Existen versiones afín de este detector, la cual tiene métodos específicos para detectar puntos invariantes ante cambios del punto de vista de la cámara respecto a la imagen.

El detector Harris busca esquinas en la imagen, es uno de los llamados "corner detector". Para ello se basa en la matriz de autocorrelación (ecuación 3.15).

$$A = \sum_{x,y} w(x,y) \begin{pmatrix} L_x^2 & L_{xy} \\ L_{xy} & L_y^2 \end{pmatrix} \tag{3.15}$$

Donde L_x y L_y son las primeras derivadas sobre ambos ejes y L_{xy} es la derivada cruzada. La función w es una Gaussiana circular utilizada para ponderar las muestras, dando mayor importancia a las centrales.

La matriz mostrada en la ecuación 3.15 obtiene la distribución del gradiente en un entorno local. Los elementos característicos se encuentran en los puntos de máxima curvatura, los cuales son máximos en la ecuación 3.16, basada en los valores de autocorrelación de la matriz A (ecuación 3.15).

$$R = det(A) - \alpha * tr^2(A) \tag{3.16}$$

Donde *det* es el determinante y *tr* es la traza de la matriz A.

Por lo tanto, los puntos que presenten un máximo en R en su entorno inmediato serán seleccionados como candidatos.

3.2.3. Detector de regiones basado en bordes (EBR)

El detector de regiones basado en bordes o Edge-Based Region Detector (EBR) [29, 30] se basa en la localización de bordes. La principal ventaja de estas estructuras es su estabilidad, ya que se mantienen ante cambios de luz, cambios de vista y de escala.

Para obtener las regiones de interés se combina el detector de esquinas Harris [16] y el detector de bordes Canny [4]. Éstas se calculan en diferentes niveles de escala para así conseguir puntos de distinto tamaño.

El EBR dispone de un sistema de obtención de regiones afín, lo cual le proporciona buenos resultados antes los cambios de punto de vista. Partiendo de una esquina obtenida con el detector de esquinas Harris y mediante la utilización de bordes, se acaba obteniendo una región en forma de paralelógramo, tal como se muestra en la figura 3.8.

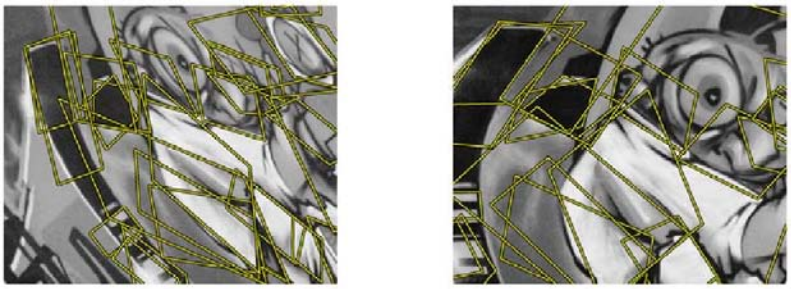


Figura 3.8: Regiones extraídas mediante el detector EBR. Figura extraída de [22].

3.2.4. Detector de regiones basado en extremos de intensidad (IBR)

El detector de regiones basado en extremos de intensidad o Intensity Extrema-based Region Detector (IBR) ([28] y [30]) se basa en la búsqueda de puntos extremos en intensidad, es decir, en este caso no se realiza ninguna modificación sobre la imagen de entrada para obtener el punto candidato a ser extraído, sino que se utiliza directamente el valor del píxel. La búsqueda de extremos en intensidad se realiza en distintos niveles de escala. Este detector también utiliza métodos afines para detectar cambios en el punto de vista.

IBR busca un punto extremo en intensidad y, a continuación, busca radialmente la zona del entorno que sigue manteniendo valores extremos en intensidad, tal como se muestra en la figura 3.9.

Una vez obtenida la región, la cual tiene forma indefinida, se aproxima en una elipse.

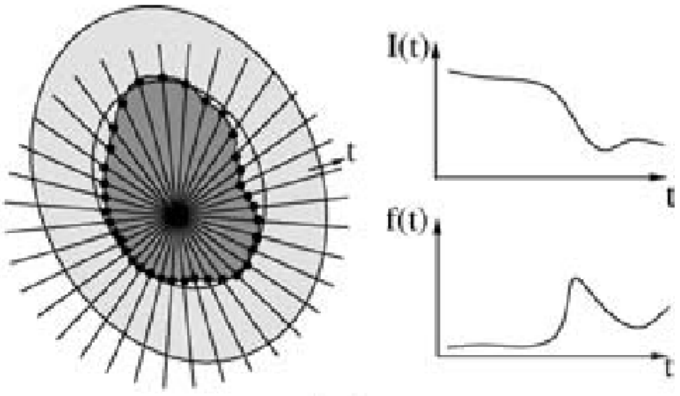


Figura 3.9: Regiones extraídas mediante el detector IBR. En la parte izquierda aparece la región extraída mientras que en la derecha puede verse la función intensidad en función de la distancia desde el centro del punto. Figura extraída de [22].

3.2.5. Detector de regiones extremas máximamente estables (MSER)

El detector de regiones extremas máximamente estables o Maximally Stable Extremal Region Detector (MSER) [19] es otro detector afín que crea regiones formadas por componentes conectados a partir de una imagen modificada utilizando un umbral. Se trata de una región extrema ya que todos los píxeles que la forman son o más brillantes o más oscuros que todos sus vecinos exteriores. Durante el proceso de modificación de la imagen de entrada mediante la utilización de un umbral se consiguen las regiones más estables de la imagen.

Las regiones extraídas con el MSER son invariantes ante cambios de iluminación lineales debido a que éstas se definen mediante la ordenación de la intensidad de los píxeles. De este modo, al evaluar la diferencia de intensidad y no la intensidad directamente, no importa si se suma o resta una constante al valor de los píxeles.

El algoritmo consiste en ordenar los píxeles de la imagen en orden creciente o decreciente en función de su intensidad. A continuación los píxeles correlativos en intensidad que están unidos espacialmente forman regiones. En este punto se utiliza el umbral para decidir cuáles de estas regiones son estables. Las regiones que alcanzan un mínimo en su relación entre el cambio del tamaño del área de la región respecto al cambio del umbral, son consideradas como las más estables.

Las regiones extraídas tienen la forma mostrada en la figura 3.10. Como se puede observar, éstas no tienen forma elíptica ni rectangular, sino que se adaptan a la forma de la región de interés.

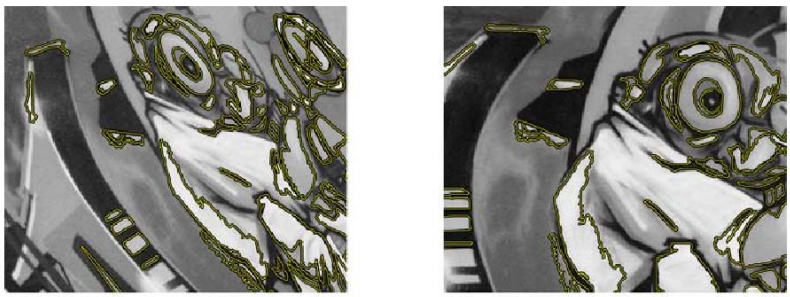


Figura 3.10: Regiones extraídas mediante el detector MSER. Figura extraída de [22].

3.2.6. Detector de regiones destacadas

El detector de regiones destacadas o Salient Region Detector [10] utiliza la función densidad de probabilidad (PDF) sobre la intensidad de los píxeles para encontrar regiones extremas en una imagen. Las regiones candidatas se eligen calculando la entropía de la PDF sobre una familia de elipses que rodean al píxel en cuestión. La entropía extrema en

cada punto es seleccionada junto con la elipse correspondiente. Hay que tener en cuenta que esta operación se realiza sobre distintas escalas y, por lo tanto, las regiones son invariantes a la escala.

Para filtrar las regiones con mejores características entre las candidatas se obtiene el cociente entre la derivada de la PDF y la escala. Las regiones con un mayor cociente son seleccionadas. Así, el número de regiones características puede ser elegido.

3.2.7. Detector de puntos característicos transformados invariantes a la escala (SIFT)

SIFT es un algoritmo compuesto por un detector y un descriptor de puntos característicos. En esta sección se explica la detección, mientras que la descripción de puntos aparece en la sección 4.2.7.

El detector de puntos característicos transformados invariantes a la escala o Scale Invariant Feature Transform (SIFT) [14] es el detector de puntos característicos más utilizado hasta el momento, ya que sus puntos obtienen un alto grado de repetibilidad ante los cambios más comunes de escala, puntos de vista e iluminación. Los puntos extraídos por SIFT pueden utilizarse para el reconocimiento de objetos, como se demuestra en [14].

SIFT presenta invariancia ante cambios de escala y de rotación. Ante cambios de iluminación y de cambios del punto de vista es parcialmente invariable. Los puntos están bien localizados tanto en el dominio espacial como en el frecuencial, lo cual favorece la detección de objetos incluso en el caso que estén parcialmente ocluidos, que aparezcan sobre un fondo muy variable o ante la presencia de ruido.

Debido a la similitud entre SIFT y el método que proponemos para la extracción de puntos característicos (sección 5), los pasos del algoritmo SIFT se muestran con más detalle que los detectores anteriores.

El algoritmo para la extracción de puntos sigue los pasos siguientes:

- Detección de puntos extremos en el espacio de escalas.
- Localización de puntos característicos.
- Asignación de orientación.
- Descriptor del punto característico

En este apartado se describen los dos primeros puntos del algoritmo, los correspondientes a la detección de puntos característicos. Los puntos restantes se muestran en la sección 4.2.7, correspondiente a la descripción de puntos.

Detección de puntos extremos en el espacio de escalas

La primera fase de la detección consiste en buscar puntos característicos que se mantengan ante cambios de escala. SIFT busca extremos en el espacio de escalas en la función de diferencia de Gaussianas (DoG) convolucionada con la imagen de entrada. Esta función se consigue con la resta de dos funciones de escalas consecutivas separadas por un factor multiplicativo k. DoG es una aproximación de la Laplaciana de Gaussiana (LoG) (sección 3.1.4).

Siendo G la función Gaussiana y L la imagen de entrada convolucionada con las distintas G, la función diferencia de Gaussianas D se define como:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma))I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.17}$$

El gráfico de la figura 3.11 muestra cómo se crean las diferentes imágenes para todo el espacio de escalas.

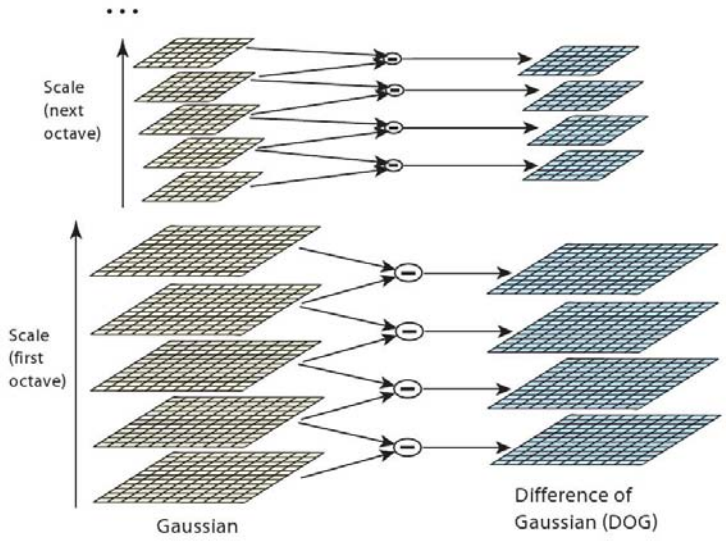


Figura 3.11: Proceso de cómputo de las DoG en SIFT. Figura extraída de [14].

Como se muestra en [14], las diferencias de Gaussianas son creadas a partir de restas de la imagen de entrada convolucionada con diferentes versiones de la función Gaussiana. Cada una de estas imágenes filtradas está separada por un factor k en el espacio de escalas.

Las imágenes que aparecen con el tamaño reducido a la mitad son las correspondientes a la segunda octava, es decir, momento en el que el parámetro σ de la función Gaussiana ha doblado su valor.

Cada una de las octavas se divide en s intervalos, siendo el factor $k = 2^{1/s}$. Las imágenes adyacentes se restan para crear las imágenes de diferencia de Gaussianas mostradas en la parte derecha de la figura 3.11. Una vez se ha procesado toda una octava, la imagen que tiene el doble del valor de σ se reduce a la mitad de su tamaño en ambos ejes seleccionando uno de cada dos píxeles en cada fila y columna.

Una vez está creada la pirámide de matrices, se buscan los extremos máximos y mínimos en $D(x, y, \sigma)$. Para ello se examina cada uno de los píxeles y se compara con sus 8 vecinos, posteriormente se compara con los 9 píxeles correspondientes a la escala superior y los 9 de la inferior. El píxel sólo se considera extremo en caso de que sea mayor o menor en todas las comparaciones y su valor absoluto esté por encima de un umbral determinado. Un ejemplo gráfico se muestra en la imagen 3.12.

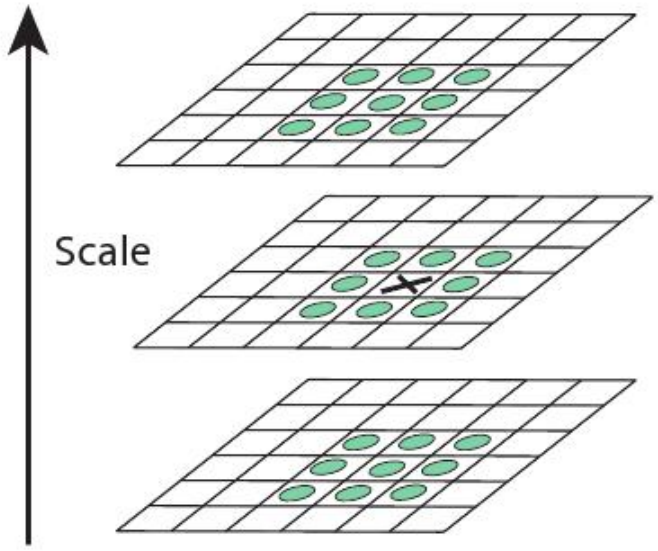


Figura 3.12: Búsqueda de puntos extremos sobre tres imágenes consecutivas en el espacio de escalas. El píxel marcado como 'X' se compara con todas las posiciones marcados con círculos. Figura extraída de [14].

Un aspecto importante al crear el espacio de escalas es la frecuencia de muestreo de escalas. Un mayor número de escalas por cada octava lleva a un mayor número de máximos, pero estos son más inestables. La figura 3.13 muestra la repetibilidad en función del número de escalas por octava. Los cálculos [14] han sido obtenidos aplicando cambios de rotación y escala a imágenes correspondientes a exteriores, caras humanas, fotografías aéreas y fotos industriales. Además se ha añadido un ruido del 1 %.

En la figura 3.13, en línea continua, aparece la repetibilidad obtenida en función del número de escalas. El valor máximo se obtiene con 3 escalas, éste es el valor utilizado

por defecto y mostrado en todos los experimentos por SIFT. La línea discontinua muestra datos correspondientes a la descripción de los puntos y por lo tanto no serán analizados.

Otro aspecto a tener en cuenta es el pre-filtrado que se va a realizar al inicio de cada una de las octavas. La repetibilidad aumenta si filtramos la imagen con una Gaussiana de σ determinada. Se han realizado experimentos [14] para determinar el valor de σ y se ha visto que los resultados mejoran con el incremento de σ. Cuanto más alto es el valor de σ, más costoso a nivel computacional es el filtrado, por este motivo se ha fijado $\sigma = 1{,}6$ y no un valor mayor como valor por defecto. El gráfico con los resultados aparece en la figura 3.14.

Para incrementar el número de puntos estables en un factor de 4, se dobla el tamaño de la imagen inicial. En [14] Lowe afirma que incrementos mayores en el tamaño no implican mejores resultados.

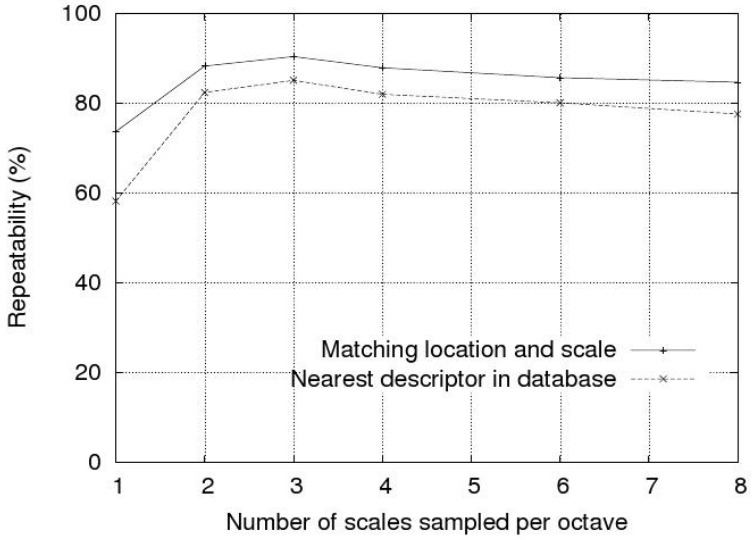


Figura 3.13: Repetibilidad en función del número de escalas por octava. El máximo se alcanza en 3 escalas. Figura extraída de [14].

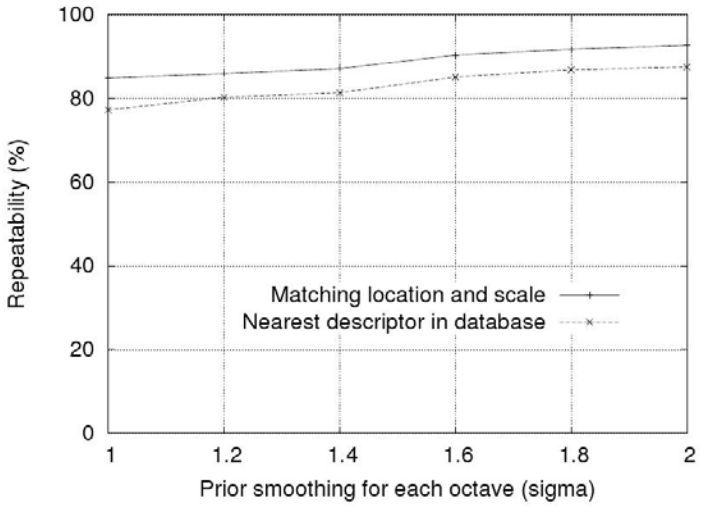


Figura 3.14: Repetibilidad en función del pre-filtrado realizado al inicio de cada octava. Los resultados mejoran con el valor de sigma. Figura extraída de [14].

Localización de puntos característicos

Como se explica en la sección anterior, los máximos se encuentran sobre matrices 3x3 situadas en la escala del punto y las escalas superior e inferior. En este paso se determina la posición en subpíxel y subescala, es decir, con precisión decimal. Para ello se utiliza una función cuadrática sobre la matriz de dimensiones 3x3x3 que rodea a cada punto [15].

Gracias a esta interpolación se consigue una mejora en cuanto a repetibilidad. Dicha mejora se ve incrementada en las octavas superiores del espacio de escalas. Esto es debido a que la imagen ha sido muestreada descartando píxeles de la imagen original y, al realizar una interpolación, el desplazamiento puede llegar a ser de varios píxeles.

Por último SIFT realiza un descarte de los puntos característicos más inestables. La función diferencia de Gaussianas presenta máximos entorno a bordes que aparecen en la escena, sin embargo algunos de estos puntos no son interesantes porque desaparecen ante pequeñas cantidades de ruido debido a que presentan un nivel de contraste bajo. Para eliminarlos se estudian las curvaturas principales alrededor de los puntos.

Estos puntos inestables presentan una alta curvatura principal a través del borde pero, a la vez, un valor bajo en la dirección perpendicular a éste. Las curvaturas principales se pueden obtener mediante una matriz Hessian de dimensiones 2x2 (ecuación 3.13). Para el cálculo de las derivadas se realiza una estimación mediante la diferencia de valores alrededor del punto de interés. Para obtener las curvaturas principales se utilizan los autovalores de la matriz Hessian H, los cuales son proporcionales a dichas curvaturas. El

cálculo directo de los autovalores es un proceso costoso, es por ello por lo que se calcula la suma de autovalores mediante la traza de H y el producto mediante el determinante [16].

En las ecuaciones siguientes α es el autovalor con mayor valor y β se corresponde con el menor.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{3.18}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha * \beta \tag{3.19}$$

Lo que realmente interesa es la relación entre autovalores,

si la relación es $r = \frac{\beta}{\alpha}$, entonces

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha * \beta} = \frac{(r * \beta + \beta)^2}{r * \beta^2} = \frac{(r+1)^2}{r} \tag{3.20}$$

La relación obtiene un mínimo cuando los dos autovalores son iguales y se incrementa con r. Fijando un umbral, fijado como $r = 10$ en [14], se pueden rechazar todos los puntos que tengan una relación mayor a éste y de esta forma los puntos inestables quedan eliminados.

3.2.8. Detector acelerado de puntos característicos robustos (SURF)

El algoritmo acelerado de puntos característicos robustos o Speeded Up Robust Features (SURF) es un detector y descriptor que proclama conseguir los mismos, o incluso mejores, resultados en cuanto a repetibilidad, distinción y robustez, que los métodos existentes en el Estado del Arte actual. La principal ventaja de SURF es el tiempo de ejecución, el cual es menor que el utilizado por los predecesores gracias a las aproximaciones y estructuras utilizadas por éste.

En este apartado se detalla el detector de puntos característicos mientras que el descriptor se muestra en la sección 4.2.10.

SURF utiliza un algoritmo basado en la matriz Hessiana (ver sección 3.1.4) para la detección, el cual busca un equilibrio entre la aproximación de métodos y la conservación de buenos resultados, llegando a crear un algoritmo rápido y con alta repetibilidad.

En cuanto a la invariancia, SURF se centra en escala y rotación, ya que éstas son las deformaciones que ocurren con más frecuencia en las escenas. Como ya se ha mencionado, los métodos para conseguir invariancia en otros aspectos tales como cambios en puntos de vista son muy costosos computacionalmente y, por lo tanto, no se consideran en SURF. Tales efectos están parcialmente cubiertos gracias a la robustez del descriptor.

Detector Hessian rápido

SURF está basado en la utilización de Gaussianas para la creación del espacio de escalas. Concretamente, en una aproximación del determinante de la matriz Hessian. Mediante esta aproximación se consigue un cálculo mucho más rápido del espacio de escalas.

Para entender la aproximación de SURF debe mostrarse previamente qué aspecto tienen las derivadas parciales que componen la matriz Hessian. Estas matrices son las que aparecen en la figura 3.15.

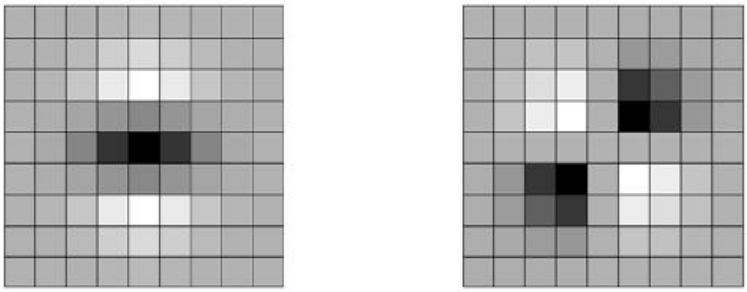


Figura 3.15: Segundas derivadas mostradas en imágenes de 9x9 píxeles. La primera imagen corresponde a la segunda derivada en el eje vertical, mientras que la segunda es la derivada sobre los ejes vertical y el horizontal. Figura extraída de [2].

SURF implementa una versión muy aproximada de las matrices de segundas derivadas parciales de la función Gaussiana. Esta aproximación se hace mediante el uso de ondas de Haar, las cuales se definen como

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2, \\ -1 & 1/2 \leq t < 1, \\ 0 & \text{otro caso.} \end{cases} \tag{3.21}$$

En la figura 3.16 se muestra un ejemplo de dicha función.

Con la mencionada aproximación de las derivadas de segundo orden Gaussianas se

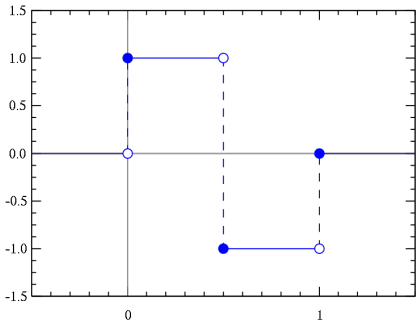


Figura 3.16: Onda de Haar.

obtienen las matrices mostradas en la figura 3.17. Las ondas de Haar son combinaciones de filtros Box, los cuales se corresponden con los espacios rectangulares de la figura 3.17.

La forma utilizada por SURF para la creación del espacio de escalas consiste en aplicar matrices cuadradas de ondas de Haar de tamaños distintos. Este método es equivalente a utilizar una matriz de tamaño único sobre el conjunto de imágenes convolucionadas con Gaussianas con distinto valor de σ.

Con el objetivo de acelerar el proceso, la imagen de entrada se transforma en una imagen integral para lograr reducir el número de accesos a memoria. Una imagen integral contiene, en la posición (x, y), la suma de los valores de todos los píxeles de la imagen original comprendidos dentro del área formada entre el origen y la posición (x, y):

$$II(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{3.22}$$

La imagen integral permite calcular la respuesta a un filtro Box con tan sólo 4 accesos a memoria, independientemente del tamaño del filtro.

Los núcleos mostrados en la imagen 3.17, de tamaño 9x9, son los equivalentes a la derivada Gaussiana de segundo orden con $\sigma = 1{,}2$ y representan la escala más pequeña utilizada en SURF, es decir, la escala con mayor resolución espacial.

A partir de este punto, las aproximaciones de las derivadas de segundo orden realizadas mediante ondas de Haar se denotarán como D_{xx}, D_{yy} y D_{xy}. Según esta nomenclatura, la matriz de aproximación Hessian $aproxH$, es

$$aproxH(x, \sigma) = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \tag{3.23}$$

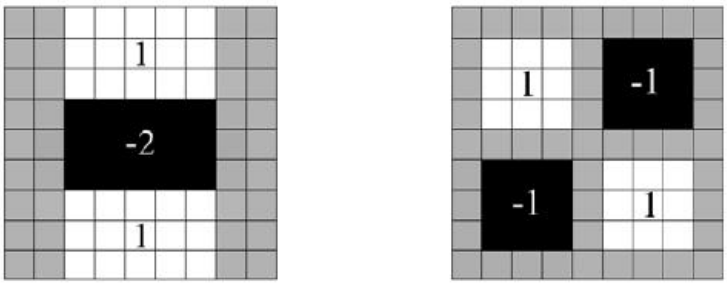


Figura 3.17: Matrices de aproximación de las segundas derivadas Gaussianas utilizando filtros Box. La primera imagen corresponde al filtro en la dirección vertical, la segunda corresponde al filtro en dirección diagonal, es decir, computado vertical y horizontalmente. La numeración indica el valor de los píxeles, siendo 0 para las áreas grises. Figura extraída de [2].

Como muestra la figura 3.17, las ponderaciones aplicadas en las matrices son 0, 1 o -1. El motivo por el que se utilizan dichas cifras es que son las que conllevan un tiempo de cómputo menor; sin embargo, no son las más apropiadas si se desea aproximar al máximo el determinante de la función Hessian de filtros Gaussianos. La solución al problema se logra con una rectificación del determinante Hessian aproximado (ecuación 3.24).

$$\frac{|L_{xy}(1,2)|_F|D_{xx}(9)|_F}{|L_{xx}(1,2)|_F|D_{xy}(9)|_F} = 0{,}912... \simeq 0{,}9 \tag{3.24}$$

siendo $|x|_F$ la norma Frobenius.

De esta forma, la aproximación del determinante de la función Hessian queda expresado como

$$detH_{approx} = D_{xx}D_{yy} - (0{,}9D_{xy})^2 \tag{3.25}$$

La creación del espacio de escalas es realmente rápida ya que cada nivel de escala se calcula de forma independiente utilizando matrices del tamaño correspondiente. La máscara de 9x9 corresponde a $\sigma = 1{,}2$. Para obtener los siguientes niveles el tamaño del filtro aumenta de 6 en 6: 15x15, 21x21, 27x27... Al pasar a la segunda octava, el paso entro filtros se dobla, pasando de 6 a 12 y de 12 a 24. Asimismo, el muestreo sobre los píxeles se dobla al incrementar la octava.

Una vez se ha obtenido el espacio de escalas, se obtienen los puntos máximos de la misma forma como se hace en SIFT (ver sección 3.2.7). Es decir, se fija una ventana de

3x3x3 obteniendo el máximo en los píxeles vecinos del mismo nivel y en los 9 píxeles correspondientes a los niveles anterior y posterior en el espacio de escalas.

Finalmente la interpolación para obtener el sub-píxel y la sub-escala se realizada con el método de Brown [15], el mismo utilizado por SIFT (sección 3.2.7).

3.3. Comparativa de detectores existentes

En esta sección se realiza una comparación valorando la repetibilidad de cada uno de los detectores mostrados en la sección 3.2. Primeramente se comparan los detectores afines mostrados por Mikolajczyk en [22]. Finalmente el detector SURF es evaluado ante los detectores de puntos característicos más relevantes del momento.

Antes de mostrar los resultados de la evaluación se describe la base de datos utilizada en la comparativa y se muestra el método de evaluación utilizado.

3.3.1. Base de datos de Mikolajczyk

La base de datos de Mikolajczyk ha sido ampliamente utilizada en numerosas publicaciones y se ha convertido en una buena referencia para la comparación de detectores y descriptores. Esta base de datos consiste en un conjunto de 8 escenas con 6 imágenes en cada una de ellas, tal como se muestra en la figura 3.18. Cada una de las escenas van a mencionarse a partir de este punto por su nombre: Graffitti, Wall, Boat, Bikes, Trees, Leuven y UBC.

La base de datos está formada por 2 tipos de escenas:

- **Imágenes estructuradas:** Este tipo de imágenes contienen regiones homogéneas con bordes claramente definidos. Son los casos de Graffitti, Boat, Bikes, Leuven y UBC.
- **Imágenes con textura:** Estas imágenes están formadas por repeticiones de texturas de distintas formas. Se puede ver en Wall, Bark y Trees.

El motivo por el que se utilizan dos tipos de escenas distintas es para poder separar el efecto que produce del provocado por el cambio de las condiciones de las imágenes en cada una de las escenas de la base de datos.

Cada una de las escenas sufre uno de los siguientes 5 cambios en las condiciones de las imágenes:

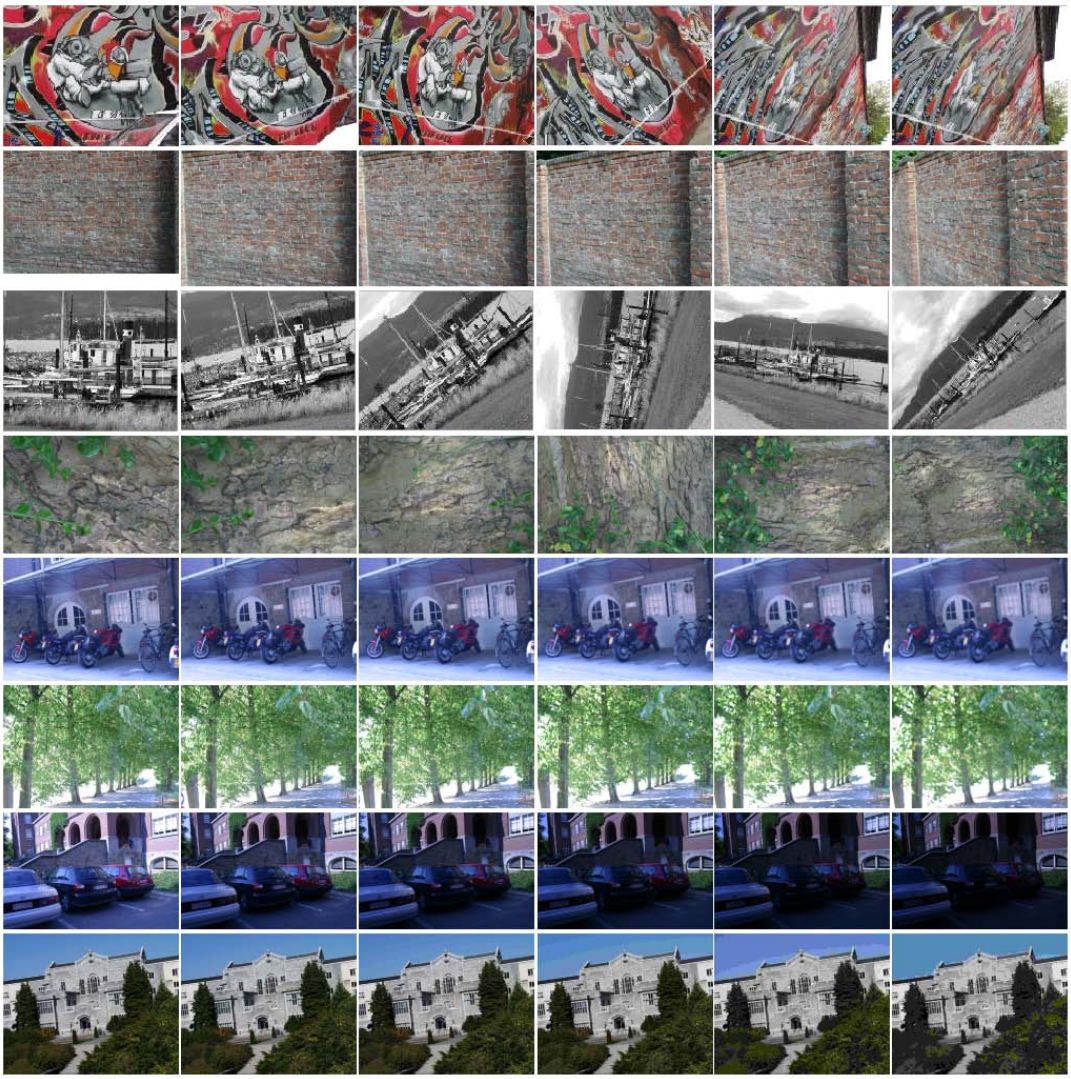


Figura 3.18: Base de datos de Mikolajczyk. De arriba a abajo escenas: Graffitti, Wall, Boat, Bark, Bikes, Trees, Leuven y UBC. Cada escena contiene 6 imágenes, donde de izquierda a derecha se aumenta la transformación aplicada. Figuras extraídas de [22].

- **Punto de vista:** Graffitti y Wall
- **Escala:** Boat y Bark
- **Enfoque:** Bikes y Trees
- **Luz:** Leuven
- **Compresión:** UBC

Las escenas sufren los cambios geométricos o fotométricos de forma gradual desde la imagen 1, tomada como imagen de referencia, hasta la imagen 6, la que presenta mayores cambios.

En el cambio de punto de vista la imagen 1 está tomada frontalmente de forma que la cámara está situada paralelamente al objeto fotografiado. El punto de vista aumenta pasando de los 0^o a los 60^o en la imagen 6. Los cambios de escala se consiguen variando el zoom de la cámara, llegando éste a cambiar en un factor 4. Para mostrar cambios de enfoque se varía el enfoque de la misma cámara. En cuanto a los cambios de luz, éstos se consiguen fijando la apertura de la cámara. Finalmente la compresión de imagen es de tipo JPEG con un factor de calidad que va desde el 40 % hasta el 2 %.

Las características de cada una de las escenas pueden consultarse en el cuadro 3.1.

Escena	**Resolución**	**Tipo**	**Cambio**
Graffitti	800 x 640	Estructural	Punto de vista
Wall	880 x 680	Textura	Punto de vista
Boat	850 x 680	Estructural	Escala
Bark	765 x 512	Textura	Escala
Bikes	1000 x 700	Estructural	Enfoque
Trees	1000 x 700	Textura	Enfoque
Leuven	900 x 600	Estructural	Luz
UBC	800 x 640	Estructural	Compresión

Cuadro 3.1: Características de las imágenes de la base de datos de Mikolajczyk.

En el siguiente apartado se definen las homografías, concepto que debe adquirirse para entender la utilidad de la base de datos utilizada durante el proceso de evaluación.

Homografías

Las imágenes de la base de datos de Mikolajczyk son, o bien escenas planas, o la cámara se encuentra fija en una posición. Dadas estas condiciones, las imágenes de una misma escena pueden relacionarse mediante una homografía.

Una homografía es una matriz que define las transformaciones proyectivas entre dos imágenes. Las transformaciones se hacen sobre un plano, por lo tanto si no se cumplieran las especificaciones definidas anteriormente, no sería posible obtener tal matriz.

Gracias a las homografías, se puede mapear una imagen sobre otra de la misma escena. Este mapeo se utiliza para obtener el "ground truth", es decir, para ver con qué posición en la imagen de test se corresponde cada uno de los píxeles de la imagen de referencia.

La base de datos proporciona las matrices de homografías que relacionan las imágenes de referencia con cada una de las de test. Estas matrices han sido creadas siguiendo 2 pasos.

1. Primero se selecciona manualmente un pequeño número de puntos correspondientes entre las 2 imágenes. Con estos puntos se calcula una homografía aproximada y la imagen de test se deforma de modo que aparezca alineada con la imagen de referencia.
2. El segundo paso consiste en tomar las imágenes de referencia y de test transformada y aplicarles un algoritmo de estimación de homografías para así obtener la homografía detallada.

3.3.2. Método de evaluación de detectores

Para comparar diversos detectores en base a sus resultados en repetibilidad se utiliza el software creado por Mikolajczyk, el cual se ha convertido en el programa estándar a la hora de realizar este tipo de evaluaciones.

El programa obtiene la repetibilidad en imágenes relacionadas por una homografía. Cada uno de los puntos característicos detectados en la imagen de referencia se corresponde con una posición en la imagen de test con la correspondiente escala, definida por la homografía. La repetibilidad cuenta el número de puntos característicos de la imagen de referencia que tienen su correspondiente punto extraído en la imagen de test. Dado que hay puntos que ya no aparecerán en la imagen de test debido a que se salen de la escena tras la transformación, el resultado es dividido por el total de puntos comunes que tienen las dos imágenes.

Para realizar el cálculo, el programa utiliza el error de superposición.

Error de superposición

Los puntos característicos se muestran como elipses situadas en la posición que marca sus coordenadas y con un tamaño definido por su escala. En el caso de utilizar detectores no afines, las elipses se convierten en círculos.

Los puntos encontrados en las imágenes de test son mapeados a la respectiva imagen de referencia mediante la homografía proporcionada.

Para determinar si dos puntos se corresponden, se mide la superposición entre los círculos. El valor determinado por Mikolajczyk es del 40 % de superposición, así, todas las correspondencias con un valor inferior se consideran puntos no repetibles.

Cabe destacar el importante papel que desempeña el tamaño de las regiones definido por su escala. Cuanto mayor es el tamaño, mayor es la superposición entre puntos, tal como se puede observar en la figura 3.19.

La solución que adopta Mikolajczyk es la normalización del tamaño de las regiones para que así, un detector que asigna regiones mayores no obtenga mejores resultados. El tamaño elegido corresponde a círculos de 30 píxeles de radio.

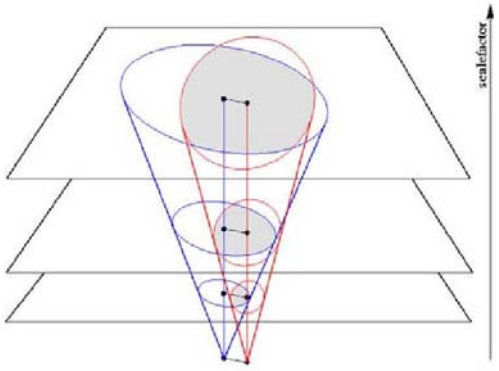


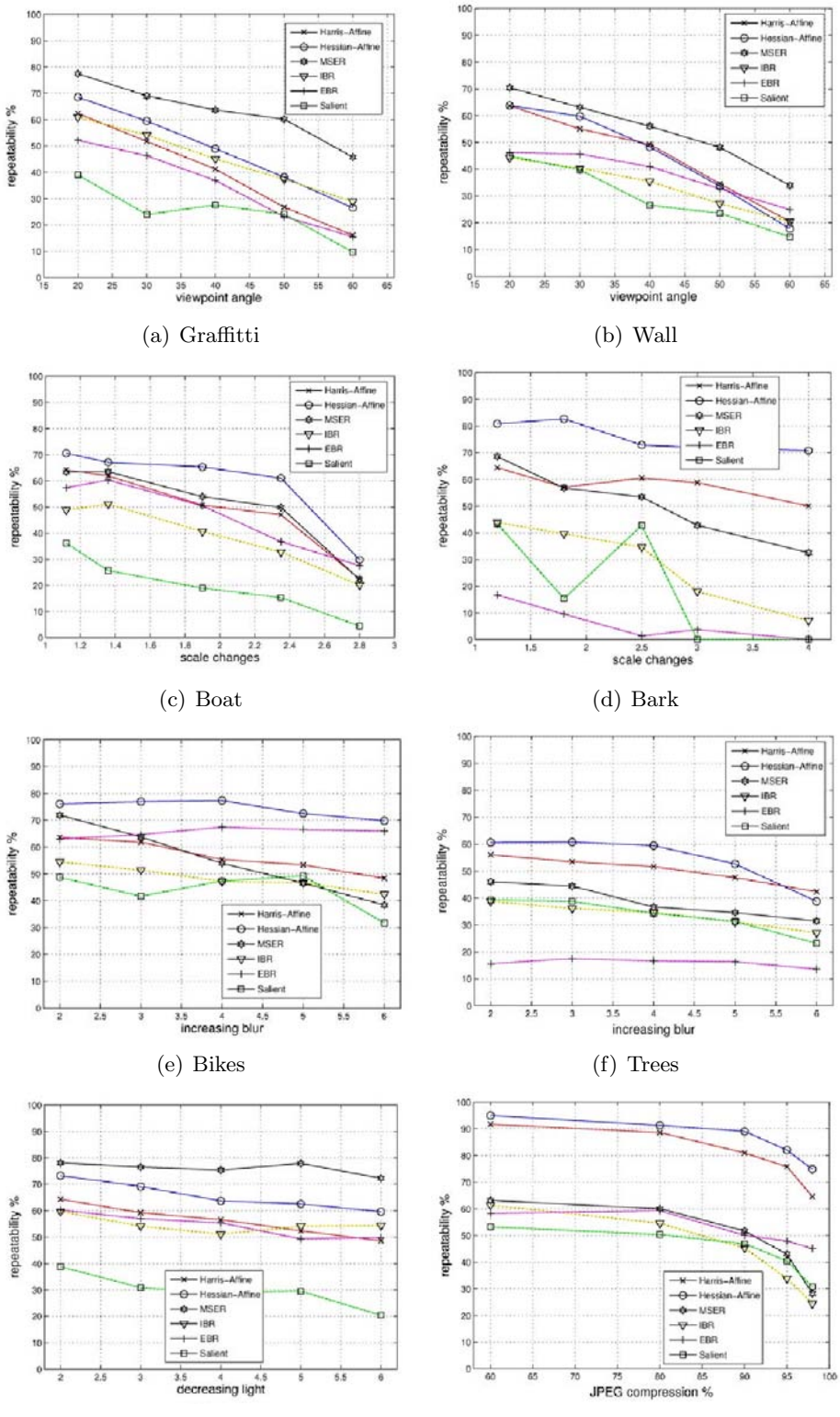
Figura 3.19: Efecto del tamaño de un punto sobre la superposición de regiones. Figura extraída de [22].

3.3.3. Resultados de la evaluación de detectores

En la figura 3.20 extraída de [22] se muestran los gráficos con los resultados en términos de repetibilidad usando los detectores en su versión afín mostrados en la sección 3.2: Harris-Afín, Hessian-Afín, MSER, IBR, EBR y regiones destacadas. Las imágenes utilizadas para dicha comparativa son las correspondientes a la base de datos de Mikolajczyk (sección 3.3.1).

Los resultados muestran Hessian-Afín, Harris-Afín y MSER como los mejores detectores, mientras que el detector Salient es el que presenta peores resultados.

Es interesante observar cómo dos de los tres mejores detectores, según los tests anteriores (Hessian-Afín y Harris-Afín), utilizan un sistema de espacio de escalas basado en la función Gaussiana. Asimismo, el detector Hessian-Afín, el cual destaca por encima de Harris-Afín en cuanto a resultados, utiliza puntos gota como estructuras características.



(a) Graffitti (b) Wall

(c) Boat (d) Bark

(e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 3.20: Comparativa de los detectores Harris-Afín (Harris-Affine), Hessian-Afín (Hessian-Affine), MSER, IBR, EBR y regiones destacadas(Salient). Figura extraída de [22].

A continuación se analizan los resultados obtenidos por SIFT y SURF, dos implementaciones basadas en aproximaciones del espacio de escalas Gaussiano que son más rápidas computacionalmente que los detectores mostrados anteriormente. Los resultados se muestran juntamente con los detectores Harris-Laplace y Hessian-Laplace, correspondientes a las versiones no afín de los detectores Harris-Afín y Hessian-Afín.

En la figura 3.21, obtenida de [2], se muestran los resultados para las escenas Graffitti, Wall, Leuven y Boat, obtenidas de la base de datos de Mikolajczyk. Los resultados mostrados en la leyenda como Fast-Hessian corresponden al detector de SURF, el cual utiliza el detector Hessian rápido. Por otro lado el detector SIFT recibe el nombre DoG, siglas pertenecientes a Difference of Gaussian (diferencia de Gaussianas), debido al sistema en el que se basa SIFT para extraer los puntos característicos (ver sección 3.2.7).

En la escena Wall, un conjunto de 6 imágenes con textura, dónde el ángulo del punto de vista varía de 0º a 60º progresivamente, los mejores resultados se obtienen con SURF, seguido de SIFT. En este caso el peor resultado es el conseguido por el detector Hessian-Laplace. Para un ángulo de 20º, SURF consigue una repetibilidad del 75 %, mientras que para un ángulo de 50º el resultado es del 48 %, aproximadamente. Para 60º la repetibilidad baja drásticamente hasta el 17 %.

La siguiente imagen que sufre un cambio en el punto de vista es Graffitti. En este caso se trata de una imagen estructural. Todos los detectores operan de una forma similar, siendo Hessian-Laplace el que destaca por encima de los otros. SURF obtiene los peores resultados. En esta imagen se obtiene una repetibilidad mucho menor que Wall, obteniendo un valor de 0 a partir de los 50º. Esto es debido a que las estructuras pertenecientes al dibujo del graffitti sufren una gran distorsión provocando que esos puntos no puedan ser detectados como característicos.

En el caso de Leuven cabe destacar que los cuatro detectores responden bien al cambio de iluminación, ya que el nivel de repetibilidad se mantiene con el decrecimiento en el nivel de luz. El detector con mejor comportamiento es SURF, mientras que Harris-Laplace tiene la peor respuesta. Los niveles se mantienen entre el 60 % y el 80 % en todos los detectores.

Finalmente, el último gráfico es el correspondiente a la escena Boat, la cual la componen imágenes estructurales que sufren un cambio de escala (zoom) y de rotación. El detector SIFT es el que funciona mejor ante cambios de escala de hasta 2 niveles. A partir de ahí, SURF y Hessian-Laplace obtienen la mejor respuesta. Una vez más los peores resultados los vemos con Harris-Laplace.

Los cambios en el punto de vista y los de zoom son dos de los más importantes ya que son muy comunes en cualquier escena. Con los resultados vistos en la figura 3.21, podemos descartar Harris-Laplace de la lista de mejores detectores. Es importante observar que, justamente, es el único que no utiliza puntos gota como estructuras características.

Las versiones de SIFT y SURF están al nivel de Hessian-Laplace y, al mismo tiempo

son implementaciones más rápidas computacionalmente que esta última. Por lo tanto la aproximación de espacio de escalas Gaussiano, así como la extracción de puntos gota en distintos niveles de escala es una buena solución a la hora de detectar puntos repetibles y a la vez hacerlo rápidamente. Asimismo, debe valorarse positivamente el hecho de que SURF, un detector que utiliza mayores aproximaciones que le permiten ser el detector más rápido entre los estudiados, consigue resultados comparables a SIFT, el cual precisa de un tiempo de cómputo mayor. Estos resultados nos indican que las mejores estructuras características son los puntos gota. Por otra parte, la aproximación del espacio de escalas Gaussiano es una buena forma de mantener los grados de repetibilidad ahorrando en coste computacional. Por esta razón, las investigaciones llevadas a cabo en la sección 5 se centran en investigar nuevas formas de crear espacios de escala Gaussianos aproximados.

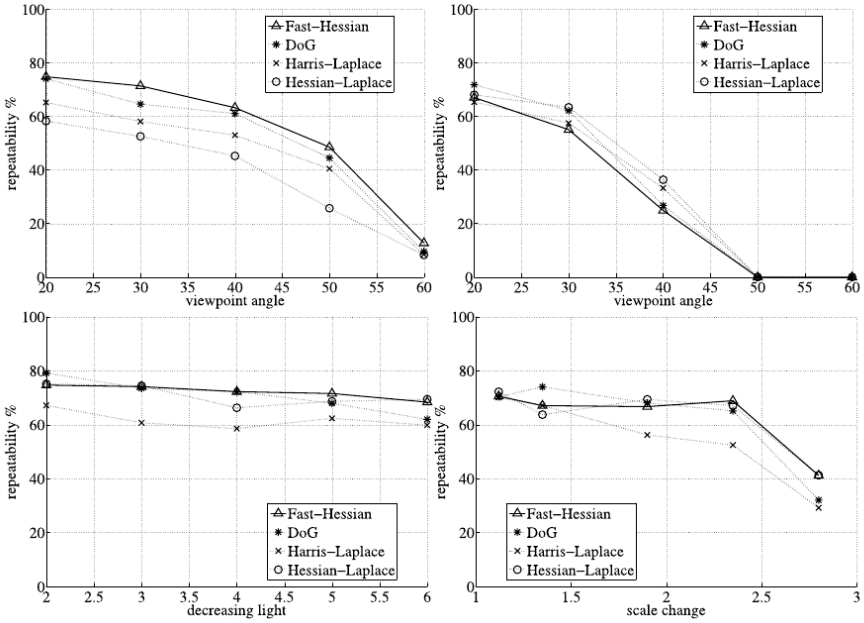


Figura 3.21: Porcentaje de repetibilidad de las imágenes Graffitti y Wall (cambio del punto de vista), Leuven (cambio del nivel de luz) y Boat (zoom y rotación) (de izquierda a derecha y de arriba a abajo). Figura extraída de [2].

Capítulo 4

Descripción de puntos característicos: conceptos básicos, estado del arte y evaluación

Una vez obtenidas las posiciones y la escala de los puntos característicos, se procede a describirlos mediante las características existentes en la región circundante a cada punto. En este capítulo se hace primeramente una breve introducción para pasar a mostrar los conceptos básicos, el estado del arte y por último una comparativa de métodos.

4.1. Conceptos básicos

Los descriptores deben cumplir unas características para poder ser utilizados posteriormente en la búsqueda de contenidos similares en otras imágenes. Las más importantes, detalladas a continuación, son la distintividad, la robustez y la invariancia. Tras las características que se deben cumplir se muestran los métodos para obtener coincidencias entre imágenes a partir de sus descriptores.

4.1.1. Distintividad y robustez

Un concepto muy frecuente cuando se habla de las características de un descriptor es la distintividad. Ésta define la capacidad que tiene un descriptor para distinguir puntos característicos en imágenes. Es importante que un descriptor tenga una alta distintividad para poder realizar buenas correspondencias entre regiones, sin embargo, un exceso de distintividad puede provocar también malos resultados. Esto es debido a que, al ser excesivamente estrictos, un mínimo detalle como un cambio de iluminación o de forma, puede

provocar una nueva descripción y por tanto ser considerado como una región distinta.

Al igual que la distintividad, la robustez es otra característica importante para un descriptor. La robustez es la característica mediante la cual un descriptor es capaz de definir regiones que se encuentran sometidas a un entorno distinto como puede ser la presencia de ruido o condiciones de luz inadecuadas.

4.1.2. Invariancia

Otro concepto muy frecuente en el mundo de los descriptores de puntos característicos es la invariancia ante sucesos como cambios de luz, de escala, de orientación, etc. La invariancia a la orientación se consigue en la mayoría de descriptores calculando previamente la orientación del punto de interés. Si no se obtuviesen los descriptores a partir de una orientación dada, el simple hecho de girar la cámara 90^o provocaría que dos regiones correspondientes al mismo punto de un objeto se describiesen de forma totalmente distinta. Por otra parte, en lo que refiere a la iluminación, existen descriptores capaces de describir regiones que han sufrido cambios lineales en iluminación, sin que el descriptor se vea afectado. Dado que estos cambios son muy frecuentes en cualquier escena de vídeo, se valora muy positivamente la invariancia ante cambios lineales de luz. Finalmente se presta una especial atención a la escala. Un descriptor con invariancia ante cambios de escala es capaz de definir de igual forma un objeto a distintas distancias, permitiendo así realizar correspondencias entre regiones de diferentes tamaños.

4.1.3. Coincidencia

Una coincidencia se obtiene al encontrar dos descriptores similares en dos imágenes pertenecientes a una misma escena, diferenciadas por una transformación geométrica. Para determinar la similitud de los descriptores, se mide la distancia de sus vectores con la distancia Euclidiana. Existen diversas técnicas de comparación entre vectores, mostrándose en este trabajo tres de ellas: las coincidencias basadas en un umbral, la búsqueda del vecino más cercano y el vecino más cercano con relación de distancias. Todas las técnicas de búsqueda de coincidencias comparan cada descriptor de la imagen de referencia con todos los descriptores de la imagen transformada. En la figura 4.1 se muestran las coincidencias encontradas en una pareja de imágenes utilizando vecino más cercano.

Para definir los métodos se utiliza la siguiente nomenclatura: A y B son regiones características pertenecientes a la imagen de referencia y a la transformada, respectivamente. Sus respectivos descriptores se definen como D_A y D_B. En el tercer método se utiliza un tercer descriptor en la imagen transformada, el D_C.

- **Coincidencia basada en umbral**. Este método permite encontrar más de una coincidencia por cada punto característico A de la imagen de referencia. Todos los

Figura 4.1: Dos imágenes mostrando el cartel de un establecimiento. La figura de arriba se toma como referencia mientras que la de abajo ha sufrido una transformación de escala y de punto de vista. Los círculos indican que un punto ha encontrado coincidencia en la otra imagen. El tamaño del círculo indica la escala del punto. Con líneas azules se unen las parejas de puntos coincidentes. Las coincidencias han sido obtenidas utilizando vecino más cercano.

descriptores de la imagen transformada que se encuentren a una distancia menor a un umbral van a ser asignados como coincidentes.

- **Coincidencia basada en el vecino más cercano**. La técnica del vecino más cercano busca, por cada descriptor de la imagen de referencia, el descriptor de la imagen transformada más cercano en cuanto a distancia. Así, D_A y D_B son coincidentes si son los vecinos más cercanos y su distancia se encuentra por debajo de un umbral. Utilizando esta técnica se obtiene una única coincidencia por cada región de la imagen de referencia.
- **Coincidencia basada en el vecino más cercano con relación de distancias entre los dos vecinos más cercanos**. Esta técnica es similar a la anterior. Siendo D_B el descriptor más cercano de D_A y D_C el segundo descriptor más cercano, se mide la relación de distancia de la siguiente forma: $||D_A - D_B||/||D_A - D_C||$. Si el resultado de la división es menor a un umbral, se obtiene una coincidencia.

La búsqueda de coincidencias basada en umbral obtiene un alto número de coincidencias y es la menos costosa computacionalmente entre las tres anteriores. Sin embargo, tiene la desventaja de ser la menos restrictiva, siendo las coincidencias menos fiables. La utilización del vecino más cercano es una forma de restringir el número de coincidencias, ya que cada descriptor en la imagen de referencia puede obtener como máximo una sola coincidencia. Finalmente la relación de distancias es la más fiable de las tres, ya que mide la distancia entre los dos vecinos más cercanos y tan solo acepta la coincidencia en caso de que el segundo vecino se encuentra distanciado del primero. De esta forma se descartan los puntos que son similares a muchos puntos de la imagen transformada, lo cual indica que no es un punto distinguible. La desventaja principal de esta última técnica es que, debido a su alto coste computacional, no puede utilizarse en aplicaciones que tengan restricciones estrictas de tiempo de ejecución.

La longitud del vector que forma un descriptor tiene un impacto directo sobre el tiempo computacional requerido para buscar coincidencias, ya que cada posición del vector debe ser procesada para buscar la distancia Euclidiana entre descriptores. Por este motivo, los descriptores deben conseguir ser distinguibles con un vector de pequeñas dimensiones. Obviamente existe un compromiso entre el buen funcionamiento de un descriptor y la reducción de la dimensionalidad del vector.

4.2. Estado del arte de los descriptores de puntos característicos

4.2.1. Introducción

En esta sección se muestran los descriptores más representativos en el estado del arte actual. Como se detalla a continuación, los descriptores se clasifican en los basados en la

distribución de la región a describir, en su espacio frecuencial, en ser diferenciales o en momentos invariantes generalizados.

- **Basados en distribución (distribution based descriptors)**. Estos descriptores utilizan histogramas para representar distintas características de aparición o forma. El descriptor basado en distribución más simple sería el que crease un histograma con las intensidades de los píxeles.

  Algunos ejemplos son las imágenes Spin creadas por Johnson y Hebert [9], donde se representa un histograma con las posiciones relativas respecto a un punto de interés; el descriptor creado por Rabih y Woodfill [34], el cual crea histogramas de relaciones recíprocas de intensidad de píxeles o el descriptor SIFT[14], que utiliza histogramas de orientación del gradiente.

- **Descriptores espacio-frecuenciales**. Estas técnicas describen el contenido frecuencial de las imágenes. Uno de ellas es la transformada de Gabor [6], utilizada frecuentemente para clasificar texturas.

- **Descriptores diferenciales**. La computación de derivadas de un orden determinado es otra forma de crear descriptores. Las derivadas locales son llamadas comúnmente "local jet" y sus propiedades han sido investigadas por Koenderink [11]. Los filtros dirigibles desarrollados por Freeman y Adelson [5] orientan las derivadas en una dirección concreta. Baumberg [1] por un lado y Schaffalitzky y Zisserman [24] por otro, utilizan filtros complejos derivados.

- **Momentos invariantes generalizados**. Van Gool introdujo en [7] este descriptor que describe la naturaleza multi-espectral de la imagen. Esta técnica se suele utilizar sobre imágenes en color, por lo cual se encuentra lejos del objetivo general de este trabajo.

Los descriptores que consiguen mejores resultados, gracias a que cumplen las características mostradas en la sección 4.1, son los que se basan en la distribución de su entorno mediante la utilización de histogramas. Por esta razón este tipo de descriptores reciben una especial atención en este trabajo. Entre ellos, el más importante es el descriptor SIFT [14], convertido en el descriptor de referencia en numerosos estudios debido a sus buenos resultados y a su satisfactorio uso en aplicaciones como el reconocimiento de objetos.

Otra razón por la que SIFT se muestra de forma más detallada es por la similitud existente entre éste y los métodos que proponemos en la sección 5. De la misma forma ocurre con SURF. Tanto uno como el otro son algoritmos que no sólo describen puntos, sino que previamente los han detectado (secciones 3.2.7 y 3.2.8) formando un algoritmo conjunto de detección y descripción de puntos característicos.

4.2.2. Intensidad en píxeles vecinos

La forma más sencilla de implementar un descriptor consiste acumular la intensidad de los píxeles situados alrededor de un punto característico. La región utilizada será directamente el número de dimensiones del descriptor, es decir, la longitud del vector descriptor. En los experimentos llevados a cabo por Mikolajczyk y Schmid[21], la región utilizada es un cuadrado de 9x9, obteniendo un descriptor de 81 dimensiones.

El problema de este simple descriptor es su varianza respecto a los típicos cambios que suceden en las escenas y su alta dimensionalidad para regiones de cómputo grandes.

4.2.3. Descriptor no paramétrico

Zabih y Woodfill [34] diseñaron un descriptor basado en el orden relativo de las intensidades en lugar de usar los propios valores de intensidad. El proceso consiste en realizar transformaciones no paramétricas sobre una región cuadrada alrededor del punto característico. Las dos transformaciones son las llamadas "rango" y "censo".

Antes de realizar cualquier transformación se compara cada uno de los píxeles de la región con el píxel central correspondiente al punto. En caso de que el píxel en cuestión sea menor que el píxel central, se asigna valor 1 a su posición. Contrariamente se le asigna el valor 0.

El rango es la suma de los valores de la matriz transformada, mientras que censo es un vector con todos los valores de la matriz.

Este descriptor resulta útil a la hora de representar texturas, pero sería necesario un descriptor de muy alta dimensionalidad para conseguir buenos resultados en la correspondencia de puntos característicos debido al tamaño de la matriz censo.

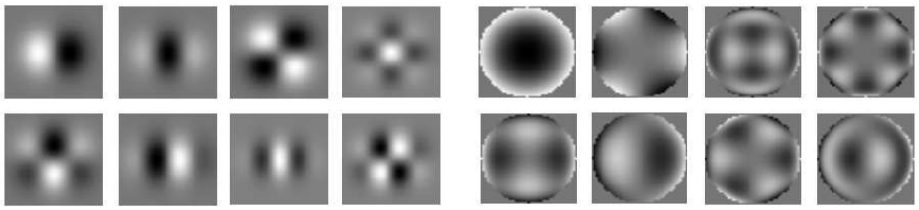
La principal ventaja de este descriptor es su invariancia ante cambios de iluminación y su buen comportamiento al describir regiones donde aparecen bordes. Este buen comportamiento ante los bordes, los cuales causan altas variaciones de intensidad del píxel, es debido al hecho de contar el número de píxeles con un determinado valor, en lugar de acumular los valores reales de éstos.

Debido a la alta dimensionalidad de este tipo de descriptores, los descriptores no paramétricos no van a ser utilizados en los experimentos realizados en este documento.

4.2.4. Filtros diferenciales

Esta clase de filtros se basa en las funciones Gaussianas derivadas en distintos órdenes. Un ejemplo visual de éstos filtros puede verse en la figura 4.2 (a).

Los filtros diferenciales, en su implementación original [5], proponen orientar núcleos en la dirección en la que se haya extraído un punto característico, sin embargo Mikolajczyk y Schmid[21] simplifican ésta operación utilizando núcleos con orientación 0 convolucionados con regiones orientadas correctamente.



(a) (b)

Figura 4.2: Filtros derivativos. En la figura (a) aparecen filtros derivativos Gaussianos de hasta cuarto orden. En la figura (b) se muestran filtros complejos de hasta sexto orden. Figura extraída de [21].

Los filtros complejos [24] se crean siguiendo la ecuación 4.1.

$$K_{m,n}(x,y) = (x+iy)^m (x-iy)^n G(x,y) \tag{4.1}$$

Estos filtros se computan sobre un disco circular alrededor del punto de interés situado dentro de un área de 41x41 muestras. Los distintos filtros se obtienen modificando las variantes m y n (ver figura 4.2 (b)). La respuesta de los filtros con valores $m = n = 0$ es el promedio de la intensidad de las muestras incluidas en la región. La rotación varía la fase pero no la magnitud, por eso se utiliza el módulo de cada respuesta del filtro complejo.

4.2.5. Descriptor de momentos invariantes

Este descriptor se utilizaba originalmente en imágenes en color [7]. El procedimiento para la obtención del descriptor consiste en obtener los gradientes en las direcciones horizontal y vertical de la imagen (x, y), para a continuación aplicar una serie de transformaciones siguiendo la ecuación 4.2.

$$M_{pq}^{a} = \frac{1}{xy} \sum xy x^{p} y^{p} [I_d(x,y)]^{a} \tag{4.2}$$

donde la suma $(p+q)$ especifica el orden mientras que a define el grado.

En los experimentos realizados por Mikolajczyk[21], los filtros complejos utilizados son de segundo orden y grado 2. La región sobre la que se computa el descriptor es de 41x41 píxeles. El descriptor resultante es de 20 dimensiones (10x2 dimensiones, ya que no se utiliza M_{00}^{a} cuando $a \neq 0$).

Debido a que los descriptores de momentos invariantes están diseñados para aplicarse en imágenes en color y a que sus resultados no son tan satisfactorios como los descriptores basados en distribución, no se realiza un estudio más exhaustivo ellos en este trabajo.

4.2.6. Imágenes Spin

Los descriptores creados mediante las llamadas imágenes Spin son de baja dimensionalidad y se basan en la creación de histogramas, por lo tanto se trata de un descriptor basado en distribución.

La región sobre la que se extraen las características es circular. Ésta está compuesta por 5 anillos consecutivos centrados en el punto característico. Sobre cada uno de los anillos se obtiene un histograma de 10 posiciones. Dicho histograma contiene los valores de intensidad de los píxeles de cada anillo cuantificados en 10 posibles valores. De esta forma el descriptor resultante tiene 50 dimensiones (5x10).

Un descriptor de 50 dimensiones es considerado de baja dimensionalidad, lo cual puede provocar que el descriptor no sea suficientemente distinguible y lleva a la obtención de resultados no satisfactorios para el caso de uso deseado en este trabajo. Por otra parte, el hecho de medir directamente los valores de intensidad de los píxeles provoca que no sea invariante ante cambios de iluminación, ni tan si quiera ante cambios lineales.

A pesar de tratarse de un método basado en distribución, este descriptor no consigue obtener el buen funcionamiento alcanzado por los descriptores detallados a continuación.

4.2.7. Descriptor SIFT

SIFT [14] es un detector y descriptor de puntos característicos. En esta sección se muestra el descriptor, mientras que el detector SIFT ha sido detallado en la sección 3.2.7.

SIFT es un descriptor basado en la distribución del gradiente de una región. Antes de describir un punto, SIFT asigna una o varias orientaciones a éste, para posteriormente describirlo en función de la orientación asignada. De esta forma se consigue invariabilidad ante cambios rotacionales de la imagen. El vector descriptor SIFT tiene una longitud de 128 posiciones. SIFT es un descriptor con un alto grado de distinción, siendo a la vez invariable ante cambios como escala o iluminación.

El descriptor SIFT es el que, hasta la fecha, muestra mejores resultados, siendo utilizado en numerosas aplicaciones. Al realizar comparativas entre descriptores suele tomarse como referencia debido a sus buenos resultados; por esta razón en este estudio se seguirá el mismo procedimiento. Sin embargo, SIFT no es un descriptor ideal en muchos aspectos, concretamente SIFT precisa de vectores de 128 posiciones, lo cual se considera como alta dimensionalidad. Este hecho provoca que el proceso de búsqueda de coincidencias requiera un tiempo de cómputo mayor, ya que cada una de las posiciones debe ser procesada (ver sección 4.1.3).

De hecho, el factor tiempo es un problema en el algoritmo completo de SIFT, hecho que imposibilita su utilización en aplicaciones donde las imágenes a procesar son los fotogramas de un vídeo que debe ser procesado en tiempo real.

A continuación se detalla el proceso de asignación de orientación de puntos característicos y posteriormente se muestra el proceso que lleva a cabo SIFT para describir sus puntos.

Asignación de orientación de SIFT

Tras haber realizado la detección de puntos y asignado una escala a cada uno de ellos (sección 3.2.7), debe asignarse una orientación a cada punto característico.

Basándose en las propiedades de la imagen alrededor del punto localizado, se asigna la orientación para que así el descriptor pueda trabajar siempre a partir de ella, consiguiendo de esta forma invariancia ente rotaciones de la imagen.

Las operaciones se realizan sobre el espacio de escalas (sección 3.1.2) extraído durante el proceso de detección de SIFT. Tal como se ha mostrado en la sección 3.2.7, la imagen que desea procesarse se convoluciona con un conjunto de filtros Gaussianos de diferente valor σ para obtener el espacio de escalas. El conjunto de imágenes resultantes forman el espacio de escalas y se denomina imágenes L (ver ecuación 3.2).

Con el fin de lograr invariancia ante cambios de escala, la orientación de un punto característico se realiza utilizando la imagen L del espacio Gaussiano que tiene el valor de escala más próximo al del punto característico.

Una vez obtenida la imagen L, se obtiene la magnitud del gradiente "m" y la orientación

$''\theta''$ sobre la región correspondiente al punto característico.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{4.3}$$

$$\theta(x,y) = tan^{-}1((L(x,y+1) - L(x,y-1))/ + (L(x+1,y) - L(x-1,y))) \tag{4.4}$$

De esta forma se acumulan los valores de magnitud y orientación de los píxeles incluidos en la región. A continuación, los valores de la orientación del gradiente se usan para formar un histograma, mostrando los valores desde 0^o hasta 360^o divididos en 36 segmentos. Los valores se añaden ponderados según la magnitud del gradiente y según una Gaussiana circular con centro en el punto característico, con una escala 1.5 veces mayor que la escala correspondiente al propio punto. Cuando el histograma ha sido creado, se busca cuál es el valor mayor para asignar dicha orientación al punto.

Por cuestiones de eficiencia, los gradientes se calculan en todas las posiciones del conjunto de imágenes L, ya que estos valores se vuelven a utilizar en la descripción de puntos.

SIFT tiene la característica de asignar múltiples orientaciones a los puntos característicos, esto significa que existen puntos con las mismas coordenadas y la misma escala pero con distintas orientaciones asignadas. Esto sucede cuando existen valores en el histograma que tienen un valor mayor al 80 % del obtenido por la máxima orientación. Para conseguir un valor más aproximado de la orientación, se aplica sobre cada una de las orientaciones asignadas una parábola alrededor de los segmentos próximos al máximo.

Sólo un 15 % de los puntos detectados reciben múltiples orientaciones, sin embargo, el hecho de asignar orientaciones múltiples contribuye positivamente en los posteriores procesos de búsqueda de coincidencias, consiguiendo una mayor estabilidad.

Descripción de puntos mediante SIFT

SIFT es un descriptor basado en distribución, es decir, utiliza histogramas para la representación de sus puntos. Se trata a la vez de un descriptor diferencial, ya que utiliza gradientes del espacio de escalas Gaussiano.

SIFT se basa en un estudio realizado por Edelman, Intrator y Poggio en 1997 [25] en el que se demuestra, basándose en un modelo biológico de la visión, que las personas responden ante gradientes de dirección y frecuencia espacial concretas. Dicho estudio revela que el cerebro humano es capaz de identificar objetos y formas aunque la posición del gradiente sufra pequeños cambios.

Utilizando las matrices de gradientes obtenidas durante la asignación de la orientación, se obtienen la dirección y la magnitud del gradiente de las muestras que están alrededor del punto extraído previamente por el detector.

Esta operación se computa para todos los píxeles de la imagen contenidos en una ventana de dimensiones 16x16 píxeles, con centro en el punto característico. La imagen del espacio de escalas L (ecuación 3.2) utilizada es la que tiene un valor de escala más próximo al asociado al punto. Dicha operación se lleva a cabo tras la asignación de orientación del punto. De esta forma los puntos descritos son invariantes a la orientación.

Las orientaciones están ponderadas según la magnitud del gradiente correspondiente. Por otra parte, una Gaussiana de σ igual a la mitad de la amplitud de la ventana de descripción se utiliza para ponderar todas las muestras. La Gaussiana se utiliza para evitar cambios bruscos en posiciones que se encuentran cercanas y para dar más importancia a las muestras que están situadas próximas al punto característico.

Una vez obtenidas las orientaciones con la ponderación correspondiente, se crean los histogramas. Para ello se divide la región de 16x16 muestras en subregiones cuadradas de 4x4 y sobre cada una de ellas se crea un histograma de orientaciones. La orientación contiene valores entre 0^o y 360^o cuantificados en 8 posibles valores. Trabajar sobre regiones de 4x4 muestras permite que pequeños cambios en la posición de un píxel no afecten al descriptor final. En la figura 4.3 puede verse un esquema que muestra el descriptor SIFT.

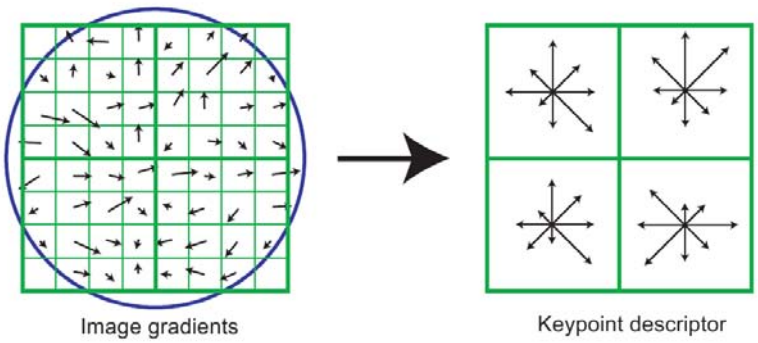


Figura 4.3: Descripción de punto característico con SIFT. Las orientaciones se representan con flechas mientras que la circunferencia representa el filtro Gaussiano. Por cuestiones de simplicidad, el descriptor mostrado es de 2x2, obtenido en 8x8 muestras. En realidad se trata de un descriptor de 4x4 obtenido en 16x16 muestras. Figura extraída de [14].

Como se ha explicado anteriormente, un aspecto importante es evitar cambios bruscos debido a un cambio de posición. Por este motivo se utiliza la función Gaussiana. Otro mecanismo que evita este problema es la interpolación trilinear. Ésta consiste en distribuir el valor de cada gradiente en las posiciones adyacentes del histograma.

El descriptor está formado por un vector que contiene todos los valores de los histo-

gramas, es decir, la longitud de las flechas mostradas en la parte derecha de la figura 4.3. Así pues, la longitud del descriptor para cada punto característico es de 128 elementos, ya que hay 4x4 = 16 histogramas con 8 posiciones cada uno (16 x 8 = 128).

Finalmente, y con el fin de evitar problemas debidos a cambios de iluminación, el vector sufre algunas modificaciones.

La primera consiste en normalizar la longitud del vector a la unidad. De esta forma, si la imagen sufre cambios de contraste, de forma que cada valor de la intensidad es multiplicado por una constante, el descriptor no se verá afectado. Por otro lado, si el cambio de iluminación se produce debido a la suma de una constante en cada muestra, tampoco afectará ya que los gradientes están formados por restas de muestras y las constantes quedan anuladas.

Gracias a la normalización del vector se consigue ser invariante ante cambios de iluminación lineales. Sin embargo, no sucede lo mismo cuando ocurren cambios no lineales debidos a la saturación de la cámara o a una iluminación no uniforme de los objetos presentes en un escenario. Dicho fenómeno provoca grandes cambios en la magnitud del gradiente pero no en la orientación. Por este motivo SIFT establece un valor máximo para la magnitud, fijado en 0.2. Una vez fijado el umbral, se vuelve a normalizar el vector a la unidad. De esta forma se da más importancia a la distribución de las orientaciones en lugar de dar importancia a una gran magnitud. El vector resultante es el descriptor SIFT.

4.2.8. Variaciones de SIFT

En esta sección se presentan 2 versiones modificadas del algoritmo SIFT creado por Lowe [14]: GLOH [21] y PCA-SIFT [21].

GLOH (Gradient Location-Orientation Histogram): La principal variación respecto a SIFT consiste en la forma de la región sobre la que se computa el descriptor. En el caso de SIFT, se establece un espacio cuadrado el cual se divide en 16 partes iguales. En GLOH, la división se hace de forma radial (ver figura 4.4), diferenciando 3 zonas. En dirección angular se crean 8 divisiones obteniendo un total de 17 regiones. Hay que tener en cuenta que el centro no se divide de forma angular.

La creación de gradientes es igual a SIFT excepto en que la orientación se guarda en 16 valores distintos en lugar de los 8 utilizados en SIFT. Por lo tanto, el vector resultante es de longitud 272 (16 x 17). Para reducir este número se utiliza PCA, con una matriz de covariancia obtenida a partir de 47000 imágenes. Los 128 autovectores de mayor valor son los utilizados por el descriptor GLOH.

PCA-SIFT: Este descriptor computa los gradientes en las direcciones horizontal y vertical en una región de tamaño 39x39 alrededor del punto característico. Esto forma un vector de dimensión 3042 que finalmente es reducido a 36 mediante PCA. Las demás

características del descriptor son las mencionadas en SIFT en la sección 4.2.7.

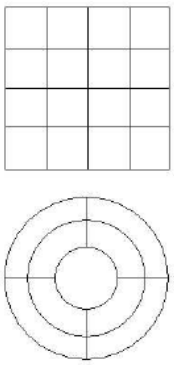


Figura 4.4: En la parte superior se muestra el diseño utilizado por el descriptor SIFT. La parte inferior muestra una división radial en 3 partes, con división angular de 4 sin incluir la muestra central. Es el utilizado por GLOH. Figura extraída de [26].

4.2.9. Descriptor con contexto de forma

El descriptor con contexto de forma es un descriptor basado en distribución, al igual que SIFT, con el que comparte algunas similitudes.

Este descriptor trabaja sobre puntos localizados mediante el detector Canny[4], el cual detecta bordes. Los histogramas del contexto de forma están formados por la orientación de dichos bordes.

La región utilizada para la descripción de puntos es la mostrada en la parte inferior de la figura 4.4, una división radial en 3 partes y otra angular de 4 partes, sin dividir la sección central. De esta forma se obtienen 9 regiones. El número de orientaciones obtenidas por cada sección son 4 (horizontal, vertical y las 2 diagonales), con lo cual se obtiene un descriptor de 36 dimensiones (9x4).Se trata por lo tanto de un descriptor de baja dimensionalidad. Esto conlleva una pérdida en distintividad pero supone un tiempo de búsqueda de coincidencias muy bajo.

En el documento comparativo de descriptores de Mikolajczyk y Schmid[21], se ponderan las orientaciones mediante la magnitud del gradiente para cada posición. Con este cambio afirman conseguir mejores resultados.

4.2.10. Descriptor SURF

SURF [2] es un algoritmo formado por un detector y descriptor de puntos característicos. El detector SURF ha sido previamente explicado en la sección 3.2.8. En esta sección se muestra de forma detallada el funcionamiento del descriptor.

El descriptor SURF está basado en las propiedades del descriptor SIFT, reduciendo las complejidades de éste último mediante mayores aproximaciones. El primer paso consiste en asignar una orientación al punto característico previamente localizado, extrayendo información de una región circular situada alrededor del punto. Contrariamente a lo sucedido en SIFT, SURF tan sólo asigna una única orientación a cada punto. A continuación se construye un área rectangular orientada según la orientación del punto para extraer el descriptor. Existe una versión que no es invariante a rotaciones llamada U-SURF, utilizada en aplicaciones dónde la cámara se mantiene en posición horizontal.

La principal ventaja de SURF sobre SIFT es la reducción del tiempo computacional, hecho que se consigue gracias a las aproximaciones utilizadas. Sin embargo, los resultados de SURF no alcanzan en según qué situaciones los obtenidos en SIFT, tal como se muestra en la sección de evaluación de descriptores 4.3.

Asignación de orientación de SURF

Para poder describir los puntos característicos de forma que sean invariantes a la orientación, primeramente se le asigna una única orientación a cada punto.

El primer paso para la asignación de la orientación es calcular las ondas de Haar (3.2.8) en dirección horizontal y vertical (ver figura 4.5). El resultado de la aplicación de la onda de Haar sobre un punto determinado indica la variación en la dirección de orientación del filtro. De esta forma puede obtenerse la variación existente en las direcciones vertical y horizontal, lo cual permite obtener la orientación del punto característico.

Estos filtros se aplican sobre una región circular situada alrededor del punto característico. El radio de dicho círculo es 6σ. El tamaño de las ondas de Haar depende de la escala del punto en cuestión, concretamente cada lado del núcleo de la función es de 4σ. Asimismo, la frecuencia espacial de muestreo también se establece en función de la escala, siendo igual a σ. A la hora de computar los filtros, tal y como sucede cuando se crea el espacio de escalas (ver sección 3.2.8), se utilizan imágenes integrales para acelerar los tiempos de cómputo. Tan sólo 6 accesos a memoria son necesarios para calcular un filtrado Haar en una dirección concreta, independientemente del tamaño que tenga el núcleo.

El resultado del filtrado con ondas de Haar se filtra de nuevo con una Gaussiana de tamaño 2.5σ centrada en el punto, siendo σ nuevamente la escala del punto.

Una vez obtenidos los resultados del filtrado con la ponderación correspondiente, éstos se dibujan como vectores en un espacio de 2 dimensiones dónde el eje de las abscisas corresponde a la onda de Haar en dirección horizontal mientras que el eje de las ordenadas se asigna al filtrado vertical.

A continuación se usa una ventana deslizante para determinar el ángulo en que las ondas de Haar han obtenido una mayor respuesta. La ventana tiene una amplitud angular de $\pi/3$; ésta recorre todo el espacio desde 0^o hasta 360^o. Los valores del filtrado contenidos para cada dirección son acumulados y el ángulo que acumula un número mayor es establecido como la orientación del punto.

Existe una versión de SURF que no es invariante a la rotación (U-SURF). Ésta ha sido diseñada para aplicaciones dónde no existe rotación de la cámara, como la navegación de robots móviles, dónde la cámara está fija en el cuerpo del robot y tan solo sufre desplazamiento. U-SURF es computacionalmente más rápida y con ella se consiguen mejores resultados para el caso de uso mencionado.

Figura 4.5: Izquierda: ondas de Haar en dirección horizontal y vertical. Derecha: Detalle de la imagen Graffitti mostrando los diferentes tamaños para el descriptor SURF. La orientación se muestra con una línea recta en el interior del cuadrado. Figura extraída de [2].

Descripción de puntos mediante SURF

El objetivo de SURF es realizar un descriptor con un funcionamiento similar a SIFT, ya que éste ha demostrado ser la referencia de los descriptores, tal como muestra Mikolajczyk en [21]. En cuanto a tiempo de computación y posterior búsqueda de coincidencias, SURF es un descriptor mucho más rápido computacionalmente, ya que tiene una dimensionalidad menor y utiliza numerosas aproximaciones.

El descriptor de SURF se calcula sobre un área cuadrada de lado 20σ, con la orientación del punto determinada previamente en la sección 4.2.10. Un ejemplo con dichas áreas se puede apreciar en la figura 4.5.

Esta área se divide en 16 subregiones (4x4) de igual tamaño. Sobre cada una de las subregiones se obtienen los componentes característicos que formarán el descriptor final.

El primer paso para obtener los componentes es calcular las ondas de Haar sobre 25 muestras regularmente espaciadas dentro de cada subregión. De esta forma se consigue, de forma aproximada, el valor del gradiente. Los filtrados Haar en direcciones horizontal y vertical (d_x, d_y) se realizan con un tamaño de filtro de 2σ. Los resultados del filtrado son ponderados con una Gaussiana de 3.3σ centrada en el punto característico.

En cada subregión se suman los valores d_x y d_y, así como las sumas de los valores absolutos $|d_x|$ y $|d_y|$. De esta forma se obtienen 4 componentes por cada subregión, lo cual, para 16 subregiones suma un descriptor de 16x4 = 64 dimensiones.

Los componentes del descriptor son distinguibles ya que los 4 valores definen formas concretas en la imagen. En la figura 4.6 se muestran algunos ejemplos con patrones de imágenes con sus respectivos componentes.

El descriptor SURF tiene invariancia ante cambios lineales en iluminación. Si se añade una constante a la intensidad de los píxeles, el descriptor no sufre modificaciones ya que la onda de Haar es una función diferencial. SURF utiliza, al igual que SIFT, la normalización del vector descriptor de forma que su suma sea igual a la unidad. De esta forma se consigue invariancia cuando la intensidad se multiplica por una constante.

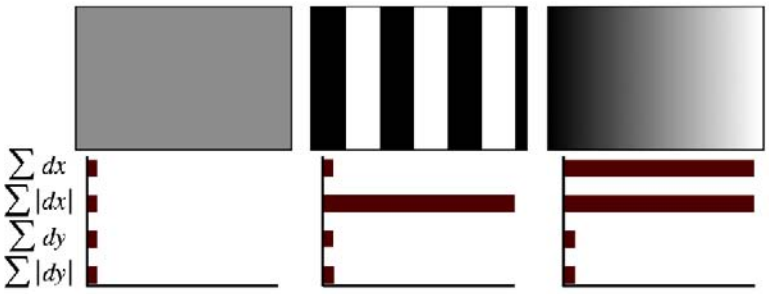


Figura 4.6: Diferentes patrones de imagen con sus correspondientes componentes descriptivos SURF. Izquierda: Una imagen uniforme tiene bajos componentes diferenciales. Centro: cambios frecuenciales en horizontal; sólo $|d_x|$ tiene un valor alto. Derecha: Intensidad gradualmente creciente en 'x'; d_x y $|d_x|$ presentan valores altos. Figura extraída de [2].

Existen 2 variaciones de la versión de SURF estándar, una simplificada: SURF-36 y otra más compleja: SURF-128. Ver cuadro 4.1.

La versión SURF-36 tan sólo se diferencia de la original por el número de subregiones,

Versión	Subregiones	Componentes / Subregión	Dimensiones
SURF	16 (4 x 4)	4	64
SURF-36	9 (3 x 3)	4	36
SURF-128	16 (4 x 4)	8	128

Cuadro 4.1: Versiones del descriptor SURF.

pasando de 16 a 9. Ésta tiene la ventaja de ser más rápida de computar y, por consiguiente, tener un tiempo de búsqueda de coincidencias menor. Sin embargo la distintividad también disminuye.

Por otra parte, SURF-128 es una versión más compleja, con mejores resultados. En lugar de utilizar 4 componentes por cada subregión, se utilizan 8. La diferencia está en que d_x y $|d_x|$ se obtienen para $d_y > 0$ y para $|d_y| < 0$. Lo mismo sucede con las variables intercambiadas. El tiempo de cómputo de esta versión es mayor y la posterior búsqueda de coincidencias, contrariamente que lo sucedido en la versión simplificada, requiere más tiempo.

4.3. Comparativa de descriptores existentes

En este apartado se realiza una comparación de los descriptores más representativos en el estado del arte, todos ellos detallados en la sección 4.2. En primer lugar se describe el criterio de evaluación de descriptores. A continuación se muestran los resultados obtenidos usando la misma base de datos de Mikolajczyk que ha sido utilizada en la evaluación de detectores (sección 3.3).

4.3.1. Método de evaluación de descriptores

El método de evaluación utilizado por Mikolajczyk y Shmid en [21], es el más utilizado actualmente y va a ser usado en todos los experimentos mostrados en este trabajo.

La evaluación utiliza unos gráficos que miden la exhaustividad en función de un parámetro llamado precisión. Dicho gráfico se obtiene para un par de imágenes de una misma escena, relacionadas a partir de una homografía(sección 3.3.1).

La exhaustividad y la precisión se basan en las correspondencias obtenidas mediante una homografía (sección 3.3.1) y en el número de coincidencias (sección 4.1.3) obtenidas en dos imágenes tras extraer los descriptores de puntos característicos.

Partiendo de dos imágenes relacionadas con una homografía, se procede a extraer puntos característicos de cada una de ellas, utilizando un detector y un descriptor concretos.

Una vez extraídos, se define una imagen como la de referencia y la imagen transformada como la de test. Sobre cada punto extraído en la imagen de referencia se establece la localización y escala del punto correspondiente en la imagen de test mediante la homografía. El proceso de búsqueda de coincidencias se encuentra detallado en la sección 4.1.3. Una coincidencia se define a partir de la comparación entre dos descriptores. Si, siguiendo un criterio determinado, los descriptores son similares, entonces existe una coincidencia entre dos puntos característicos.

Combinando los conceptos de correspondencia y coincidencia puede determinarse si una coincidencia es verdadera o falsa. Una coincidencia es cierta en caso de que exista correspondencia entre ambos puntos característicos, en caso contrario la coincidencia es falsa. Las correspondencias indican la posición que ocupa un punto en la imagen transformada, por lo tanto, si dos puntos coincidentes no tienen correspondencia, sabemos que esos puntos pertenecen a objetos distintos en cada imagen.

Exhaustividad y precisión son términos independientes, ya que la exhaustividad depende del número de correspondencias mientras que 1-precisión depende tan sólo del número de coincidencias.

La exhaustividad mide la relación entre el número de coincidencias correctas frente a todas las correspondencias existentes en dos imágenes.

$$exhaustividad = \frac{num.\ coincidencias\ correctas}{num.\ correspondencias} \tag{4.5}$$

Por otro lado, la precisión tan sólo utiliza las coincidencias. Este parámetro se muestra normalmente como 1-precisión, ya que ello hace los gráficos más intuitivos. 1-precisión mide la relación entre el número de coincidencias falsas frente al número total de coincidencias.

$$1\text{-}precision = \frac{num.\ coincidencias\ falsas}{num.\ coincidencias\ correctas + num.\ coincidencias\ falsas} \tag{4.6}$$

Los gráficos muestran la exhaustividad en función de la 1-precisión. Para variar el valor de éste último se varía el umbral que determina si dos descriptores son coincidentes. Al incrementar el umbral se crean más coincidencias, sin embargo éstas son menos fiables ya que la distancia entre los descriptores es mayor y por lo tanto éstos son menos similares. Ésto provoca que la 1-precisión aumente ya que el porcentaje de coincidencias falsas será mayor.

Interpretación de las curvas exhaustividad - 1-precisión: Un descriptor ideal tiene exhaustividad igual a 1 para cualquier valor de precisión. Esto se debe a que todas

las correspondencias existentes entre 2 imágenes tienen una coincidencia correcta independientemente del valor del umbral de coincidencia ya que los descriptores correspondientes a regiones equivalentes son idénticos.

Una curva totalmente horizontal con un valor de exhaustividad $\neq 1$ indica que la exhaustividad ha sido alcanzada con un alto valor de precisión. Este caso puede darse cuando un número concreto de correspondencias tiene los descriptores muy similares pero, por otra parte, el resto de correspondencias tienen descriptores muy distintos que no pueden llegar a ser coincidentes aunque se aumente el umbral.

El comportamiento normal de la curva es un incremento lento a medida que el 1-precisión aumenta, creando normalmente una curva cóncava. Este comportamiento demuestra que el descriptor se ve afectado por las transformaciones sufridas por la escena.

Los descriptores se comparan ante una pareja de imágenes midiendo las curvas de exhaustividad - 1-precisión. El mejor descriptor es el que obtiene curvas con valores de exhaustividad superiores. Si la comparativa entre dos curvas correspondientes a dos descriptores muestra que las curvas son totalmente distintas, significa que los descriptores tienen una distintividad y una robustez distinta para ese tipo de transformación.

La base de datos utilizada para la evaluación de descriptores es la mostrada en la sección 3.3.1. Concretamente, siempre se comparan la primera y la cuarta imagen en todas las escenas.

El software utilizado para realizar las comparativas mediante curvas de exhaustividad - 1-precisión en este documento es el proporcionado por Mikolajczyk. Éste puede obtener coincidencias según los tres criterios descritos en la sección 4.1.3: umbral, vecino más cercano y vecino más cercano con relación de distancias.

4.3.2. Resultados de la evaluación de descriptores

En primer lugar se muestra la comparativa realizada por Mikolajczyk y Schmid en [21]. Este documento merece una especial atención debido a su fiabilidad, igualdad de condiciones en las características de los descriptores y a la utilización de la base de datos mostrada en la sección 3.3.1, la cual afronta numerosas situaciones identificadas independientemente. En esta comparativa no se encuentra SURF, uno de los algoritmos más representativos actualmente, por este motivo se muestra posteriormente la comparativa llevada a cabo en [2], la cual evalúa las distintas versiones de SURF con SIFT y algunos de sus algoritmos similares.

En ambas evaluaciones el método de búsqueda de coincidencias utilizado es el determinado por umbral (ver sección 4.1.3). Según se demuestra en [21], normalmente el método de obtención de coincidencias no es un factor influyente en el orden de clasificación de los descriptores.

En la comparativa de Mikolajczyk se muestran 10 descriptores distintos, todos ellos detallados en la sección 4.2.

1. Intensidad de píxeles vecinos
2. Filtros dirigibles
3. Descriptor diferencial invariable
4. Filtros complejos
5. Momentos invariantes
6. Contexto de forma
7. Imágenes Spin
8. SIFT
9. PCA-SIFT
10. GLOH

Los descriptores anteriores se aplican sobre regiones detectadas mediante Hessian-Afín en el caso de Graffitti, Wall, Bikes, Trees, Leuven y UBC; en las escenas Boat y Bark el detector aplicado es Harris-Afín. El descriptor GLOH se utiliza con ambos detectores, tal como se muestra en la leyenda de los gráficos.

En la figura 4.7 aparecen los resultados en forma de exhaustividad - 1-precisión para la base de datos de Mikolajczyk. En todas las figuras se han utilizado la primera y la cuarta imagen de cada una de las 8 escenas.

En las escenas Graffitti y Wall, donde se produce un cambio del punto de vista juntamente con una rotación, los descriptores que obtienen mejores resultados son los basados en SIFT (SIFT, GLOH y PCA-SIFT). En estos dos gráficos también se utiliza el detector GLOH combinado con un detector no afín, obteniendo peores resultados. Este hecho demuestra que los detectores afín, especializados en cambios de vista de la imagen, funcionan mejor que los no afín es situaciones donde el ángulo de visión varía considerablemente. Hay que tener en cuenta que, ante esta transformación, cualquier descriptor tendrá dificultades para encontrar similitud entre las imágenes. Los resultados son mejores en Wall que en Graffitti debido a que en Wall las estructuras son ladrillos de pequeño tamaño que varían menos su forma ante el cambio de ángulo de la imagen.

En las escenas con cambios de escala, la mejor respuesta está nuevamente en descriptores del tipo SIFT. El contexto de formas, otro descriptor basado en distribución, ofrece otra vez resultados aceptables. En general, los resultados son mejores que en las dos escenas anteriores, lo cual demuestra que los descriptores son más invariantes ante este cambio de escala que ante un cambio del punto de vista de, aproximadamente, 40^{o}.

Las escenas con desenfoque (figuras 4.3.2 (e) y (f)) obtienen bajos resultados ya que este efecto provoca cambios en la intensidad de los píxeles así como deformación de las estructuras existentes de una forma difícil de predecir. La escena con textura, correspondiente a Trees, es la que está más afectada. Los descriptores con mejores resultados son SIFT, PCA-SIFT y GLOH. El contexto de formas, basado en bordes, tiene serias dificultades antes efectos de desenfoque altos ya que los bordes desaparecen. Ante un efecto muy elevado de desenfoque los descriptores no pueden ser distintivos, ya que el área que describen acaba siendo casi idéntica a cualquier otra.

La figura 4.3.2 (g) muestra las curvas para la imagen Leuven, que sufre cambios de iluminación. Nuevamente son los descriptores basados en SIFT los que obtienen mejores resultados, concretamente GLOH, que consigue buenas curvas incluso con puntos detectados con Harris-Afín, los cuales suelen tener una peor respuesta que Hessian-Afín. Por último aparece la imagen UBC, la cual ha sido comprimida mediante JPEG con tan sólo un 5 % de la calidad de la imagen original. SIFT, GLOH y PCA-SIFT son los mejores descriptores para esta escena, juntamente con el contexto de forma. Por lo tanto, nuevamente los descriptores basados en la distribución de características responden mejor que el resto.

Con lo visto en los resultados anteriores, se llega a la conclusión que los descriptores que responden mejor ante los cambios mostrados son los basados en SIFT: SIFT, GLOH y PCA-SIFT. Concretamente GLOH obtiene resultados tan buenos o mejores incluso que el propio SIFT. De los descriptores no basados en SIFT, el mejor de ellos es contexto de forma. El resto, basados en otras técnicas distintas a la distribución de contenidos, tienen resultados peores.

A continuación se muestra la comparativa [2] de 2 versiones de SURF con SIFT y algunos algoritmos similares.

En [2] se comparan dos versiones del descriptor SURF: SURF estándar y SURF-128 (ver cuadro 4.1) con los descriptores que tienen mejores resultados en [21]. El detector utilizado para fijar la posición y la escala de los puntos es el Hessian rápido, detector utilizado en el algoritmo SURF. Por esta razón, los resultados no son en absoluto comparables a los mostrados anteriormente, donde los puntos están extraídos mediante detectores afines.

En la figura 4.8 aparecen las curvas de exhaustividad - 1-precisión para las escenas Wall, Boat, Bikes, Trees, Leuven y UBC. Para todas ellas se han utilizado las imágenes 1 y 4, excepto para Wall, donde se utilizan la 1 y la 5. La técnica utilizada para la búsqueda de coincidencias es la basada únicamente en un umbral (ver sección 4.1.3).

Los resultados muestran como SURF-128 presenta los mejores resultados, seguido de su versión estándar. Posteriormente aparecen SIFT y GLOH con resultados muy similares. En la comparativa el descriptor con resultados más bajos es PCA-SIFT.

Estos resultados demuestran la competitividad de SURF, ya que es un extractor que precisa de un tiempo de cómputo mucho más bajo que sus competidores y, al tener su

descriptor menos dimensiones, el tiempo de búsqueda de coincidencias también es menor. Asimismo, hay que tener en cuenta que el detector utilizado es el Hessian rápido, lo cual puede provocar que los puntos extraídos por SURF sean los más adecuados para ser posteriormente descritos con el descriptor SURF. Tras estudiar los resultados mostrados en esta sección, hemos visto como los mejores descriptores son los que están basados en la distribución de las características; concretamente los que obtienen mejores resultados son SIFT, GLOH y SURF. Por este motivo hemos decidido que el camino a seguir para desarrollar nuevos descriptores es basarse en los principios de distribución de SIFT.

Cabe destacar los buenos resultados obtenidos por el descriptor GLOH, el cual utiliza un área de descripción circular en lugar de la rectangular utilizada por SIFT. Este hecho nos indica que se debe experimentar en la forma en que se diseña el área de extracción de un descriptor.

Finalmente, los resultados sorprendentes de SURF, demuestran que puede obtenerse un buen descriptor utilizando tan sólo 4 estructuras en cada subregión, consiguiendo así un vector de tan sólo 64 dimensiones.

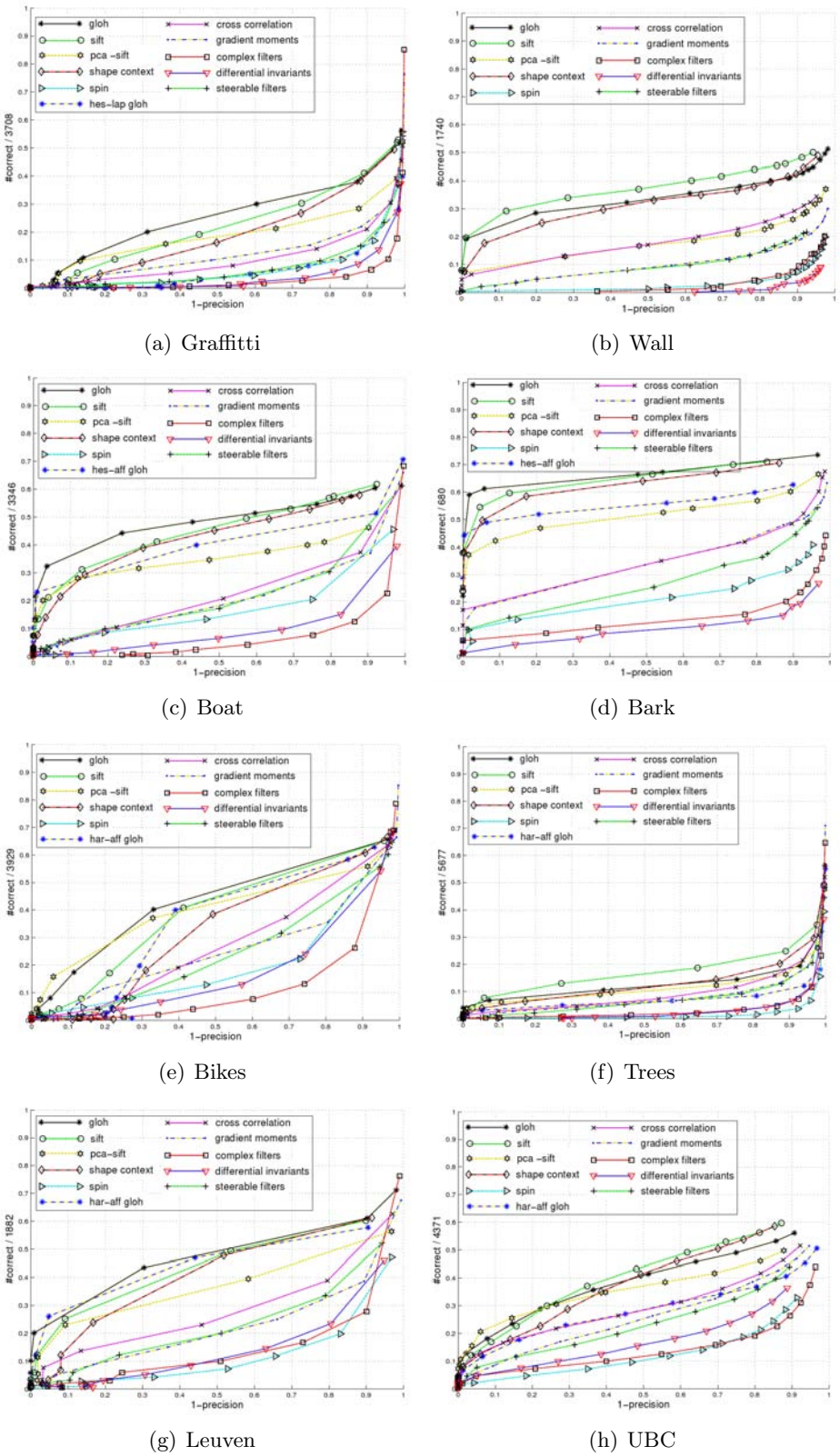


Figura 4.7: Comparativa de descriptores. Figura extraída de [21].

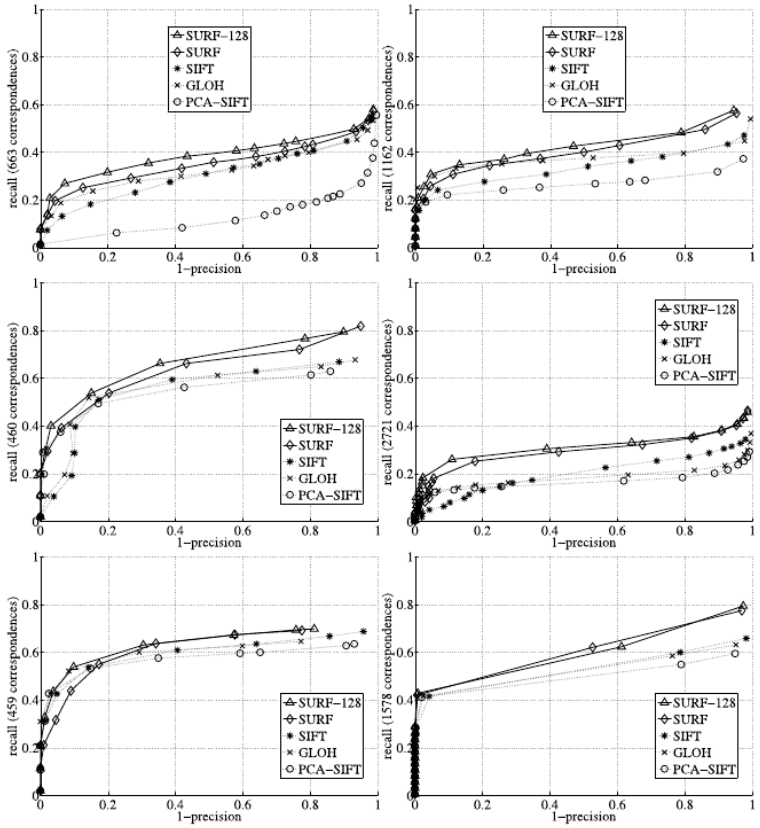


Figura 4.8: Curvas exhaustividad comparando SURF, SURF-128, SIFT, PCA-SIFT y GLOH. Escenas de izquierda derecha y de arriba a abajo: Wall, Boat, Bikes, Trees, Leuven y UBC. Figura extraída de [2].

Capítulo 5

Métodos propuestos: descripción y evaluación

Tras haber mostrado y estudiado el estado del arte, en esta sección proponemos nuevos métodos para la detección y descripción de puntos característicos, basándonos en los resultados obtenidos en las secciones anteriores. Asimismo, proponemos el algoritmo optimizado DART, el cual permite detectar y describir puntos de forma más rápida y con mejores resultados que los algoritmos SIFT y SURF.

5.1. Introducción

Tras estudiar el estado del arte de los detectores y descriptores de puntos característicos, en esta sección presentamos una serie de métodos alternativos para detectar puntos característicos utilizando un espacio de escalas (ver sección 3.1.2) basado en la matriz Hessiana. Seguidamente, implementamos un descriptor de puntos característicos basado en los estudios realizados por Winder y Brown en [26, 32]. A continuación mostramos detalladamente un algoritmo conjunto de detección y descripción de puntos, llamado DART [18], el cual utiliza una metodología eficiente para ahorrar tiempo computacional durante su ejecución. Para finalizar el capítulo, realizamos una serie de experimentos sobre DART y comparamos su funcionamiento con los mejores algoritmos existentes actualmente en este ámbito. En este último apartado se demuestra que DART obtiene mejores resultados que SIFT [14] y SURF [2], a la vez que se reduce el coste computacional requerido por éstos dos últimos.

Los detectores que presentamos a continuación realizan una extracción de puntos gota invariantes al tamaño y a la escala. Estas estructuras han demostrado ser las más apropiadas en la evaluación de detectores de la sección 3.3. Para la creación del espacio de escalas se presentan tres opciones. La primera de ellas utiliza el determinante de la matriz

Hessian, mientras que las dos siguientes son aproximaciones de dicha matriz. La primera aproximación está creada a partir de filtros no uniformes mediante imágenes integrales ponderadas simétricamente, mientras que la segunda utiliza filtros triangulares que aproximan la función Gaussiana.

Durante la evaluación de los descriptores existentes en el estado del arte (ver sección 4.3), se ha visto que los mejores descriptores son los basados en la distribución de características, tales como SIFT, GLOH o SURF. Por este motivo, hemos creado un descriptor basado en los mismos principios. Inspirándonos en el algoritmo SURF, hemos extraído estructuras mucho menos complejas que las utilizadas por SIFT, para así conseguir un descriptor de menores dimensiones.

El algoritmo DART que presentamos detecta los puntos con el algoritmo propuesto, utilizando los filtros triangulares para crear el espacio de escalas. La reutilización del espacio de escalas durante la descripción, así como la optimización del área de extracción, consiguen que DART requiera de tiempos computacionales 4 veces menores que los necesarios en SURF.

Tras detallar el algoritmo, mostramos los experimentos que han sido necesarios para llegar a la configuración más óptima de DART. Finalmente comparamos DART frente a los detectores y descriptores más relevantes existentes actualmente y vemos cómo conseguimos superar el estado del arte en este ámbito.

5.2. Métodos propuestos para la detección de puntos característicos

5.2.1. Introducción

A partir de un análisis sobre los detectores existentes en el estado de arte, se han implementado diversos algoritmos. Tal como se ha visto en la sección 3.3.3, los algoritmos con mejores resultados y más eficientes en cuanto a tiempo de cómputo son SIFT y SURF. En el caso de SURF, una aproximación muy simplificada del espacio de escalas Gaussiano, es dónde se consiguen los tiempos de ejecución más bajos.

La primera de las versiones presentada, muestra un algoritmo el cual hemos tomado como referencia, ya que utiliza el espacio de escalas Gaussiano sin ninguna aproximación. A continuación se presentan 2 aproximaciones del primer método, con una serie de mejoras algorítmicas introducidas en cada una de ellas. Son los detectores basados en los filtros llamados SWII [17] y triangular. El detector utilizado en el algoritmo DART (ver sección 5.4), utiliza el filtro triangular para detectar los puntos característicos.

Todos los algoritmos implementados utilizan el determinante de la matriz Hessian,

utilizada anteriormente en los detectores Hessian-Laplace, Hessian-Afín y SURF. De esta forma, las estructuras extraídas son puntos gota, los cuales han demostrado obtener mejores resultados.

Asimismo, se utilizan métodos de obtención de escala característica para cada uno de los puntos detectados siguiendo los principios mostrados por Lindeberg en [13].

5.2.2. Descripción general del algoritmo

El algoritmo detector de regiones características recibe como parámetro de entrada una imagen y obtiene un fichero con los puntos extraídos, indicando las coordenadas y la escala para cada uno de ellos.

Los pasos que sigue el algoritmo son los siguentes:

1. Pre-procesado.
2. Creación de espacio de escalas.
3. Búsqueda de extremos.

Pre-procesado

En algunos casos puede resultar interesante procesar la imagen de entrada antes de ejecutar el algoritmo propiamente dicho.

En este caso el pre-procesado consiste en filtrar la imagen con un filtro paso-bajo Gaussiano para así eliminar el ruido y las estructuras de más alta frecuencia, las cuales puede que no sean de interés.

Esta fase del algoritmo es opcional y tan sólo se utiliza en algunas de las versiones de los diversos detectores implementados.

Creación del espacio de escalas

La imagen que se desea procesar se filtra con un conjunto de filtros Gaussianos (ver sección 3.1.3) o aproximaciones de éstos. En los algoritmos creados se varía el parámetro σ, correspondiente a la desviación estándar de la función Gaussiana.

$$\sigma = 2^{(n/3)}; n : [0..N-1] \tag{5.1}$$

donde n es el nivel de escala utilizado en el algoritmo, mientras que N es el número de escalas utilizadas.

Así, según la fórmula anterior, al primer nivel le corresponde una $\sigma = 1$. La correspondencia de valores entre σ y n puede verse en el cuadro 5.1.

n	**Sigma**
0	1
1	1.25992105
2	1.587401052
3	2
4	2.5198421
5	3.174802104
6	4
7	5.0396842
8	6.349604208
9	8
10	10.079368399
11	12.699208416
12	16
13	20.158736798
14	25.398416831

Cuadro 5.1: Correspondencias entre n y σ. n se corresponde con el nivel de escala mientras que σ es la escala del punto.

Normalmente, el conjunto de imágenes sobre las que se trabaja no son directamente las imagen filtrada con Gaussianas de diferente σ, sino que se utilizan matrices creadas a partir de dichas imágenes. Las más comunes son la Laplaciana de la Gaussiana y el determinante Hessiano (ver sección 3.1.4). En los algoritmos que mostramos en este trabajo siempre se utiliza el determinante Hessiano.

Búsqueda de extremos

Una vez creado el espacio de escalas, el siguiente paso consiste en obtener los niveles de intensidad máximos en el conjunto de matrices, para así determinar la posición y la escala de los puntos gota. El nivel de escala sobre el que se extrae el máximo indica el tamaño del punto. Como se muestra en la sección 3.1.4, los extremos mínimos en el determinante de Hessian no se corresponden con puntos gota.

La búsqueda de extremos se basa en los estudios llevados a cabo por Brown en [15], aplicados anteriormente en los algoritmos SIFT y SURF. Este método, detallado en la sección 3.2.7,busca los valores máximos en cada uno de los píxeles del espacio de escalas. El valor de cada posición se compara con sus vecinos del mismo nivel de escala y con

los vecinos correspondientes en las escalas superior e inferior (figura 3.12). El punto se considera máximo en caso de superar un valor umbral determinado por el algoritmo y siendo a la vez superior a todos sus vecinos. El tamaño de las ventanas aplicadas es un parámetro sobre el que hemos experimentado en cada uno de los algoritmos. Una opción sobre la que se ha experimentado y aplicado en algunos de los algoritmos es la de utilizar una ventana de tamaño variable en función del nivel de escala.

Una vez establecidos los máximos en coordenadas y niveles de escala se obtiene la sub-posición y la sub-escala para tener una mayor precisión, tal como se hace en SIFT y en SURF. Este procedimiento se encuentra detallado en la sección 3.2.7.

En los algoritmos que proponemos a continuación, a diferencia de SIFT y SURF, no se reduce el tamaño de la imagen cada vez que se incrementa una octava. De este modo pueden obtenerse extremos en todas las escalas, excepto en la primera y la última, ya que no tienen un nivel inferior y superior, respectivamente, al que compararse. Reduciendo el tamaño de la imagen, tan sólo pueden obtenerse máximos en las escalas que tienen niveles superiores e inferiores en la misma octava, ya que no pueden compararse con imágenes de distinto tamaño.

5.2.3. Determinante de la matriz Hessian sin aproximaciones

Este método crea un espacio de escalas con filtros Gaussianos y a continuación computa el determinante de la matriz Hessiana. De esta forma se obtiene un método sin aproximaciones Gaussianas que obtiene buenos resultados en repetibilidad, el cual vamos a utilizar como referencia respecto a las aproximaciones realizadas en los próximos algoritmos. Sin embargo, este algoritmo requiere altos tiempos de computación, así como un elevado número de accesos a memoria, por lo cual es menos interesante su aplicación.

El primer paso del algoritmo es un procesamiento de la imagen que consiste en aplicar un filtro paso-bajos con un filtro Gaussiano de tamaño 5x5. Hemos realizado diversos experimentos dónde el tamaño de dicho filtro ha sido variado e incluso eliminado. El tamaño seleccionado ha proporcionado los mejores resultados en repetibilidad.

A continuación se crea un espacio de escalas de 15 intervalos. Sobre las matrices filtradas con el filtro Gaussiano se aplica el determinante de la matriz Hessian. Al crear las matrices de determinantes deberá tenerse en cuenta la siguiente norma para que los valores resultantes sean similares entre escalas, para así poder ser comparados en la búsqueda de extremos.

$$DetHnorm = \sigma^4(D_{xx}D_{yy} - D_{xy}^2) \tag{5.2}$$

La normalización consiste en añadir el factor σ^4, de esta forma se evita que los valores

del determinante de la Gaussiana disminuyan en función de la escala y su valor medio se mantenga constante en todos los niveles.

Sobre estas matrices se realizan los procesos de selección de máximos detallado anteriormente en 5.2.2.

5.2.4. Filtrado no uniforme mediante imágenes integrales ponderadas simétricamente (SWII)

Este método utiliza la combinación de Symmetric Weighted Integral Images (SWII) [17] para aproximar los filtros de las derivadas de segundo orden Gaussiano de forma rápida. Como ya sabemos, estas derivadas son necesarias para obtener la matriz Hessiana.

La aproximación de las derivadas parciales se crea mediante la combinación de imágenes integrales ponderadas. Una imagen integral contiene en la posición (x, y) la suma de los valores de todos los píxeles de la imagen original comprendidos dentro del área formada entre el origen y la posición (x, y) (ver sección 3.2.8).

Los 5 filtros SWII para un espacio de 2 dimensiones son los mostrados en las ecuaciones siguientes, donde la función $f(x, y) \rightarrow \Re$ con $(x, y) \in [1, W]x[1, H]$ es la imagen de entrada y H, W, son las dimensiones vertical y horizontal de ésta.

$$S_0(x, y) = II(x, y) = \sum_{i \leq x, j \leq y} f(i, j) \tag{5.3}$$

$$S_x(x, y) = \sum_{i \leq x, j \leq y} (i - x) f(i, j) \tag{5.4}$$

$$S_{-x}(x, y) = \sum_{i \leq x, j \leq y} (x - i + 1) f(i, j) \tag{5.5}$$

$$S_y(x, y) = \sum_{i \leq x, j \leq y} (j - y) f(i, j) \tag{5.6}$$

$$S_{-y}(x, y) = \sum_{i \leq x, j \leq y} (y - j + 1) f(i, j). \tag{5.7}$$

La ecuación 5.3 es la correspondiente a una imagen integral corriente. La función S_x crea una imagen integral ponderada de forma que se forme una pendiente positiva en la dirección horizontal de la imagen, mientras que la ecuación S_{-x} crea la pendiente negativa en la misma dirección. Finalmente las funciones S_y y S_{-y} realizan la misma función en la dirección vertical de la imagen. A partir de combinaciones de estas ecuaciones podemos crear filtros no uniformes de distintas formas.

En este caso en concreto los filtros que se desean crear son los derivados de segundo grado del filtro Gaussiano. Sus aproximaciones se denominan como D_{xx}, D_{yy}, D_{xy}, correspondiendo a las derivadas en la dirección horizontal, vertical y cruzada, respectivamente.

La aproximación la creamos mediante la combinación de algunos de los filtros SWII, de forma que las pendientes estén parcialmente solapadas. Un ejemplo gráfico, para el caso D_{xx}, se puede apreciar en la figura 5.1.

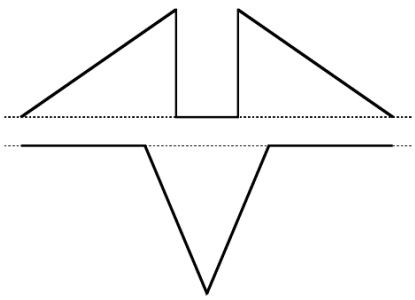
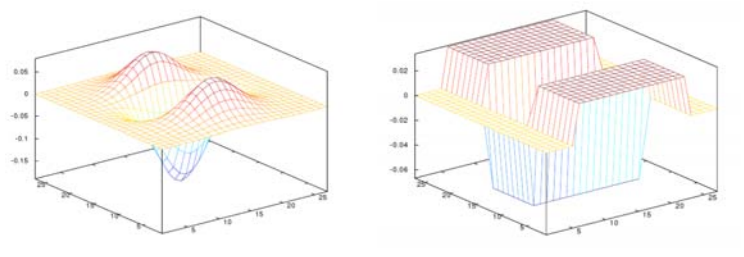


Figura 5.1: Funciones creadas mediante la combinación de filtros SWII. Con la suma de las dos se obtiene la forma aproximada de la segunda derivada Gaussiana en dirección horizontal (D_{xx}).

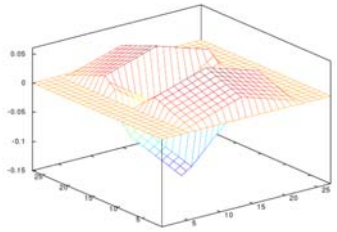
En la figura 5.2 puede verse la segunda derivada Gaussiana y las aproximaciones obtenidas usando imágenes integrales y SWII, respectivamente. Para la obtención del filtro mediante SWII se ha llegado a un compromiso entre fidelidad a la forma no aproximada y número de accesos a memoria necesarios. Por ello cabe destacar que la aproximación podría ser más precisa sobreponiendo más pendientes con distintas ponderaciones. Como puede observarse, la aproximación lograda a partir de SWII obtiene una forma más aproximada a la segunda derivada Gaussiana que la obtenida con imágenes integrales, la cual se utiliza en SURF.

En las próximas secciones se utilizará el término "filtros SWII" al referirse a la combinación de filtros SWII que proporcionan las derivadas de segundo orden descritas anteriormente.

En la sección 5.5 se muestran los resultados obtenidos mediante la utilización del método de detección que hemos propuesto en 5.2.2 utilizando filtros SWII para la creación del espacio de escalas.



(a) Segunda derivada Gaussiana. (b) Aproximacion con imagenes integrales.



(c) Aproximacion con SWII.

Figura 5.2: Segunda derivada de la Gaussiana en dirección horizontal (D_{xx}) y las aproximaciones obtenidas a partir de imágenes integrales y de SWII, respectivamente.

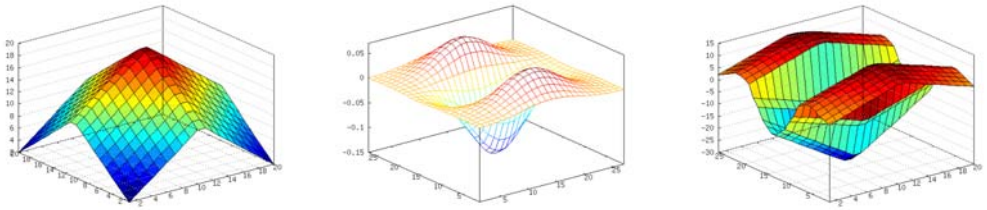
5.2.5. Filtro triangular

El filtro triangular que proponemos utiliza aproximaciones de las segundas derivadas de la función Gaussiana para calcular el determinante de la matriz Hessian de forma rápida y eficiente. Este procedimiento se realiza para todo el espacio de escalas.

Inicialmente, para cada una de las escalas k, la imagen a procesar se transforma con un filtro de 2 dimensiones de forma triangular (ver figura 5.3 a)), obteniendo el espacio de escalas $L(k, i, j)$. Donde i y j representan las coordinadas de la imagen.

El filtrado en múltiples escalas es un procedimiento que consume un tiempo considerable, por ese motivo se propone una forma eficiente de calcular las matrices triangulares. Tal como muestra Heckbert en [8], los filtros Gaussianos pueden aproximarse convolucionando de forma iterativa filtros Box. Heckbert también identificó una relación entre las n veces que se convolucionan los filtros Box y el acceso a diversas muestras de la enésima integral de una función. Según Heckbert, filtrar mediante un filtro triangular una señal de 1 dimensión (equivalente a convolucionar 2 veces un filtro Box), requiere tan sólo acceder a 3 muestras de la señal y a continuación integrar 2 veces. Para una sola escala k, $L(k, i, j)$ puede obtenerse con tan sólo 2 pasos sobre la imagen, una horizontalmente y la otra de forma vertical.

A continuación, las derivadas de segundo orden se computan accediendo a $L(k, i, j)$ en



(a) Aproximación del filtro Gaussiano mediante el núcleo triangular

(b) Segunda derivada Gaussiana.

(c) Segunda derivada Gaussiana aproximada mediante filtros triangulares.

Figura 5.3: Aproximaciones de la función Gaussiana y su segunda derivada mediante filtros triangulares.

diferentes puntos. En la figura 5.3 (c) puede observarse el resultado de la aproximación.

Las ecuaciones siguientes muestran los accesos al espacio de escalas necesarios para obtener las aproximaciones de las segundas derivadas Gaussianas en un punto (i, j) de la imagen, con escala k.

$$\partial^k_{xx} = L(k, i - d_1, j) - 2 \cdot L(k, i, j) + L(k, i + d_1, j) \tag{5.8}$$

$$\partial^k_{yy} = L(k, i, j - d_1) - 2 \cdot L(k, i, j) + L(k, i, j + d_1) \tag{5.9}$$

$$\partial^k_{xy} = L(k, i - d_2, j - d_2) - L(k, i + d_2, j - d_2) - \tag{5.10}$$

$$L(k, i - d_2, j + d_2) + L(k, i + d_2, j + d_2) \tag{5.11}$$

donde d_1 y d_2 son elegidos experimentalmente siendo proporcionales a la σ de la segunda derivada Gaussiana aproximada. Las aproximaciones anteriores no son equivalentes a utilizar con un filtro triangular y a continuación convolucionar con un filtro derivativo de segundo orden [1, -2, 1], ya que eso provocaría efectos no deseados.

Como vemos en las ecuaciones anteriores, la computación de las derivadas en un punto se consigue accediendo tan sólo 9 veces a L. El número de accesos en SURF [2] es mucho mayor, ya que la aproximación del determinante de Hessian se consigue accediendo 32 veces a la imagen integral.

La forma cómo se calcula el filtro triangular y las posteriores derivadas parciales permiten obtener el conjunto de matrices con pocos accesos a memoria, lo cual reduce el tiempo de cómputo. Por otra parte, las matrices filtradas triangularmente pueden aprovecharse en el proceso de descripción de puntos característicos, tal como se muestra en la sección 5.3.3.

5.3. Descripción de puntos característicos propuesto

5.3.1. Introducción

Tras estudiar el estado del arte de descriptores de puntos característicos y analizar los resultados obtenidos en las sección 4.3.2, hemos diseñado un descriptor basado en la distribución de características que obtiene mejores resultados en las curvas de exhaustividad - 1-precisión (ver sección 4.3.1) que los obtenidos por SIFT, el descriptor con mejores resultados hasta el momento. La reutilización del los filtrados utilizados durante la detección de puntos, así como la eficiente implementación del diseño del descriptor, permiten ejecutar el algoritmo en tiempos inferiores a los logrados por SURF, el más rápido actualmente.

El descriptor que proponemos está basado en SIFT y en los estudios llevados a cabo por Winder y Brown en [26], en los cuales se proponen nuevos diseños en el área de extracción de componentes, así como estructuras características más simples que las utilizadas por SIFT.

Previamente a la descripción de puntos propiamente dicha, se determina la orientación del punto característico para que así el descriptor sea invariante a la rotación. Este método es similar a los utilizados por SIFT y SURF.

5.3.2. Asignación de orientación del algoritmo propuesto

La descripción de un punto característico, como ya ha sido explicado anteriormente en la sección 4, se hace en relación a la orientación asignada al punto. Para el cálculo de la orientación se utilizan características de los métodos utilizados en SIFT y en SURF.

Primeramente se asigna un área circular alrededor del punto de interés de tamaño proporcional a la escala del punto, tal como sucede en SURF. Sobre esta área se calculan los gradientes. Para ello se utiliza la imagen del espacio de escalas (ver sección 3.1.2) con la escala más próxima a la del punto de interés. Los gradientes se calculan sobre un número de muestras contenidas dentro del área circular, muestreadas según σ.

A continuación, tal como sucede en SIFT (ver sección 4.2.7), se crea un histograma de 36 posiciones con la orientación de cada una de las muestras. Dichos valores están ponderados por el módulo del gradiente y por una Gaussiana de tamaño 2.5σ, siendo σ la escala del punto. Una vez creado el histograma se aplica un proceso de suavizado del mismo para evitar la aparición de máximos no significativos. El posterior ajuste de segundo grado realizado por SIFT no se utiliza en este caso ya que hemos visto que utiliza un tiempo de cómputo excesivo mientras que no aparecen mejoras significativas.

Finalmente se aplica el mismo sistema de asignación múltiple de orientación que el

utilizado en SIFT.

5.3.3. Descriptor de puntos característicos propuesto

A continuación se explica el descriptor utilizado para describir los puntos extraídos mediante los detectores mostrados en la sección 5.2. Dicho descriptor utiliza las imágenes filtradas $L(k)$ del espacio de escalas, creadas durante el proceso de detección.

Los componentes del descriptor son similares a los que se obtienen en SURF. Éstos han sido inspirados en el trabajo de Winder y Brown [26]. Asimismo, el diseño del área de extracción utilizado ha sido escogido de la publicación de los mismos autores [32].

Diseño del área de extracción de componentes

En la comparativa de descriptores realizada por Mikolajczyk [21], aparecen diversos descriptores que no son más que modificaciones de SIFT. Una de ellas, GLOH, obtiene resultados incluso mejores que el primero. En dicha versión, el área de extracción de componentes utilizado es circular, contrariamente a la sucedido en SIFT, donde se utiliza una malla rectangular de 4x4.

En [26] se realiza una comparativa con distintos diseños para el área de extracción, donde aparecen áreas rectangulares y circulares ponderadas bilinearmente o mediante Gaussianas. Estos diseños pueden apreciarse en la figura 5.4. El diseño llamado S4, formado por un área circular con 17 puntos, es uno de los que obtiene mejores resultados en los experimentos llevados a cabo por Winder y Brown.

Basándonos en los buenos resultados obtenidos en el descriptor GLOH y en los estudios llevados a cabo en [26], hemos decidido utilizar la configuración S4 citada anteriormente. En ella se utiliza el punto central y dos anillos concéntricos con 8 componentes en cada uno, lo cual proporciona 17 segmentos. Tras realizar varios experimentos (ver sección 5.4.2) modificando el número de anillos y de segmentos hemos decidido utilizar la misma configuración.

Una vez mostrado el diseño del área de extracción, mostramos a continuación el proceso que se lleva a cabo para extraer los componentes que formarán el vector del descriptor resultante.

Los componentes se extraen sobre cada uno de los 17 segmentos del descriptor. La separación entre anillos y entre el punto central y el primer anillo se fija en función de la escala del punto característico en cuestión, siendo 4σ la distancia que nos proporciona mejores resultados.

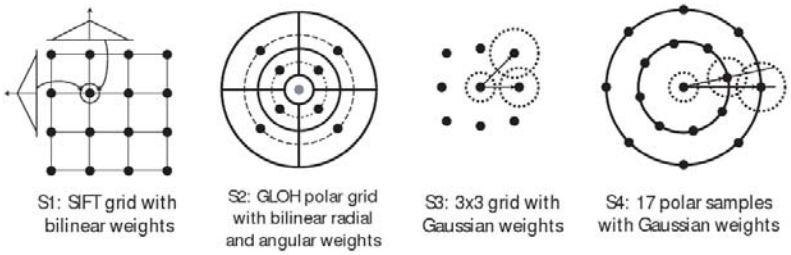


Figura 5.4: Posibles diseños para el área de extracción de componentes. Cada punto indica un segmento, sobre cada uno de ellos se obtienen los componentes del descriptor. S1 y S2 utilizan ponderación bilinear, S3 y S4 ponderan mediante Gaussiana. Las circunferencias punteadas hacen referencia al tamaño del filtro Gaussiano. El descriptor que proponemos utiliza el diseño S4. Figura extraída de [26].

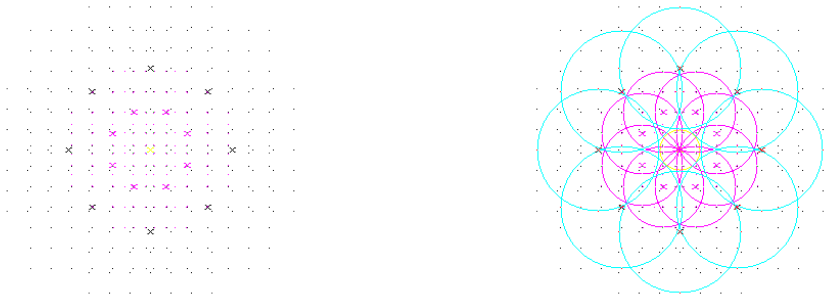
Sobre cada uno de los segmentos se extraen los componentes del descriptor sobre un área que rodea el segmento. El tamaño de dicha área se establece en función del anillo sobre el que se encuentre el segmento, siendo mayor a medida que se aleja de la muestra central. Los componentes de cada uno de las áreas se muestrean en función de la escala del punto característico, siendo la distancia de 2σ entre muestras. Asimismo, los componentes se ponderan utilizando una Gaussiana centrada sobre el segmento. El filtro Gaussiano incrementa su tamaño en cada anillo, ya que las áreas de extracción varían de la misma forma. Las áreas elegidas tras realizar varios experimentos son de dimensiones 3x3, 5x5 y 7x7, correspondiendo al segmento central, los segmentos del primer anillo y los del segundo, respectivamente.

Tras experimentar con los tamaños de los filtros Gaussianos, hemos decidido utilizar una función Gaussiana de tamaño superior al área cubierta, descartando las respuestas con poca intensidad de los extremos del filtro. De esta forma se consiguen resultados similares a los obtenidos con áreas de extracción de 5x5, 7x7 y 9x9, con un filtro Gaussiano ajustado sobre las áreas. La reducción en el tamaño de las áreas es una ventaja ya que se consigue reducir el número de accesos a memoria.

El resultado del diseño final del descriptor se muestra en la figura 5.5. En ella puede observarse como las áreas de extracción están solapadas una sobre la otra. Este hecho va a ser aprovechado para reducir los accesos a memoria requeridos, tal como mostraremos posteriormente en la sección 5.4.

Componentes del descriptor (features)

Tras mostrar el diseño del área de extracción, se describen a continuación los componentes extraídos. Éstos se extraen obteniendo los gradientes en el espacio de escalas calculado durante el proceso de extracción.



(a) Diseño del descriptor

(b) Diseño del descriptor mostrando los filtros Gaussianos

Figura 5.5: Diseño del descriptor. Cada anillo se muestra de un color distinto. a) Las cruces muestran los segmentos que componen el descriptor, mientras que las muestras de adquisición de componentes son los puntos. b) Los círculos indican el tamaño del filtro Gaussiano para cada uno de los segmentos.

Como hemos visto, los componentes del descriptor se extraen sobre cada uno de los 17 segmentos. Para la muestra central se extraen los gradientes en 9 posiciones (3x3), para el primer anillo se realiza en 25 (5x5) y para el segundo la operación sucede 49 veces (7x7). En cada segmento se realiza una suma ponderada por la Gaussiana de los componentes obtenidos en el área correspondiente.

Los componentes extraídos son los llamados T2 en [26]:

$$|D_x| - D_x$$
$$|D_x| + D_x$$
$$|D_y| - D_y$$
$$|D_y| + D_y$$

donde D_x es la derivada horizontal mientras que D_y es la derivada vertical, siendo la orientación del punto característico la que define los ejes. Estos componentes son similares a los que utiliza SURF (ver sección 4.2.10).

Así, para cada segmento se obtienen 4 componentes. Por lo tanto, para 17 segmentos, se obtienen 68 componentes (4x1 + 4x8 + 4x8). De esta forma, el descriptor resultante es un vector de 68 valores.

El vector final se normaliza de forma que la suma de sus componentes sea unitaria

para así lograr invariancia ante cambios lineales de iluminación. Este mismo proceso se lleva a cabo en los descriptores SIFT y SURF.

5.4. Configuración óptima: Algoritmo DART

DART es un extractor de puntos característicos que detecta puntos invariantes a la iluminación y al punto de vista. Los puntos son descritos mediante una implementación del descriptor DAISY [26]. El espacio de escalas se computa de forma eficiente y esta información es reutilizada en el descriptor. Con DART se obtienen mejores resultados en cuanto a exhaustividad - 1-precisión que en SIFT y en SURF, asimismo el tiempo computacional se reduce en factores de 6x y 4x, respectivamente.

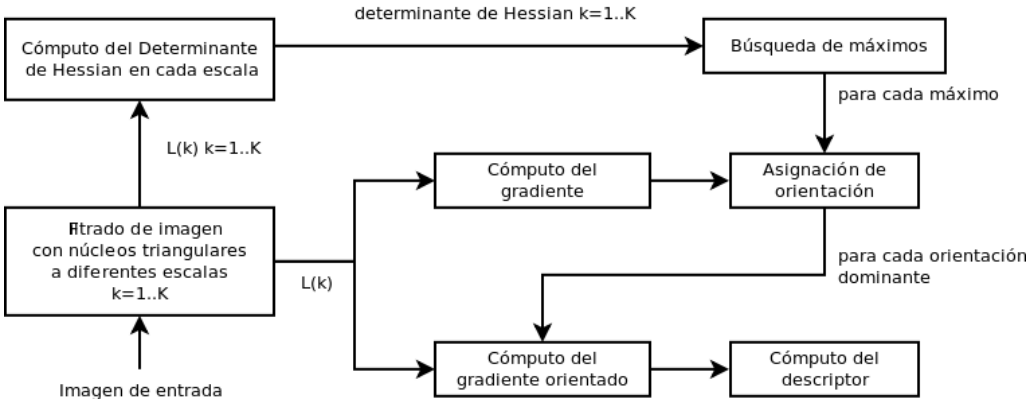


Figura 5.6: Diagrama de bloques del detector y descriptor de puntos DART. La imágenes filtradas con triángulos se reutilizan para la obtención de la orientación y la descripción de puntos.

5.4.1. Detección y descripción de puntos mediante DART

Mediante el algoritmo DART, el cual mejora las características de los detectores y descriptores más reconocidos actualmente, como son SIFT y SURF, se consigue extraer de forma rápida y robusta puntos característicos en imágenes. Los resultados obtenidos con las curvas de exhaustividad - 1-precisión demuestran que DART obtiene los mejores resultados. El algoritmo se ejecuta con un tiempo de cómputo inferior a los obtenidos por SIFT y SURF.

Para el proceso de detección de puntos utilizamos el algoritmo presentado en la sección 5.2.2. El espacio de escalas Gaussiano aproximado se obtiene mediante los filtros triangulares propuestos y detallados anteriormente(ver sección 5.2.5). Seguidamente se calculan las matrices de segundas derivadas parciales necesarias para obtener el determinante de

la matriz Hessiana.

El descriptor utilizado es el que proponemos en la sección 5.3.3, el cual está basado en la distribución de componentes propuesta por SIFT y en los estudios realizados en [26]. Para reducir la carga computacional del algoritmo, reutilizamos las estructuras creadas durante el proceso de detección. Tanto la asignación de la orientación como la descripción de puntos utilizan los gradientes de las matrices del espacio de escalas obtenido anteriormente mediante los filtros triangulares. El algoritmo DART que proponemos incluye un método de obtención rápida del área del descriptor que permite ahorrar notablemente el número de accesos a memoria.

Seguidamente mostramos detalladamente los pasos que seguimos para la obtención de puntos DART. El algoritmo extrae los puntos característicos con sus respectivos descriptores a partir de una imagen de entrada. El número de puntos a extraer se determina como parámetro de entrada del algoritmo, contrariamente a SIFT y SURF, los cuales utilizan un umbral como parámetro, lo cual no permite saber de antemano el número de puntos que van a ser extraídos.

Número de niveles de escala

El primer paso del algoritmo es determinar el número de niveles a calcular en el espacio de escalas. Éste se obtiene en función del tamaño de la imagen de entrada.

Como hemos visto previamente, los puntos se describen en un área con tamaño proporcional a la escala. Por lo tanto, los puntos de escalas mayores requieren un área mayor. Partiendo de este principio, hemos decidido descartar los niveles de escala que requieran un área de descripción mucho mayor que la imagen. Para ello definimos un área rectangular centrada en la imagen de tamaño (0.2Wx0.2H), siendo W y H la amplitud y altura de la imagen, respectivamente.

Empezando por la escala más baja, $\sigma = 1$, se obtiene el tamaño del descriptor correspondiente a un punto situado en una de las esquinas de la región. Si el 30 % del descriptor queda ubicado dentro de la imagen, se procede a ejecutar el mismo procedimiento para el siguiente nivel de escala. El número de niveles se obtiene repitiendo el procedimiento hasta no poder ubicar un descriptor, estableciendo como máximo 15 niveles.

Obtención del espacio de escalas y de puntos máximos

El número de puntos extremos en una imagen varía según la resolución y el contenido de ésta. Por este motivo, no es posible predecir el número de puntos característicos que tendrá una imagen. La mayoría de algoritmos utilizan un umbral sobre el valor del determinante de la Gaussiana para determinar si un punto será finalmente escogido. De

esta forma, se evita obtener un número extremadamente alto de puntos característicos, eliminando los extremos con un valor menor. Sin embargo, fijando el umbral anterior, es imposible obtener un número de puntos concreto. Por este motivo se ha implementado un sistema que determina el umbral en función del número de puntos deseados.

Tras haber fijado el número de niveles, extraemos el espacio de escalas mediante filtros triangulares y, seguidamente, las matrices con los valores del determinante Hessiano. Al formar estas matrices se aprovecha para crear una función de densidad de probabilidad (PDF) con todos los determinantes del espacio de escalas, la cual nos servirá para fijar un primer umbral. Sobre los puntos situados por encima del umbral se ejecuta, posteriormente, la búsqueda de extremos. El número final de puntos obtenidos es el especificado como parámetro de entrada.

Es interesante tener el control del número de puntos que se extraen en una imagen para así seleccionar el número deseado según lo requiera una determinada aplicación.

- **Selección del primer umbral.** Durante la creación de las matrices de determinantes de Hessianas, se obtiene una función de densidad de probabilidad con todos sus valores. A partir de la PDF se obtiene un umbral que sirve para filtrar los puntos sobre los que va a aplicarse la función de búsqueda de extremos, consiguiéndose así una reducción computacional. Todos los puntos situados por debajo del umbral quedan descartados.

  Para obtener el valor del umbral, se recorre la PDF empezando por el final hasta obtener un número de candidatos igual al número de puntos que queremos extraer multiplicado por 200. [1]

- **Clasificación de extremos.** La función de búsqueda de extremos se aplica sobre todos los puntos que tienen un valor de determinante de Hessian superior al umbral fijado anteriormente. Con el valor de todos los máximos se crea una nueva PDF.

  Al recorrer todas las matrices de determinantes Hessianos del espacio de escalas, se selecciona un área de extracción en cada matriz en función del nivel de escala. Ésta se crea de tal forma que el área cubierta por el descriptor esté al menos en un 30 % dentro de la imagen.

  Para seleccionar el número de extremos que van a extraerse se utiliza el mismo sistema utilizado con la PDF de candidatos a extremo. Así se establece un nuevo umbral y se extrae el número de puntos deseado. Debido a que cada uno de los puntos extremos puede tener asignado más de una orientación, se seleccionan un 75 % de los puntos que realmente queremos. Durante la asignación múltiple de orientación puede ocurrir que se alcance el número de puntos deseado. En ese caso se detiene la asignación de puntos para no superar dicho límite. La selección de puntos empieza en la escala mayor hasta la más pequeña para que el descarte se realice en puntos

[1] Con el fin de tener una idea aproximada de cuántos puntos de entre todos los candidatos van a ser extremos, se ha calculado el porcentaje de extremos para la primera imagen de cada una de las escenas de la base de datos de Mikolajczyk.

pequeños, ya que los puntos de mayor escala han demostrado ser más representativos en nuestros experimentos.

Una vez obtenidos los extremos, se determina su posición en sub-píxel y sub-escala siguiendo el método de Brown [15].

Con los pasos anteriores podemos seleccionar el número de puntos característicos que extraemos, sin embargo, existen imágenes que tienen pocas regiones características. En este caso se obtienen menos puntos gracias a la utilización de umbral mínimo para el valor de los determinantes fijado en 4000. Si no se utilizase este umbral, podría darse el caso en que se extraen puntos en regiones que no tienen un contenido representativo en la imagen.

Asignación de orientación y descripción de puntos

A cada punto se le asigna un máximo de 4 orientaciones. El procedimiento seguido ha sido detallado previamente en la sección 5.3.2.

A continuación se extrae el descriptor de 68 dimensiones siguiendo los pasos mostrados en 5.3.3. Con el objetivo de reducir los tiempos de cómputo, se realizan dos aproximaciones que se detallan seguidamente. La primera de ellas consiste en utilizar valores enteros para la obtención de los componentes del descriptor, lo cual significa un gran ahorro en espacio de memoria frente a la utilización de valores decimales en coma flotante. Finalmente se muestra cómo obtener el área de descripción reduciendo notablemente el número de accesos a memoria.

- Utilización de valores enteros en los componentes descriptivos DART. El extractor DART utiliza valores enteros de 8 bits en los componentes de su descriptor. Concretamente, se trata de un "unsigned char" en el lenguaje de programación C. El intervalo de valores para esta variable es de [0, 255].

  El motivo por el que se utiliza un "'unsigned char" en substitución de un "float" es la reducción del espacio de memoria que supone. El ahorro de memoria es especialmente importante al portar el extractor a un terminal móvil, ya que se consiguen grandes reducciones en el tiempo de ejecución, debido a las limitaciones de memoria y del procesador de estos dispositivos.

  En los experimentos de la sección 5.4.2 demostramos como la utilización de valores enteros no repercute prácticamente en las evaluaciones, por este motivo no van a usarse componentes decimales en la implementación DART.

- Diseño del área de descripción DART. Como hemos visto en la sección 5.3.3, el área de extracción de componentes descriptores está compuesta por un segmento central y 2 anillos concéntricos con 8 segmentos cada uno. El tamaño de los núcleos utilizados para la acumulación de componentes es, del centro al segundo anillo, de 3x3, 5x5 y 7x7 muestras.

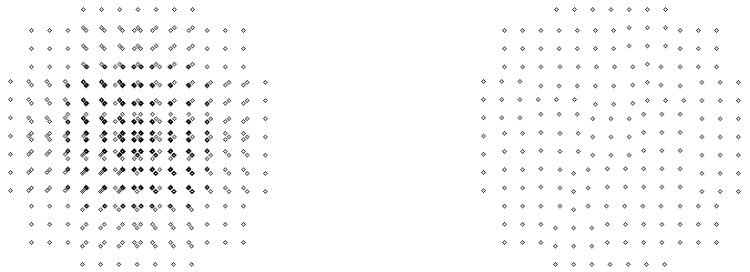
Versión	Muestras contenidas	Accesos a memoria
Diseño del área sin reducción	601	2404
Diseño reducido	197	788

Cuadro 5.2: Número de accesos a memoria en la computación del descriptor para cada punto característico utilizando los dos tipos de áreas de extracción.

En la figura 5.7 (a) se muestra la posición que ocupan cada una de las muestras sobre las que tendremos que acceder para calcular los gradientes necesarios para obtener los componentes del descriptor. Debido al solapamiento existente entre las muestras de diversos segmentos, algunas de ellas se encuentran en posiciones muy cercanas.

El método para optimizar la computación del área de extracción consiste en utilizar tan sólo una de estas muestras para todos los puntos cercanos. Sobre cada una de las muestras se obtiene el gradiente en dos direcciones (ver sección 5.3.3), por lo tanto, estas operaciones se realizarán solamente una vez en aquellas zonas ocupadas por puntos cercanos. El resultado obtenido es el que aparece en la figura 5.7 b), con el cual se consigue reducir el número de muestras de 601 a 197.

Finalmente se obtienen los componentes T2 del descriptor (ver sección 5.3.3). Para ello se acumulan los componentes de las muestras pertenecientes a cada segmento, con la ponderación Gaussiana correspondiente.



(a) Diseño del área sin optimización. (b) Diseño del área con optimización.

Figura 5.7: Áreas del descriptor DART. 2 anillos, 8 segmentos por anillo. Núcleos (de dentro a fuera) 3x3, 5x5, 7x7.

Gracias a la reducción de muestras, el tiempo de cómputo se reduce considerablemente ya que se ahorran operaciones de gradiente, las cuales precisan accesos a memoria, como muestra el cuadro 5.2. Anteriormente, por cada punto característico, se realizaban 2404 accesos durante la computación del descriptor. Con el nuevo diseño esta cifra se ve reducida a 788 accesos, lo que significa un ahorro de 1616 accesos por punto.

En la siguiente sección se evalúan los resultados obtenidos mediante un diseño del área de descripción optimizado, frente a los logrados con la implementación original.

5.4.2. Experimentos realizados sobre DART

Para llegar a la configuración DART mostrada en la sección anterior (ver sección 5.4) hemos realizado una serie de experimentos para evaluar los resultados. Los dos primeros muestran cómo afecta a las curvas de exhaustividad - 1-precisión (ver sección 4.3.1), la conversión de valores de coma flotante a enteros, así como la optimización del diseño del área de extracción de componentes descriptivos. Seguidamente se muestran las configuraciones que hemos considerado antes de seleccionar un descriptor de 68 dimensiones y los diseños finales para el cálculo de la orientación y la descripción de puntos. Finalmente se plantea una opción que no ha sido implementada en el algoritmo DART en su configuración definitiva pero que puede ser interesante en función de la aplicación que se utilice. El método consiste en muestrear la imagen de entrada para reducir los tiempos de cómputo.

Conversión de valores en coma flotante a enteros de 8 bits

El siguiente experimento muestra el impacto que tiene la utilización de valores enteros en los componentes del descriptor. Los resultados se muestran como curvas de exhaustividad - 1-precisión en la figura 5.8.

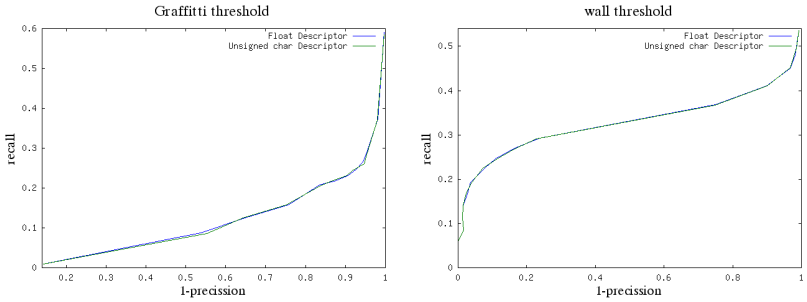
Tal como puede observarse, el cambio de coma flotante (float) a entero de 8 bits (unsigned char) no supone ninguna pérdida de calidad en el descriptor, ya que los resultados en las curvas de exhaustividad son prácticamente idénticos.

Optimización de la computación del área del descriptor

El método optimizado para extraer el descriptor de un punto característico consigue reducir el número de accesos a memoria considerablemente, tal como se ha detallado en la sección 5.4.1. Seguidamente se muestra el impacto que tiene sobre los resultados, utilizando nuevamente las curvas de exhaustividad - 1-precisión para su evaluación.

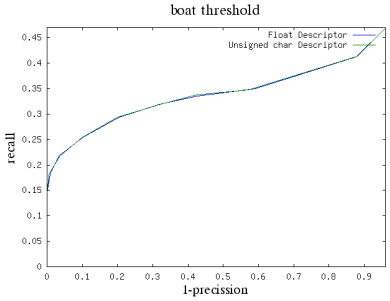
El resultado se muestra en la figura 5.9, donde vT2-4-2r8s-L357-Sigma579-GaussNorm es la versión correspondiente al área sin optimizar y vT2-4-2r8s-L357-Sigma579-GaussNorm-NewLayout pertenece a la versión optimizada.

Como vemos, el impacto de funcionamiento es mínimo. Los resultados en exhaustividad son prácticamente idénticos para las dos versiones, tan sólo en las escenas de Wall, Trees y Leuven puede verse un pequeño empeoramiento en la versión optimizada.

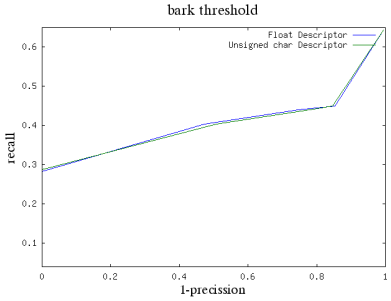


(a) Graffitti

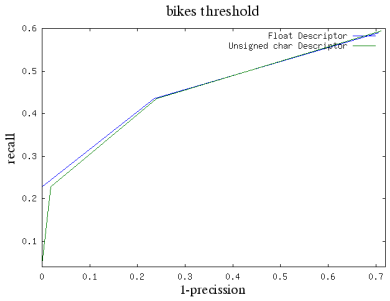
(b) Wall



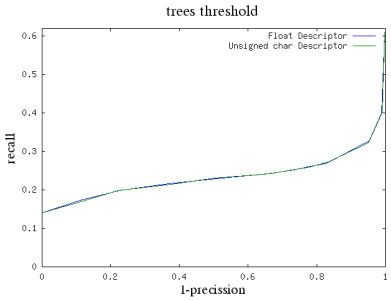
(c) Boat



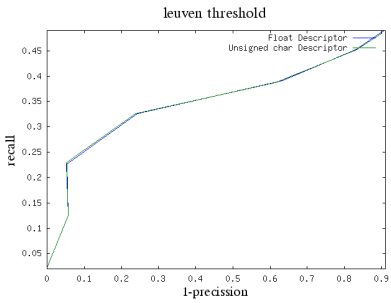
(d) Bark



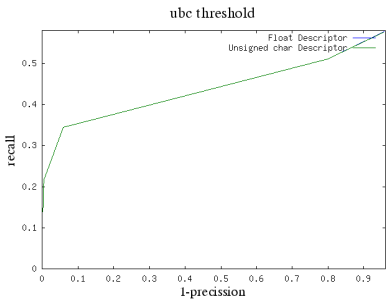
(e) Bikes



(f) Trees

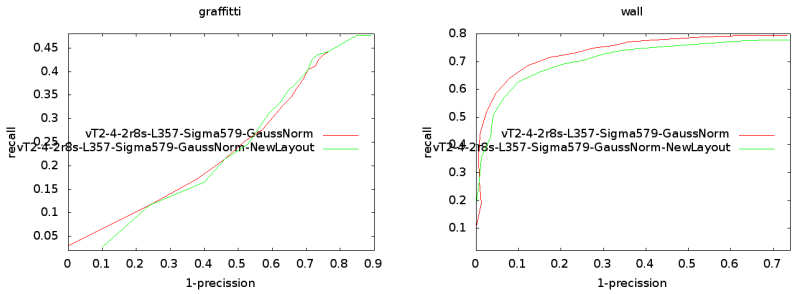


(g) Leuven



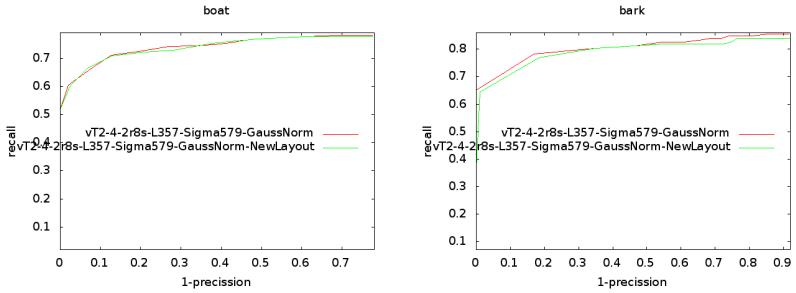
(h) UBC

Figura 5.8: Comparación de dos versiones del extractor DART. Una utiliza coma flotante en los componentes de su descriptor mientras que la otra usa enteros de 8 bits. Las coincidencias han sido obtenidas por umbral (ver sección 4.1.3).



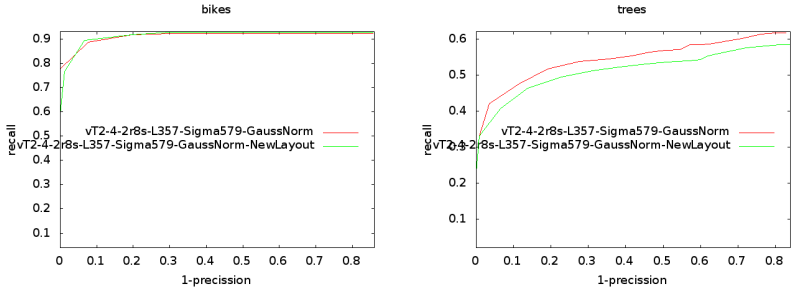
(a) Graffitti

(b) Wall



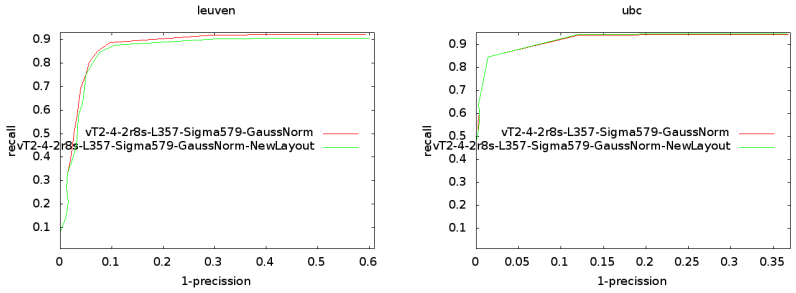
(c) Boat

(d) Bark



(e) Bikes

(f) Trees



(g) Leuven

(h) UBC

Figura 5.9: Curvas de exhaustividad para el extractor DART comparando el diseño original con su versión optimizada. Coincidencias obtenidas mediante vecino más cercano (ver sección 4.1.3).

Con los resultados obtenidos mediante estos experimentos se ha decidido utilizar siempre la versión optimizada, ya que el tiempo de ejecución se reduce significativamente mientras que los resultados son prácticamente los mismos.

Dimensionalidad del descriptor

Tal como se explica en la sección 5.3.3, el descriptor implementado tiene 68 dimensiones. Antes de establecer esta cifra como definitiva, hemos experimentado con descriptores de mayores y menores dimensiones. Para ello se ha modificado el número de anillos y segmentos sobre los que se extraen los componentes descriptivos.

En los gráficos mostrados en la figura 5.10, se comparan descriptores de 36, 68 y 100 dimensiones. Para detectar los puntos se ha usado el detector propio con Gaussianas aproximadas con filtros triangulares (ver sección 5.2.5). Asimismo, todos los resultados se comparan con el detector y descriptor SIFT estándar de 128 dimensiones.

Las características de los descriptores implementados se muestran en el cuadro 5.3.

Dimensiones	**Anillos (sin muestra central)**	**Segmentos / anillo**
36	2	4
64	2	8
100	3	8

Cuadro 5.3: Descriptores con distinta dimensionalidad.

En Graffitti y Wall, el descriptor propio con mejores resultados es el de 68 dimensiones, seguido del descriptor de 100 dimensiones y finalmente por el de 36. En Graffitti, SIFT obtiene los peores resultados, mientras que lo contrario ocurre en Wall, dónde es el que consigue los mejores resultados en exhaustividad.

En la escena Boat, dónde se producen cambios de escala, los resultados de los descriptores propuestos aparecen ordenados por orden de dimensionalidad, siendo el de 100 el que obtiene mejores resultados. SIFT obtiene los peores resultados. En Bark, una escena con textura como Wall, vuelve a ocurrir lo contrario, SIFT actúa como el mejor descriptor.

La evaluación de desenfoque muestra como la exhaustividad y la dimensionalidad del descriptor son directamente proporcionales. Todas las versiones del extractor propio superan a SIFT.

Leuven, escena donde se produce un cambio de iluminación, así como UBC, dónde la imagen se comprime utilizando JPEG, no se ven especialmente afectadas por la dimensionalidad del descriptor, ya que todas las versiones del descriptor aportan resultados muy similares. SIFT vuelve a obtener peores resultados en cuanto a exhaustividad.

Observando los gráficos de la figura 5.10 conjuntamente vemos que cualquiera de las

versiones propuestas de detección y descripción supera la versión estándar de SIFT. Un estudio más preciso se lleva a cabo en la sección 5.5.2.

Las conclusiones a las que hemos llegado tras realizar estos experimentos son las siguientes.

1. El descriptor con 36 dimensiones es claramente inferior a las versiones con 68 y 100 dimensiones.
2. La mejora obtenida al pasar de 68 a 100 dimensiones es muy pequeña o incluso inexistente.

A parte de las versiones aquí mostradas, se han implementado otras versiones con un número de dimensiones que va desde 36 a 100 en intervalos de 8 dimensiones. Entre los descriptores de 36 a 68 dimensiones, los resultados mejoran de forma aproximadamente lineal. A partir de las 68 dimensiones esta mejora incrementa de forma más lenta hasta prácticamente desaparecer.

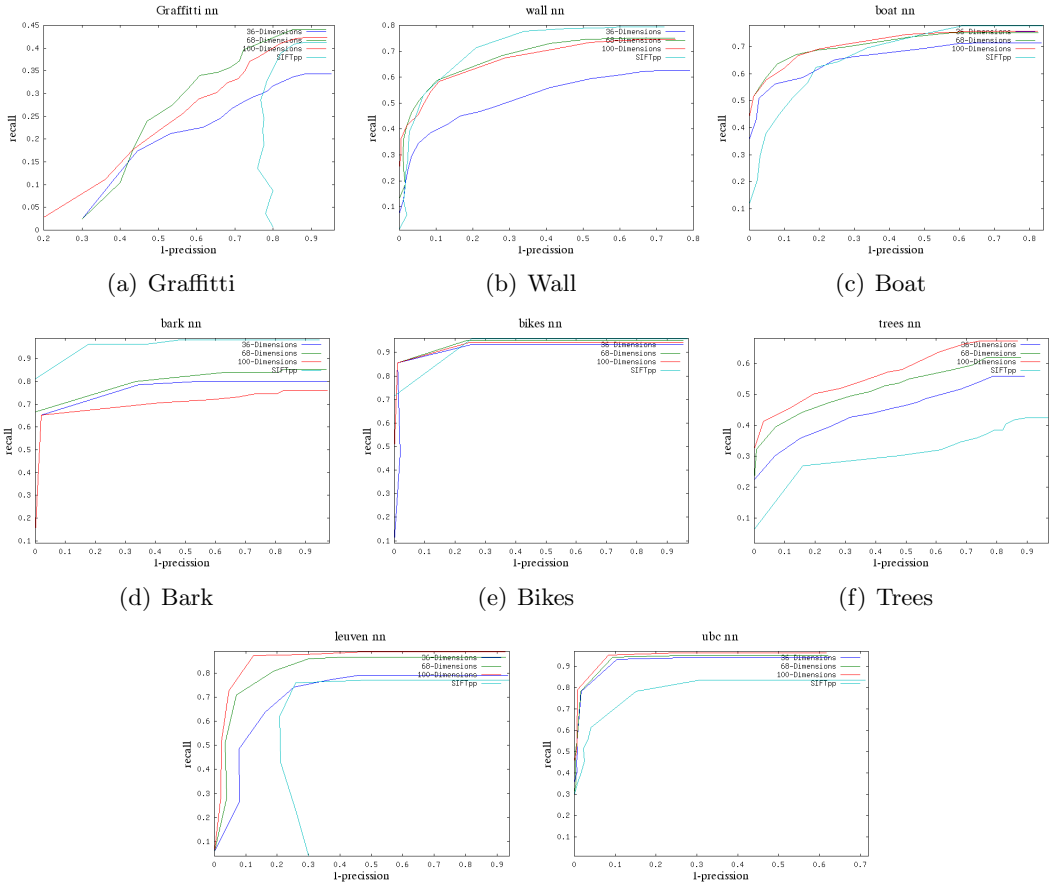
Para la creación de un descriptor de 100 dimensiones se han utilizado varias versiones modificando el número de anillos y segmentos, así como variando el espaciado entre muestras pertenecientes a cada uno de los anillos. Hemos podido apreciar que, a partir de más de 8 segmentos por anillo, el descriptor prácticamente no mejora. Los resultados mostrados en la figura 5.10 son los que consiguen mejores curvas.

El incremento de la dimensionalidad del descriptor implica tiempos de cómputo mayores y una búsqueda de coincidencias más costosa. Por lo tanto, no tiene sentido incrementar la dimensionalidad si no se obtienen mejoras evidentes en los resultados. Por esta razón hemos elegido el descriptor de 68 dimensiones descrito en el cuadro 5.3.

Cabe destacar que, con un descriptor de 68 dimensiones, se consiguen resultados superiores en las curvas de exhaustividad a los obtenidos por SIFT (que tiene un descriptor de 128 dimensiones), con todas las ventajas que supone la reducción de la dimensionalidad.

Variación del diseño del orientador y del descriptor

Como ya se ha visto anteriormente, para computar la orientación y el descriptor de un punto característico se establece un área alrededor del punto, para cada uno de los casos. En el caso del orientador se trata de una región circular dónde el único parámetro variable es el diámetro. En el descriptor, sin embargo, pueden variarse 3 parámetros: el número de anillos, el número de segmentos y el tamaño de los núcleos. Según los valores que se elijan, el funcionamiento variará. La versión mostrada en la sección 5.4 es la implementación estándar, pero en algunos casos puede interesar utilizar alguna de las alternativas propuestas a continuación.



(a) Graffitti (b) Wall (c) Boat

(d) Bark (e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 5.10: Comparación del descriptor propio con distinto número de dimensiones (36, 64 y 100), juntamente con SIFT. Las técnicas de búsqueda de coincidencias utilizadas es por vecino más cercano (4.1.3).

Diámetro	Muestras contenidas	Accesos a memoria
11	109	436
7	45	180

Cuadro 5.4: Número de accesos a memoria en la computación de la orientación para cada punto característico en función del diámetro del área circular.

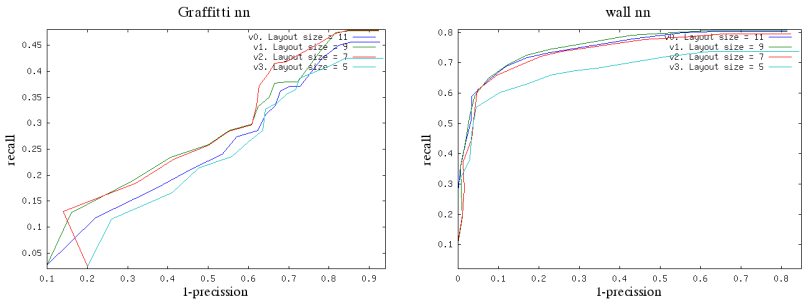
- **Modificación del área del orientador**. En el caso del orientador tan sólo se va a modificar el diámetro del área sobre el que se extraen muestras de la matriz (ver sección 5.3.2). En su versión estándar, el diámetro tiene una longitud de 11σ. En experimentos no mostrados en este trabajo vimos cómo aumentando el área los resultados apenas mejoraban. Debido al coste computacional que requiere cada acceso a la matriz, decidimos establecer un diámetro de 11σ como valor máximo.

  En la figura 5.11 aparecen los resultados para cuatro versiones del extractor con diámetros para el área circular de 5σ, 7σ, 9σ y 11σ, respectivamente.

  Los resultados muestran como, en la mayoría de casos, las versiones de DART están ordenadas por tamaño del área, de mayor a menor, como era previsible. Sin embargo, los diseños de tamaños 7σ, 9σ y 11σ tienen resultados muy similares, mientras que el área de tamaño 5σ presenta peores valores de exhaustividad.

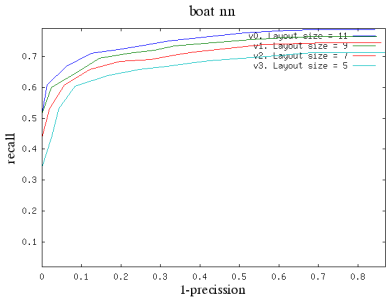
  Así pues, el área del orientador se puede reducir hasta un diámetro de 7σ. En este caso, los resultados son apenas peores y se reducen los accesos a memoria, pasando de 436 a 180 accesos por punto característico, tal como muestra el cuadro 5.4.

  Esta reducción puede ser interesante en caso de querer portar el extractor DART a un terminal móvil, ya que reduciendo los accesos a memoria se consigue disminuir el tiempo de ejecución, especialmente en estos dispositivos. Sin embargo, en su versión general, no reducimos el diámetro debido a la pequeña disminución en los resultados. Como veremos a continuación, el número de accesos a memoria puede ser reducido modificando el área del descriptor.

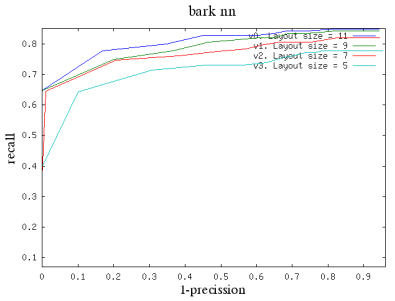


(a) Graffitti

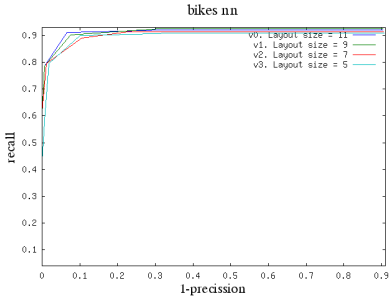
(b) Wall



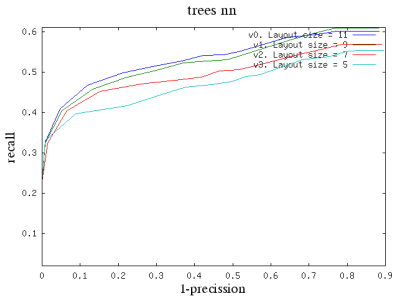
(c) Boat



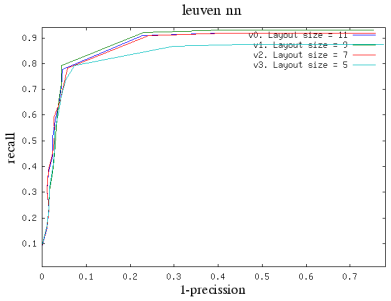
(d) Bark



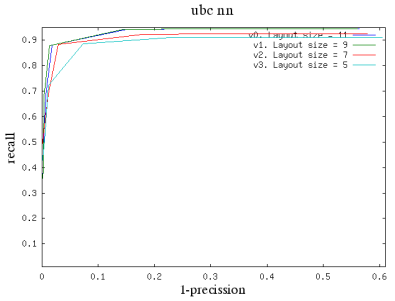
(e) Bikes



(f) Trees



(g) Leuven



(h) UBC

Figura 5.11: Curvas de exhaustividad para el extractor DART con distintos diámetros en el área circular del orientador: 5σ, 7σ, 9σ y 11σ. Coincidencias obtenidas mediante vecino más cercano (ver sección 4.1.3).

- **Modificación del área del descriptor**. En esta sección se demuestra porqué la

opción final elegida es más eficiente que todas las opciones contempladas anteriormente.

Inicialmente se utilizaba un área con tamaños de los núcleos, del punto central al segundo anillo, de 5x5, 7x7 y 9x9. El número de anillos seguía siendo 2, así como 8 segmentos por anillo. Sin embargo, en este caso se utilizaban matrices Gaussianas de tamaño igual y no mayor al área de los núcleos (ver sección 5.2.2).

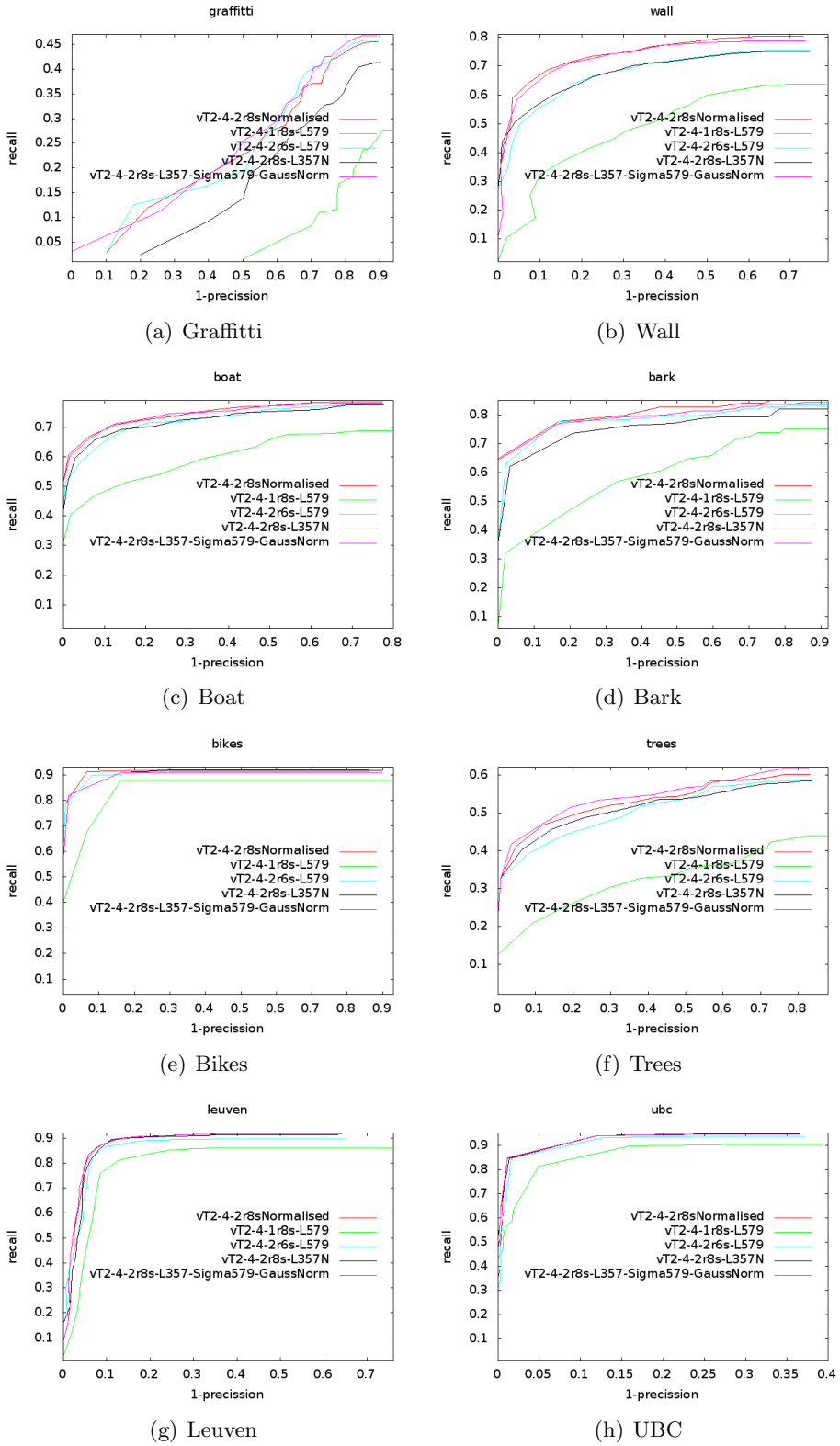
En la figura 5.12 se compara esta versión con la versión final definida en la sección 5.4, así como con otras 3 versiones, descritas a continuación.

- vT2-4-2r8sNormalised: versión inicial, 2 anillos, 8 segmentos, núcleos 5x5, 7x7 y 9x9.
- vT2-4-1r8s-L579: 1 anillo, 8 segmentos, núcleos 5x5, 7x7 y 9x9.
- vT2-4-2r6s-L579: 2 anillos, 6 segmentos, núcleos 5x5, 7x7 y 9x9.
- vT2-4-2r8s-L357N: 2 anillos, 8 segmentos, núcleos 3x3, 5x5 y 7x7.
- vT2-4-2r8s-L357-Sigma579-GaussNorm: versión final, 2 anillos, 8 segmentos, núcleos 3x3, 5x5 y 7x7 con Gaussianas correspondientes a núcleos 5x5, 7x7 y 9x9.

La versión que utiliza un solo anillo tiene, claramente, peor comportamiento. Si se reduce el número de segmentos de 8 a 6, el deterioro no es tan evidente pero también existe. En cambio, la versión final es prácticamente igual a la original y consigue un considerable ahorro en el número de accesos a memoria (ver cuadro 5.5). La reducción de accesos conseguida es incluso mayor que el conseguido en el experimento anterior, correspondiente a la orientación.

Versión	**Muestras contenidas**	**Accesos a memoria**
vT2-4-2r8s-L579 (versión inicial)	**1065**(5*5*1 + 7*7*8 + 9*9*8)	4260
vT2-4-2r8s-L357-Sigma579-GaussNorm (versión final)	**601** (3*3*1 + 5*5*8 + 7*7*8)	2404

Cuadro 5.5: Número de accesos a memoria en la computación del descriptor para cada punto característico en función del diámetro del área.



(a) Graffitti (b) Wall

(c) Boat (d) Bark

(e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 5.12: Curvas de exhaustividad para el extractor DART con distintos diseños para el descriptor. Coincidencias obtenidas mediante vecino más cercano (ver sección 4.1.3).

Sub-Muestreo

En los extractores SIFT y SURF, durante la creación del espacio de escalas, se realiza un sub-muestreo dividiendo la imagen entre dos en ambos ejes cada vez que se dobla la escala. De esta forma se consigue reducir el tiempo de cómputo.

En el siguiente experimento se implementan dos formas de sub-muestreo sobre el extractor DART estándar. La principal diferencia al compararlo con SIFT y SURF radica en que, en DART, se mantiene la posibilidad de buscar extremos en cualquier nivel del espacio de escalas excepto el primero y el último, comparando imágenes de distinto tamaño pertenecientes a distintas octavas.

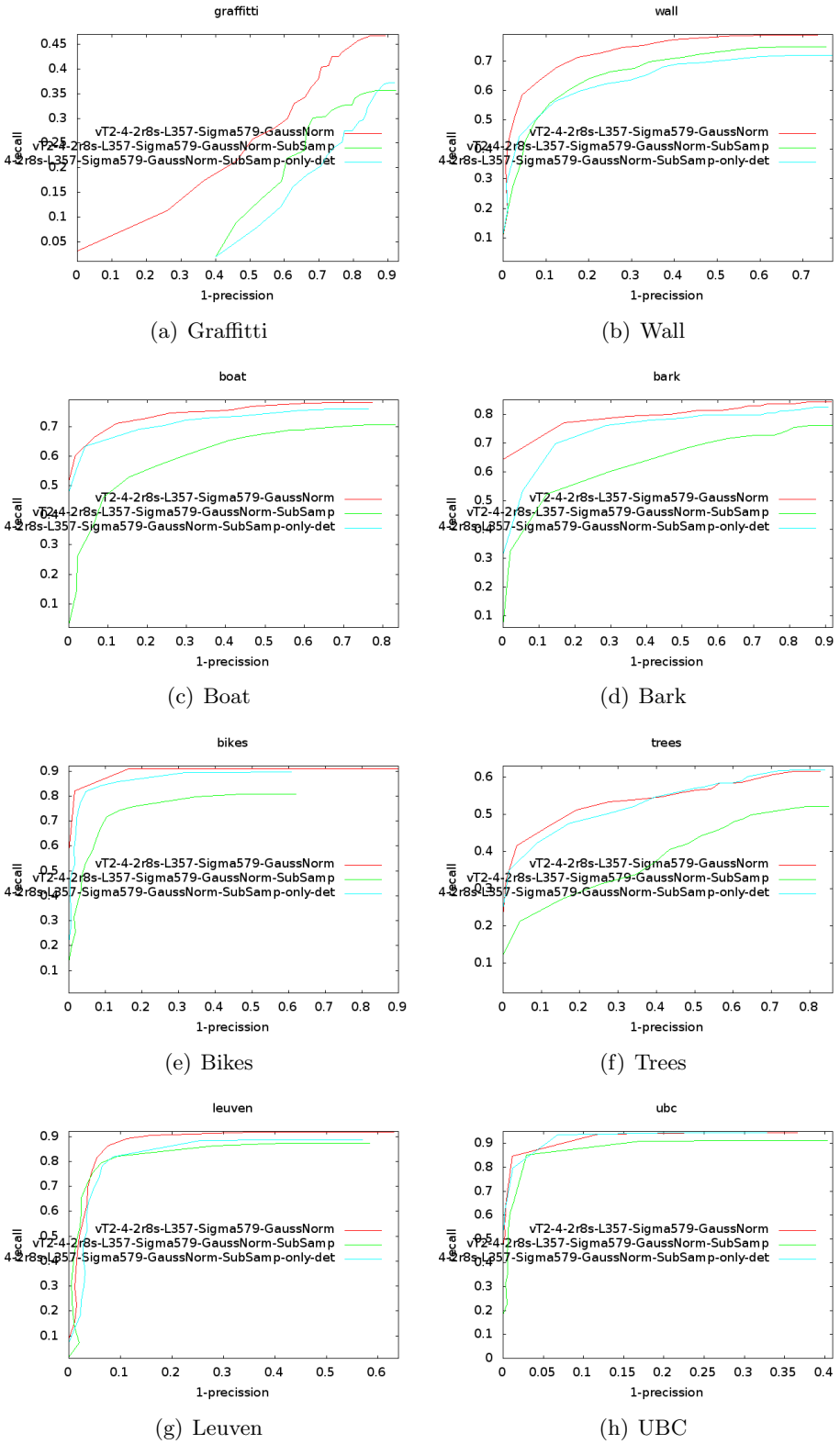
- vT2-4-2r8s-L357-Sigma579-GaussNorm-SubSamp. En esta versión, el sub-muestreo se realiza sobre los filtros triangulares computados durante la creación del espacio de escalas. Posteriormente, las matrices de aproximación del determinante Hessian mantienen dicho sub-muestreo. De esta forma se reduce considerablemente el número de operaciones a realizar, aunque el hecho de reducir los filtros triangulares provoca que éstos pierdan precisión.

- vT2-4-2r8s-L357-Sigma579-GaussNorm-SubSamp-only-det. En este caso se trata de una versión de sub-muestreo no tan agresiva. Los filtros triangulares mantienen su tamaño inicial y la reducción no se produce hasta la formación de las matrices Hessian. De esta forma el espacio de escalas mantiene las características iniciales y tan sólo se pierden posiciones sobre las que obtener extremos.

En la figura 5.13 puede verse como las curvas de exhaustividad para la segunda versión son mejores que para la primera. De hecho, el sub-muestreo sobre los filtros triangulares es el que provoca una reducción drástica de resultados. Por este motivo, la primera de las versiones será descartada y sólo el sub-muestreo sobre las matrices Hessian será tenida en cuenta en versiones futuras.

Obviamente, el sub-muestreo tiene una repercusión negativa sobre los resultados. Sin embargo, la aplicación tan sólo en las matrices de determinantes hace que en algunos casos los resultados sean muy similares a los originales. Las escenas más afectadas por el sub-muestreo son Graffitti, Wall y Bark, aquellas dónde se producen cambios de escala, rotaciones y cambios de ángulo.

El sub-muestreo es una de las opciones disponibles en DART si se requiere una reducción drástica del tiempo de cómputo del programa. Esta opción tan sólo se usará en los casos en que la reducción de puntos a extraer y la reducción de núcleos no sea suficiente. En todo caso, el sub-muestreo tan sólo será necesario si se desea utilizar DART en un terminal móvil con prestaciones no muy altas.



(a) Graffitti (b) Wall

(c) Boat (d) Bark

(e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 5.13: Comparativa entre el descriptor DART estándar y sus dos versiones con sub-muestreo. Coincidencias obtenidas mediante vecino más cercano (ver sección 4.1.3).

5.5. Evaluación de los métodos propuestos

A continuación se evalúan los detectores propuestos en la sección 5.2, obteniendo la repetibilidad en las escenas proporcionadas por Mikolajczyk (ver sección 3.3.1). Los tres detectores que mostramos se diferencian principalmente por el filtro que utilizan para la creación del espacio de escalas. El primero de ellos utiliza la función Gaussiana sin ningún tipo de aproximación, mientras que los otros dos utilizan aproximaciones a partir de los filtros SWII y los triangulares, respectivamente.

Tras obtener la repetibilidad de los detectores, se procede a una evaluación mediante las curvas de exhaustividad - 1-precisión. En ellas se comparan dos versiones del descriptor propuesto en la sección 5.3.3, los cuales utilizan el detector propuesto con filtros triangulares. Junto a ellos se evalúa el detector y el descriptor SIFT, que sirve como referencia ya que es el algoritmo con mejores resultados hasta la fecha. Igualmente mostramos una versión que extrae los puntos con filtros triangulares y posteriormente los describe con SIFT. De esta forma podemos separar la evaluación del detector con la del descriptor.

5.5.1. Evaluación del detector (repetibilidad)

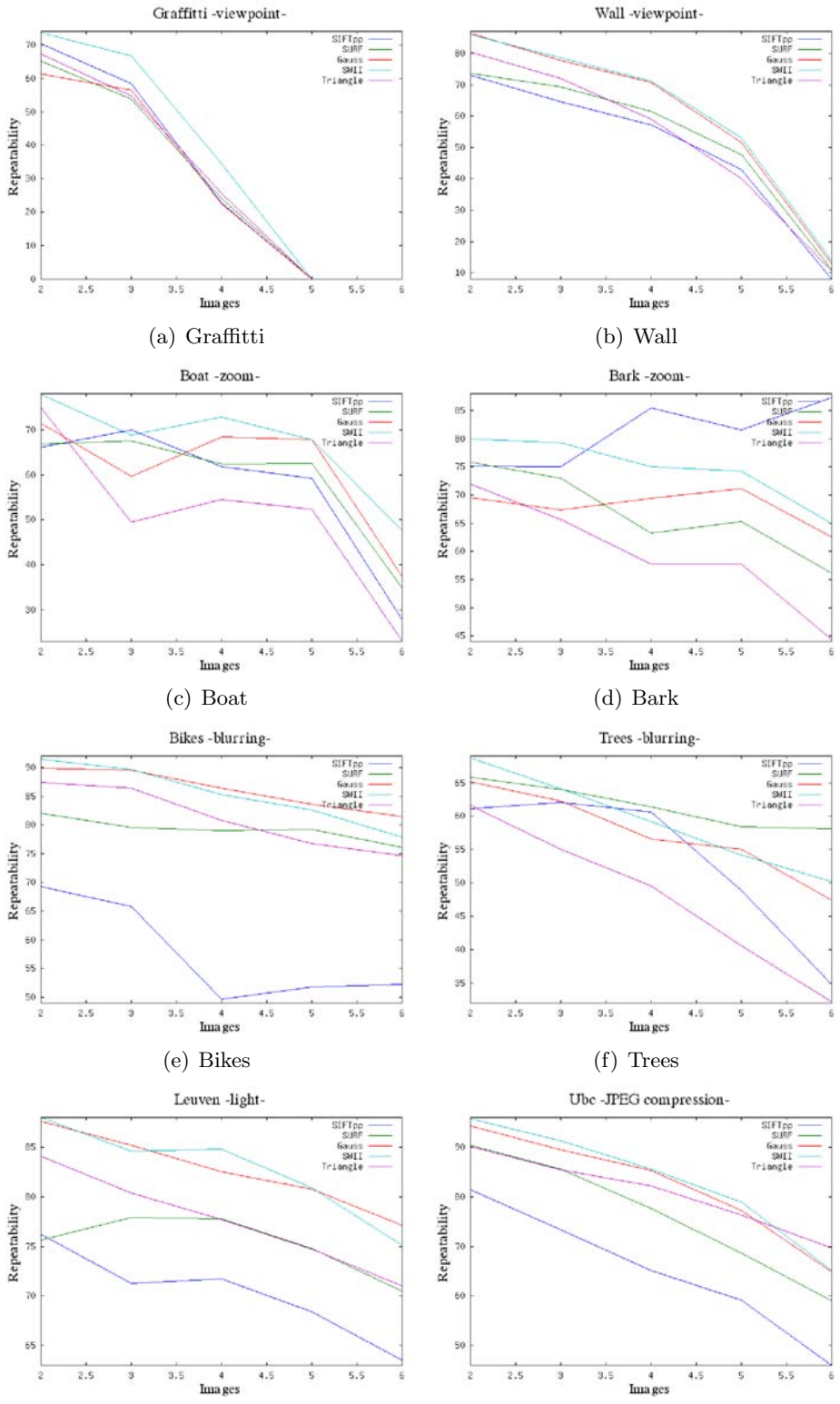
A continuación mostramos los gráficos con la repetibilidad obtenida para cada una de las escenas de la base de datos de Mikolajczyk. Los detectores que evaluamos son los tres que proponemos (basados en Gaussianas, filtros SWII y filtros triangulares), juntamente con SURF y SIFTpp, una implementación de código abierto de SIFT con la cual se obtienen resultados muy similares a los obtenidos en su implementación original.[2]

Una visión general de las figuras muestra que el detector con mayor repetibilidad es SWII. A priori, lo lógico sería que la mejor respuesta fuese para el detector de filtros Gaussianos, ya que utiliza el determinante de la matriz Hessian sin ningún tipo de aproximación. Sin embargo, tanto el detector SWII como el basado en filtros triangulares, presentan mejoras en el algoritmo que no han sido utilizadas en la versión sin aproximar. Algunos ejemplos son la variación del número de niveles en el espacio de escalas o la variación del tamaño de las ventanas durante la obtención de máximos.

En las imágenes con cambio de vista los mejores detectores son los basados en Hessianas sin aproximar y en SWII, mientras que el filtro triangular se encuentra al nivel de SIFTpp y SURF.

En cuanto a las escenas con cambio de escala, vemos como SIFTpp tiene claramente la mejor respuesta en la escena bark, mientras que, en este caso, el filtro triangular obtiene la peor repetibilidad.

[2]Con el objetivo de conseguir unos niveles de repetibilidad comparable, el número de puntos extraído para cada imagen es similar en todos los detectores.



(a) Graffitti (b) Wall

(c) Boat (d) Bark

(e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 5.14: Comparativa de los detectores SIFTpp, SURF, Gaussiano, SWII y triangular.

Las escenas con cambio en el enfoque obtienen resultados muy distintos en Bikes y en Trees, por este motivo se analizan por separado. La escena Bikes, formada por imágenes estructurales, muestra cómo el detector basado en Hessianas sin aproximar y en SWII, una vez más, tienen resultados superiores a SURF, el cual está al mismo nivel del filtro triangular. En este caso, SIFT está muy por debajo del resto de detectores. En Trees, una escena con textura, es SURF quien consigue los mejores resultados, seguido de SWII y Hessian, que tienen resultados muy parecidos. El detector basado en filtros triangulares tiene una mala respuesta para esta escena. Los buenos resultados de SURF para las escenas con desenfoque se atribuyen a los filtros Box que utilizan. Éstos son equivalentes a las derivadas parciales Hessianas filtradas a muy baja frecuencia, es decir, extremamente desenfocados.

A continuación se analiza el comportamiento ante cambios lineales sobre la iluminación en la escena Leuven. Los detectores basados en el determinante Hessiano y los SWII vuelven a obtener las mejores respuestas, seguidos del basado en filtros triangulares. Las tres versiones son mejores que SIFTpp y SURF. En la escena UBC, donde se produce una compresión JPEG, los detectores siguen el mismo comportamiento.

Tras estudiar estos resultados, hemos concluido que el mejor detector en cuanto a repetibilidad es el obtenido mediante filtros SWII. Esta versión supera el detector de puntos de referencia, el cual obtiene un espacio Gaussiano sin aproximaciones. Dicha mejora se debe a las mejoras que hemos introducido en el algoritmo de detección de puntos basado en aproximaciones del espacio de escalas. SWII es en la mayoría de escenas la mejor opción, situándose por encima de SIFTpp y SURF.

El detector basado en filtros triangulares obtiene resultados similares a SURF en todas las escenas, excepto en las que se produce un cambio de escala. En estas imágenes los filtros triangulares obtienen peores niveles de repetibilidad debido a las limitaciones creadas por las aproximaciones triangulares.

SIFTpp varía mucho sus resultados dependiendo del cambio que se produzca en la escena. Así, en la escena Bark actúa como el mejor de los descriptores, pero en cambio en Bikes obtiene resultados muy pobres.

Aunque los resultados del detector basado en filtros triangulares no alcanzan los niveles del basado en SWIIs, hemos decidido utilizar el primero en al algoritmo DART. El motivo principal es la velocidad a la que podemos crear el espacio de escalas. Igualmente, en la mayoría de transformaciones, se consiguen resultados similares a SURF, uno de los mejores detectores existentes en la actualidad.

5.5.2. Evaluación del descriptor (exhaustividad - 1-precisión)

El nuevo descriptor propuesto en la sección 5.3.3, se compara con SIFT en términos de curvas 1-precisión - exhaustividad. El motivo por el que se compara con SIFT en lugar

de con SURF es por los mejores resultados obtenidos en el primero, los cuales nos sirven como referencia a superar.

En la comparativa se utilizan distintas combinaciones de detectores y descriptores. La descripción de cada una de ellas se describe a continuación.

- timKPDescriptor. Los puntos se extraen con el detector basado en filtros triangulares para posteriormente ser descritos por el descriptor propuesto. El tamaño de los núcleos para cada uno de los anillos es de 5x5, 7x7 y 9x9, empezando por el punto central y siguiendo con los segmentos del primer y segundo anillo, respectivamente.
- timKPDescriptorFast Se trata de una versión más rápida de computar que timKPDescriptor. Tan sólo se diferencian por el tamaño de los núcleos usados. En este caso los tamaños son 1x1, 3x3 y 5x5.
- DetHTriangleExtractorSIFTppDescriptor Esta versión utiliza el mismo detector que los dos anteriores, así como el mismo proceso de obtención de orientación. Sin embargo, el descriptor utilizado es la versión de código abierto de SIFT, SIFTpp.
- SIFTpp Finalmente el extractor SIFTpp es utilizado tanto para detectar, asignar la orientación y describir los puntos característicos.

Con las cuatro versiones anteriores se pueden obtener todas las comparativas necesarias. Comparando timKPDescriptor con SIFTpp se ve si las versiones propuestas de detección y descripción mejoran los resultados existentes en SIFT. Por otra parte, la evaluación de timKPDescriptor frente a DetHTriangleExtractorSIFTppDescriptor, tan sólo diferenciados por el descriptor, consigue valorar únicamente los descriptores. Finalmente, timKPDescriptor y timKPDescriptorFast son comparados para ver hasta qué punto se ven perjudicados los resultados por reducir el número de muestras utilizadas en la generación de componentes del descriptor.

En los resultados mostrados en la figura 5.15 podemos ver como timKPDescriptor es superior a SIFTpp en todas las imágenes excepto en wall y bark. El motivo es que el detector triangular sufre más que SIFT ante cambios de puntos de vista y de escala. Este mal comportamiento es atribuido a la aproximación de los filtros Gaussianos mediante filtros triangulares.

En la comparación de descriptores (timKPDescriptor frente a DetHTriangleExtractorSIFTppDescriptor), el primero supera en prácticamente todas las escenas al descriptor SIFTpp, por lo que podemos afirmar que el descriptor propio es superior a SIFT en cualquiera de los casos representados en esta base de datos.

La versión rápida del extractor propio, timKPDescriptorFast es inferior en todos los casos a su versión más compleja. Dependiendo de la escena, el perjuicio varía. Sin embargo timKPDescriptorFast es mejor que SIFTpp en todas las escenas dónde timKPDescriptor también lo es.

Esta versión tan sólo se utilizará en casos en que sea de especial importancia la reducción del tiempo de cómputo o/y el número de accesos a memoria.

5.5.3. Evaluación del coste computacional de DART

Como hemos visto en las secciones anteriores, el algoritmo de detección y descripción de puntos característicos DART lo hemos diseñado de forma que sea eficiente computacionalmente. En esta sección, se muestra el tiempo requerido para extraer un número de puntos determinado en la primera imagen de Graffitti de la base de datos de Mikolajczyk (ver figura 3.18).

Este procedimiento los hemos llevado a cabo en una máquina Intel Core 2 Duo con una CPU de 2.33 GHz y una memoria RAM de 2GB. Los resultados obtenidos se comparan con los de SIFT y SURF. Los archivos binarios de estos dos algoritmos los hemos extraído de las páginas web de los autores.

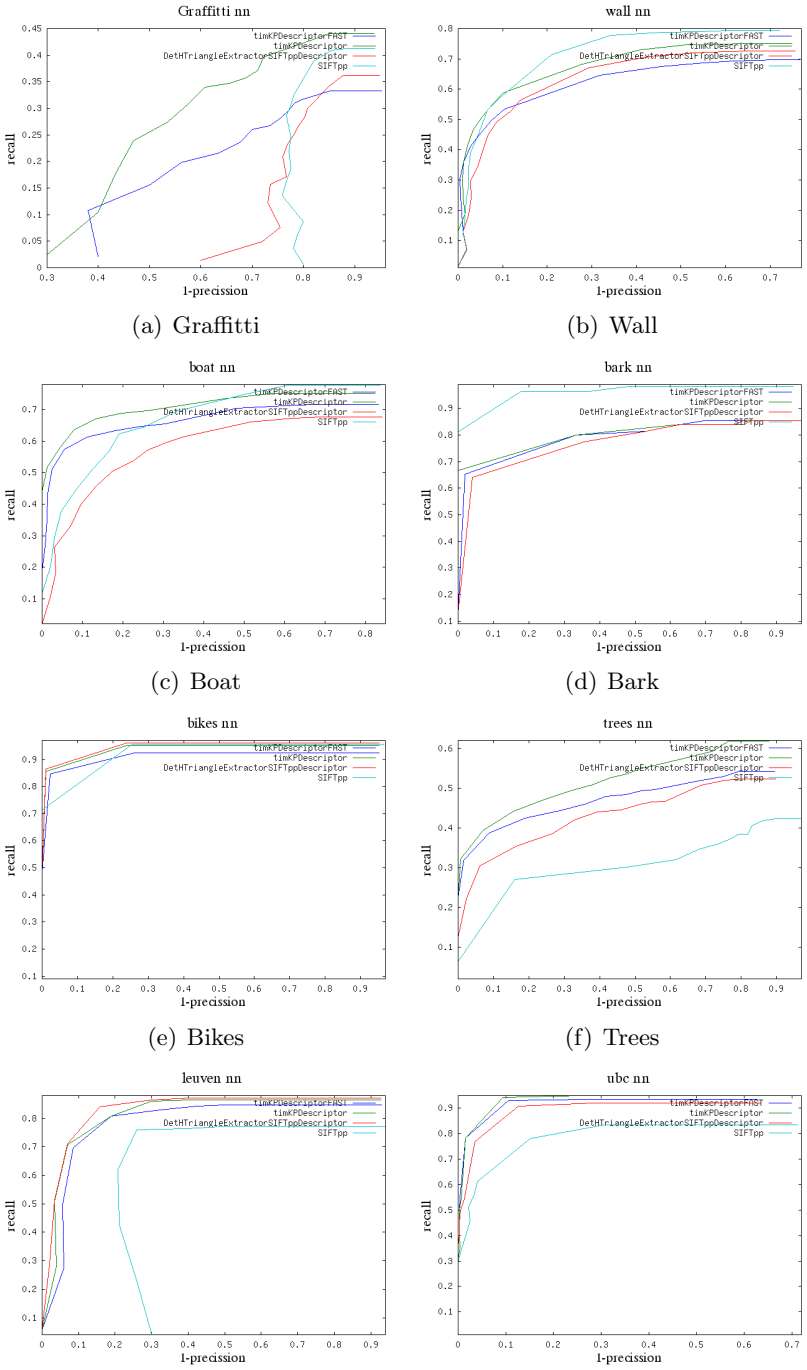
El número de puntos extraídos determina directamente el tiempo de ejecución del algoritmo. Por esta razón se han extraído un número de puntos similares a los obtenidos por SIFT y SURF, ya que de otra forma los resultados temporales no podrían compararse.

En el cuadro 5.6, se muestran los tiempos necesarios para cada uno de los algoritmos. Debe tenerse en cuenta que el tiempo mostrado incluye la carga de la imagen Graffitti (800x640), así como la escritura de los puntos extraídos, junto con sus correspondientes descriptores, en un fichero ASCII.

Método	Tamaño del descriptor	Núm de puntos	Tiempo [s]
SIFT	128	3106	3.356
DART	68	3044	0.536
SURF	64	1557	1.207
DART	68	1540	0.394

Cuadro 5.6: Tiempos requeridos para diferentes métodos de extracción de puntos característicos sobre la primera imagen de la secuencia Graffitti.

Los resultados muestran como DART se ejecuta 6 veces más rápido que SIFT y 4 veces más deprisa que SURF. Aunque en este trabajo no se ha realizado, debemos destacar la posibilidad de computar las matrices del espacio de escalas de forma paralela. De esta forma los ya pequeños tiempos de ejecución podrían reducirse más.



(a) Graffitti (b) Wall

(c) Boat (d) Bark

(e) Bikes (f) Trees

(g) Leuven (h) UBC

Figura 5.15: Gráficos con curvas exhaustividad - 1-precisión. Dos versiones del conjunto formado por el detector (filtros triangulares) y descriptor propios son comparadas con SIFTpp y una combinación del detector propio y el descriptor SIFTpp. Coincidencias obtenidas mediante vecino más cercano (ver sección 4.1.3).

Capítulo 6

Aplicaciones

En esta sección se presentan 3 aplicaciones que utilizan los puntos DART, mostrados en la sección 5.4, obteniendo un buen funcionamiento. De esta forma demostramos que los puntos DART pueden utilizarse en algunos de los problemas más comunes existentes en el campo de la visión por computador. Las 3 aplicaciones han sido creadas por miembros del área de Internet y multimedia de la empresa Telefónica I+D.

La primera de las aplicaciones utiliza DART para realizar un seguimiento sobre un objeto determinado, para a continuación sobreimponer sobre éste una figura virtual en 3 dimensiones. La superposición de elementos virtuales sobre un escenario se conoce como realidad aumentada (ver sección 6.1).

La segunda aplicación se trata de un motor de búsqueda visual, el cual es capaz de realizar búsquedas a partir de imágenes. Los puntos DART se utilizan para encontrar la imagen de una base de datos con más similitud a la imagen con la que se realiza la búsqueda. El funcionamiento de un motor de búsqueda visual está explicado con más detalle en la sección 6.2.

La última de las aplicaciones mostrada recrea un escenario 3D a partir de un vídeo filmado sobre la misma escena. Los punto DART son utilizados para llevar a cabo la reconstrucción de la escena.

6.1. Seguimiento de objetos para realidad aumentada

El seguimiento de un objeto en un vídeo se determina a partir de la posición y la postura del objeto respecto a la cámara, la cual puede permanecer estática o en movimiento. Aplicaciones como la realidad aumentada lo utilizan frecuentemente.

El caso particular mostrado en este trabajo realiza el seguimiento sobre objetos planos, es decir, objetos que tienen una superficie plana tales como cajas, libros, cubiertas de discos compactos, etc.

Utilizando objetos planos, la posición de un objeto respecto a la cámara puede ser determinada buscando las coincidencias existentes entre los puntos encontrados en una imagen de referencia del objeto a seguir y los puntos DART extraídos en los fotogramas del vídeo. Este procedimiento es conocido como "seguimiento a partir de la detección".

Para conseguir determinar la posición del objeto en el vídeo se utiliza un algoritmo que permite obtener una homografía (ver sección 3.3.1) a partir de las coincidencias de puntos DART. Una vez extraída la homografía, ésta se utiliza para obtener la posición de las cuatro esquinas del objeto plano en el fotograma correspondiente y así determinar la posición del objeto. Este proceso se repite para cada uno de los fotogramas del vídeo y de esta forma logra seguirse el objeto en cuestión. Ya que los puntos DART son invariables ante distintas transformaciones geométricas, las coincidencias entre puntos se realizan correctamente en situaciones donde el objeto aparece en diferentes ángulos, tamaños y orientaciones.

En la figura 6.1 se muestra un ejemplo visual, dónde el objeto seguido es una caja del terminal móvil iPhone de Apple. En estas imágenes pueden apreciarse las coincidencias entre puntos, unidas por líneas entre la imagen de referencia y un fotograma del vídeo. El área rectangular dibujada sobre el objeto en el fotograma indica la posición calculada a partir de las coincidencias.

El seguimiento de un objeto puede utilizarse para crear aplicaciones de realidad aumentada. La realidad aumentada es el término que define la visión de un entorno físico del mundo real, el cual se visualiza combinado con elementos virtuales, creando una realidad aumentada digitalmente en tiempo real.

En este trabajo se incluye un vídeo de demostración donde aparece la caja del iPhone sobre la que se ha realizado el seguimiento y, superpuesto sobre ésta, aparece la figura de un dragón en 3 dimensiones.[1]

[1]Los detalles del funcionamiento de la aplicación no pueden ser detallados en este trabajo por cuestiones de confidencialidad.

Figura 6.1: Seguimiento a partir de la detección utilizando puntos característicos DART para la búsqueda de coincidencias. Arriba: Imágenes sintéticas de referencia. Abajo: Diferentes fotogramas de un vídeo sobre el que se realiza el seguimiento del objeto bajo distintos ángulos y escalas.

6.2. Motor de búsqueda visual

La búsqueda visual consiste en utilizar una imagen como parámetro de entrada para obtener a partir de ella imágenes similares en contenido. Los motores de búsqueda visual obtienen correspondencias dentro de una base de datos de forma rápida y efectiva. El motor de búsqueda utilizado en este trabajo hace uso de los puntos DART para buscar la imagen con mayor semejanza a la de referencia. A continuación se explican los pasos que sigue el motor de búsqueda desarrollado por un equipo de Telefónica I+D[2] y finalmente se muestran los resultados que hemos obtenido en un experimento realizado para evaluar los resultados.

El objetivo de un motor de búsqueda visual es encontrar, en una larga base de datos, las imágenes más similares a la imagen sobre la cual se desea encontrar correspondencia. Para ello, los métodos que han resultados más exitosos utilizan la extracción de puntos característicos.

El primer paso del algoritmo extrae los puntos DART de cada una de las imágenes de la base de datos. A continuación, se obtienen los descriptores de cada uno de los puntos y se crea un diccionario de "palabras visuales" [23] [27]. Las palabras visuales son una forma de cuantificar los descriptores, obteniendo un número de palabras mucho menor al número de descriptores inicial, consiguiendo así realizar búsquedas de forma rápida. Una vez creado el diccionario, las búsquedas se realizan extrayendo los puntos DART sobre una imagen de test. A partir de los descriptores, se calculan las palabras visuales correspondientes y

[2]Los detalles del motor de búsqueda visual desarrollado en Telefónica I+D no pueden ser mostrados en este trabajo por cuestiones de confidencialidad.

con de ellas se obtiene la imagen de la base de datos con mayor similitud.

A continuación detallamos el experimento que hemos llevado a cabo para evaluar la búsqueda de imágenes en una base de datos. Primeramente se ha utilizado una base de datos formada por imágenes de productos típicos de un supermercado, tales como latas, envases, botellas, etc. Con un total de 53 imágenes, con un producto distinto en cada una de ellas. A continuación, se ha utilizado un conjunto de 121 imágenes de test, dónde en cada una de ellas aparece uno de los objetos de la base de datos capturado desde distintas distancias y puntos de vista. Por lo tanto, cada una de las imágenes de test se corresponde con una de las imágenes de la base de datos.

Mediante el motor de búsqueda visual, se ha obtenido para cada imagen de test una lista con las imágenes de la base de datos con mejores resultados en cuanto a coincidencia. Los elementos de la lista aparecen ordenados según el nivel de coincidencia, siendo la primera la imagen más semejante. Los resultados que hemos obtenido sitúan la posición media de la imagen de referencia correcta en 1.74, siendo 1 el resultado ideal.

6.3. Reconstrucción 3D

La reconstrucción en tres dimensiones de escenarios reales consiste en obtener un entorno virtual en 3D a partir de un vídeo. A continuación mostramos la reconstrucción de una escena realizada a partir de los puntos característicos DART.[3]

Para la reconstrucción de una escena, el algoritmo utiliza los puntos DART extraídos de los fotogramas de un vídeo de la escena en cuestión. A partir de la triangulación de puntos de dos o más vistas consistentes, se consigue crear un conjunto de puntos situados en un espacio 3D.

La representación geométrica de una escena a partir de un vídeo se consigue mediante la representación de estructuras a partir del movimiento y la geometría epipolar.

La representación de estructuras a partir del movimiento es el proceso con el cual se determinan estructuras tridimensionales analizando el movimiento de un objeto en el tiempo. Para ello se extraen puntos característicos en fotogramas consecutivos de un vídeo para determinar el movimiento de las estructuras de la escena.

La visión epipolar hace referencia a la geometría de la visión binocular. Cuando dos cámaras enfocan un escenario en 3 dimensiones desde diferentes posiciones, existen un número de relaciones geométricas entre los puntos 3D y sus proyecciones a los puntos 2D de las imágenes.

[3]Los detalles del funcionamiento de la aplicación no pueden ser detallados en este trabajo por cuestiones de confidencialidad.

Utilizando las tecnologías anteriores junto con los puntos DART, la reconstrucción de una escena se muestra en un ejemplo visual en la figura 6.2. El ejemplo muestra la reconstrucción parcial del monumento de la Sagrada Familia mediante puntos situados en un entorno 3D.

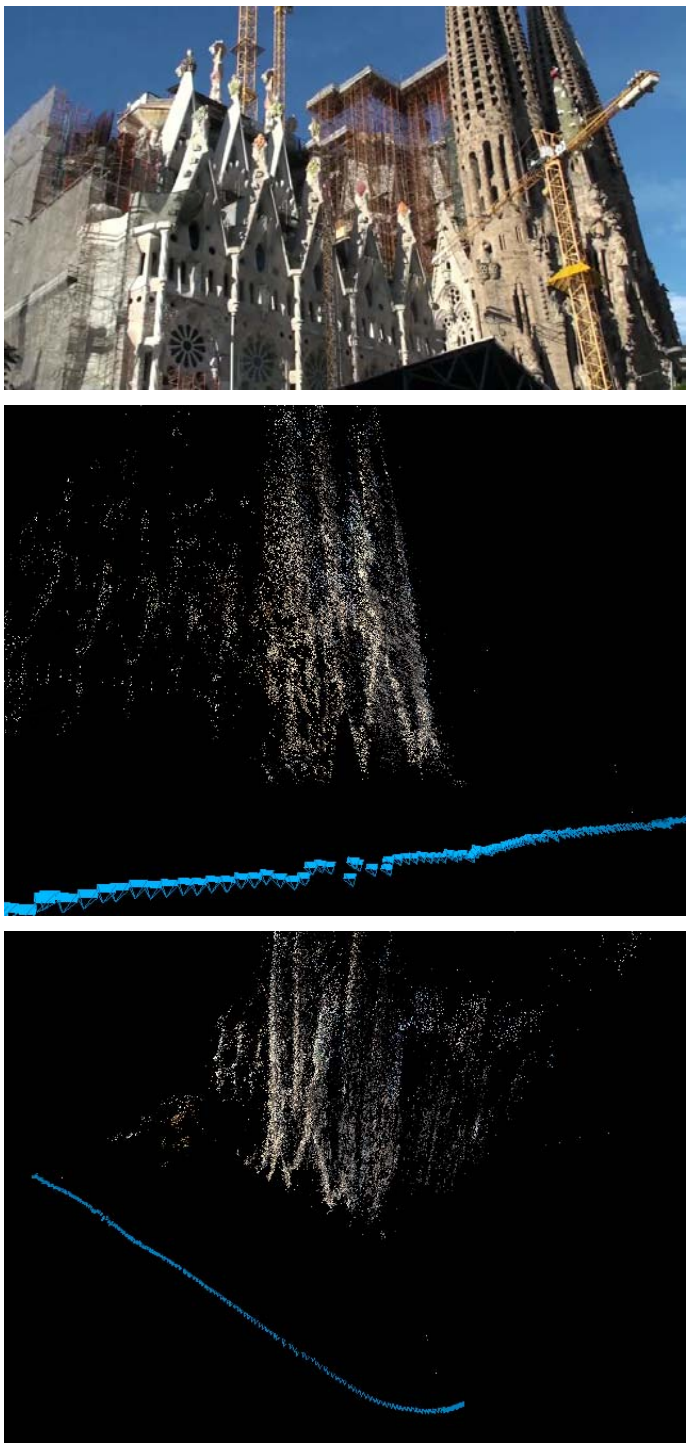


Figura 6.2: Reconstrucción 3D utilizando puntos característicos DART en una secuencia de vídeo. Izquierda: primer fotograma. Centro y derecha: conjunto de puntos 3D y localización de la cámara para cada fotograma (mostrado en azul).

Capítulo 7

Conclusiones

La realización de este trabajo ha dado como resultado el nuevo algoritmo de detección y descripción de puntos característicos DART [18], el cual obtiene mejores resultados que los métodos SIFT [14] y SURF [2], los más representativos hasta la fecha.

La utilización de filtros triangulares ha permitido obtener una aproximación del espacio de escalas Gaussiano que proporciona un detector con altos niveles de repetibilidad, así como invariabilidad ante los cambios de escala, rotación, puntos de vista e iluminación. Gracias a esta estructura ha sido posible crear un detector basado en el determinante Hessiano que requiere de pocos accesos a memoria para su ejecución.

Posteriormente se ha diseñado un descriptor basado en la distribución de componentes característicos, tal como hizo en su momento SIFT. Aprovechando el espacio de escalas creado anteriormente, se han obtenido los gradientes de forma rápida, lo cual ha hecho posible crear puntos característicos distinguibles que consiguen mejores coincidencias entre imágenes que sus competidores, según las evaluaciones realizadas. Asimismo hemos implementado un nuevo diseño para el descriptor, inspirándonos en los estudios realizados por Brown y Winder en [26], con una optimización que permite reducir considerablemente los costes computacionales.

Los objetivos por lo tanto han sido cumplidos y la aplicabilidad de los puntos DART ha sido demostrada en su utilización para el seguimiento de objetos, el motor de búsquedas visuales y la reconstrucción virtual de espacios reales en 3 dimensiones.

Bibliografía

[1] A. Baumberg. Reliable feature matching across widely separated views. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, 1:774 – 781, 2000.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Proc. European Conference in Computer Vision (ECCV)*, 110:404–417, 2006.

[3] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. *Computer Vision and Image Understanding*, 71:385–392, 1998.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[5] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891 – 906, 1991.

[6] D. Gabor. Theory of communication. *I.E.E.*, 93:429–457, 1946.

[7] L. Van Gool, T. Moons, and Ungureanu. Affine photometric invariants for planar intensity patterns. *Proceedings of the 4th European Conference on Computer Vision*, 1:642–651, 1996.

[8] P. Heckbert. Filtering by repeated integration. *Proc. Computer Graphics (SGGRAPH)*, 20:315–321, 1986.

[9] A. Johnson and M. Hebert. Object recognition by matching oriented points. *Proceedings of the Conference on Computer Vision and Pattern Recognition.*, page 684–689, 1997.

[10] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. *Proceedings of the 18th European Conference on Computer Vision*, pages 345–457, 2004.

[11] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[12] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11:283–318, 1993.

[13] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–106, 1998.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60:91–110, 2004.

[15] D. Lowe M. Brown. Invariant features from interest point groups. *In British Machine Vision Conference*, 2002.

[16] C. Harris M. Sthephens. A combined corner and edge detector. *In Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[17] D. Marimon. Method for filtering data with symmetric weighted integral images. *Solicitud de patente: US12635784*, 2009.

[18] D. Marimon, T. Adamek, A. Bonnín, and R. Gimeno. Darts: Efficient scale-space extraction of daisy keypoints. *Submitted to CVPR*, 2010.

[19] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Proceedings of the Biritish Machine Video Conference*, pages 384–393, 2002.

[20] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *Eighth IEEE International Conference*, 1:525–531, 2001.

[21] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, and et al. A comparison of affine region detectors. *Intl. Journal of Computer Vision*, 65:43–72, 2005.

[23] D.Ñister and H. Stewenius. Scalable recognition with a vocabulary tree. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:2161–2168, 2006.

[24] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark.*, 1:414–431, 2002.

[25] S.Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. *Vision Reserch*, 42:2547–54, 1997.

[26] A. Simon, J. Winder, and M. Brown. Learning local image descriptors. *International Conference on Computer Vision and Pattern Recognition*, 1:1–8, 2007.

[27] J. Sivic. Efficient visual search of images and videos. *PhD thesis at University of Oxford*, 31:591–606, 2006.

[28] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. *Proceedings of the 11th British Machine Video Conference*, 2:412–425, 2000.

[29] T. Tuytelaars and L.Van Gool. Content-based image retrieval based on local affinely invariant regions. *Int. Conf. on Visual Information Systems*, 1:493–500, 1999.

[30] T. Tuytelaars, Tinne, and L. L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal on Computer Vision*, 59:61–85, 2004.

[31] M. Villamizar and A.Sanfeliu. Scale normalized gaussian functions: Theory and application to feature extraction. 2006.

[32] S. Winder, G. Hua, and M. Brown. Picking the best daisy. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).*, 0:178–185, 2009.

[33] A. Witkin. Scale-space filtering. *8th Int. Joint Conf. Artificial Intelligence*, 2:1019–1022, 1983.

[34] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, 801:151–158, 1994.