Trabajo Fin de Grado
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Descripción de imágenes de eventos: HOG, GIST y DFT

Autor: Juan Antonio Sánchez Díaz
Tutores: José Ramiro Martínez de Dios
Raúl Tapia López

Dpto. de Ingeniería de Sistemas y Automática
Grupo de Robótica, Visión y Control
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Grado
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Descripción de imágenes de eventos: HOG, GIST y DFT

Autor:
Juan Antonio Sánchez Díaz

Tutores:
Raúl Tapia López
José Ramiro Martínez de Dios
Catedrático de Universidad

Dpto. de Ingeniería de Sistemas y Automática
Grupo de Robótica, Visión y Control
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2021

Trabajo Fin de Grado: Descripción de imágenes de eventos: HOG, GIST y DFT

Autor: Juan Antonio Sánchez Díaz

Tutores: José Ramiro Martínez de Dios
Raúl Tapia López

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia, por el apoyo constante recibido en cada paso que doy.

A mis compañeros, en especial a Ana, Iván, Paula y Gañán, porque todo esto no hubiera sido lo mismo sin ellos.

A Ramiro y Raúl, por su dedicación y confianza.

Gracias por todo

Resumen

Las cámaras de eventos aportan nuevas ventajas y posibilidades a la robótica móvil, por lo que día a día aumenta el interés en ellas. Debido a su forma de obtener información del entorno basada en interrupciones, muy diferente a cómo se ha realizado hasta la fecha, es necesario realizar un estudio y crear nuevos métodos para procesar la información que permitan aprovechar al máximo sus propiedades.

Por ello, el objetivo del proyecto es procesar los eventos obtenidos de un sensor de eventos, para conseguir descriptores que puedan obtener información temporal y espacial con las que caracterizar el entorno y el desplazamiento de un ornitóptero robótico, algo innovador y que no se ha realizado hasta la fecha. En particular, nos centraremos en el uso de tres descriptores globales utilizados en imágenes tradicionales: HOG, GIST y DFT, para los que analizaremos su funcionamiento sobre imágenes de eventos planteando modificaciones para su uso.

Abstract

Event-based cameras bring new advantages and possibilities to mobile robotics, and interest in them is growing day by day. Due to its way of obtaining information from the environment based on interruptions, very different from how it has been done to date, it is necessary to study and create new methods to process the information to take full advantage of its properties.

Therefore, the objective of the project is to process the events obtained from a event sensor, to create descriptors that can obtain temporal and spatial information with which to characterize the environment and the displacement of a robotic ornithopter, something innovative and that has not been done to date. In particular, we will focus on the use of three global descriptors used in traditional images: HOG, GIST and DFT, for which we will analyze their performance on event images, proposing modifications for their use.

Índice

ÍNDICE DE CÓDIGOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Notación

e	Número e
tan	Función tangente
arctan	Función arco tangente
sen	Función seno
cos	Función coseno
det(A)	Determinante de A
trace(A)	Traza de A
max	Valor máximo
$<$	Menor o igual
$>$	Mayor o igual
Δ	Incremento
∇	Operador Nabla

1 INTRODUCCIÓN

1.1 Cámaras de eventos

Las cámaras de eventos equivalen a una nueva forma de entender la visión espacial y percepción del entorno que nos rodea. Basadas en cómo recibe información el ojo humano, en 1991 surge este nuevo sensor, capaz de obtener información de manera **asíncrona**, otorgándole grandes ventajas frente a las cámaras tradicionales cuyo funcionamiento es síncrono y con una latencia fija.

Estas *"Silicon Retinas"* [1] funcionan a través de un DVS (*Dynamic Vision Sensor*) y proporcionan un flujo de datos conocidos como eventos. Estos eventos son producidos de manera **asíncrona e independiente** del resto de píxeles cuando alguno de ellos detecta un cambio de intensidad con respecto al último valor (almacenado por el chip) y a un cierto umbral. El dato devuelto contiene la posición en el sensor x e y, el instante en el que se produjo t y un valor booleano p llamado polaridad, que indica si el umbral se superó al alza o a la baja, obteniendo el valor 1 o 0 respectivamente.

Debido a este funcionamiento, se obtienen ventajas interesantes frente los sensores visuales tradicionales [1]:

- **Bajo consumo**: Debido al funcionamiento asíncrono y a la detección únicamente de cambios de iluminación en cada píxel, permite la eliminación de datos redundantes que consumen energía cuando no es necesario. Este tipo de ventajas son esenciales sobre todo en robótica aérea como es el caso, ya que es necesario encontrar elementos que reduzcan el consumo para aumentar la autonomía de los robots.
- **Alto rango dinámico**: Debido a que los receptores del sensor funcionan en escala logarítmica y de manera independiente, el rango dinámico de este tipo de cámaras se ha visto drásticamente aumentado, permitiéndole obtener información de escenarios con muy baja y alta iluminación simultáneamente. Todo sin tener el problema de falta o exceso de exposición que poseen las cámaras tradicionales.
- **Alta resolución temporal y baja latencia**: La independencia de cada píxel provoca una baja latencia al funcionamiento del *stream* de datos del sensor, ya que no hay que esperar un tiempo determinado para enviar la información. A esto se le añade que cada evento es detectado con una resolución de microsegundos, evitando problemas como el *"motion blur"* o desenfoque por movimiento que ocurría en las cámaras tradicionales.

Estas ventajas son interesantes para robótica, por lo que es necesario la creación de métodos de visión por computador que consigan ser eficientes y robustos en el tratamiento de eventos. Los métodos actuales están diseñados para visión síncrona y con gran densidad de información por imagen (textura, contrastes, luminosidad,…). Además, los estudios sobre ellas han permitido crear modelos del sensor, que pueden ayudar a eliminar ruido y no linealidades, modelos que no se poseen para cámaras de eventos.

1.2 Marco de realización

Con la necesidad de estudio sobre las cámaras de eventos nace este trabajo, motivado por el proyecto internacional GRIFFIN [2], iniciado en 2018 por el Grupo de Robótica, Visión y Control (GRVC). Su logotipo se muestra en la figura 1.1. El objetivo del proyecto es la creación de un robot de ala batiente capaz de volar de forma autónoma haciendo uso de visión por computador, aprovechando las ráfagas de viento de manera eficiente y teniendo capacidad de manipulación de objetos para la interaxión con humanos en búsqueda de transportar material y realizar operaciones inaccesibles para nosotros. Esto produce menor consumo que con UAVs realizados hasta la fecha, que poseen una vida útil de vuelo de unos 20 minutos.

Figura 1.1 Logotipo del proyecto GRIFFIN

Según las características citadas, el uso de una cámara de eventos abordo otorgaría al robot menor consumo, por lo que se conseguiría aumentar el tiempo de vuelo, además de, gracias a su baja latencia y a su alta resolución temporal, ser capaz de reaccionar a obstáculos de manera más precisa y rápida.

En la actualidad, existen modelos de cámaras que poseen la unión de un sensor tradicional con uno de eventos, para poder aprovechar las ventajas que ambos otorgan, realizando filtrados de las imágenes y uniéndolas para conseguir resultados más complejos y con mayor calidad. En particular, el modelo usado para la obtención de datos es una de ellas, en particular el DAVIS346, que se caracteriza por unir un sensor tradicional con un sensor basado en eventos.

1.2.1 Hardware

Por ello, para obtención de datos y realización de pruebas se ha utilizado una cámara DAVIS346, ver figura 1.2, con las características de la tabla 1.1. El flujo de datos de la cámara se ha obtenido de datasets facilitados por GRVC [2], producidos de observar dos figuras geométricas (*triangle* y *square*) realizando movimientos en diferentes dirección y sentidos, y sobre tres escenarios reales (*Testbed*, *Hills* y *Soccer*) durante pruebas de vuelo del ornitóptero.

Figura 1.2 Cámara DAVIS346

Tabla 1.1. Características de la cámara DAVIS346

Resolución sensor DVS	340x260 píxeles
Rango dinámico	120 dB
Latencia mínima	~ 20 us
Ancho de Banda	12 MEventos/s
Consumo	<180 mA @ 5V DC
Peso	100 g

1.2.2 Software

Para la implementación de los métodos se ha hecho uso de MATLAB. MATLAB [3] combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de manera matricial. En especial, se ha utilizado la librería *Image Processing Toolbox*, con la que podremos procesar imágenes. MALTAB nos permite realizar pruebas con un lenguaje sencillo para posteriormente aplicar en entornos mejor optimizados para su uso en tiempo real e industrial, como ROS, pero es poco eficiente en el tratamiento de las matrices.

1.3 Objetivos

Como se ha comentado anteriormente, el objetivo que persigue este proyecto es el desarrollo de métodos que trabajen con cámaras de eventos y consigan información del medio que resulte útil para el robot de ala batiente, permitiendo la distinción particular de cada escena.

La idea principal es realizar pruebas y obtener un vector que caracterice cada una de las imágenes creadas con eventos, haciendo uso de un descriptor global basado en métodos tradicionales de extracción de información. Por ello, siguiendo la figura 1.3, es necesario buscar técnicas que agrupen de la mejor manera posible la información del conjunto de eventos que se recibe del sensor, añadiendo estrategias de filtrado de ruido para conseguir imágenes más limpias. Tras haber generado las imágenes de eventos, se pretende aplicar transformaciones que devuelvan la descripción deseada. Todo esto, con el fin último de que dicha información pueda ser utilizados para etapas de procesamiento posteriores, como clasificación, que ayuden al ornitoptero a poseer información espacial y temporal durante su vuelo.

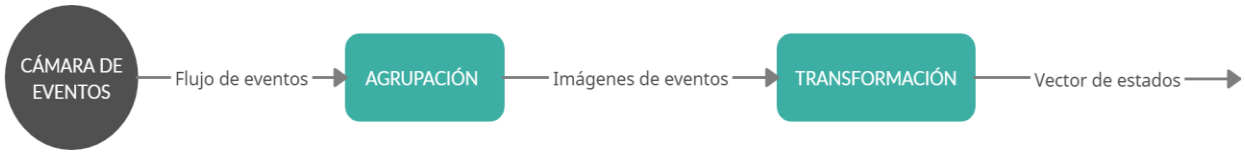


Figura 1.3 Proceso de obtención de características

1.4 Estructura del trabajo

El trabajo de fin de grado posee un total 6 capítulos. En el Capítulo 2 se muestra un estudio del estado del arte, en el que se comentan los métodos de extracción de características más utilizados, comenzando con cámaras tradicionales para poder comprender así, los métodos desarrollados en cámaras de eventos. Los Capítulos 3 y 4 presentan un desarrollo más detallado de las diferentes etápas que componen el sistema de procesamiento de eventos, comenzando por la agrupación (Capítulo 3), en la que se mostrarán diferentes métodos de empaquetado y filtrado de eventos, y terminando por el proceso de transformación (Capítulo 4), en el que se explicará el procedimiento de obtención de características Se mostrarán las diferencias entre los métodos elegidos y se utilizará pseudocódico e imágenes para facilitar su comprensión. En el Capítulo 5 se mostrarán los resultado de aplicar los diferentes métodos de obtención de características. Finalmente, en el Capítulo 6 se realizarán conclusiones sobre los resultados obtenidos para posibles mejoras y nuevos expermientos a realizar en desarrollos futuros.

2 ESTADO DEL ARTE

2.1 Introducción

En este capítulo se realizará una introducción a la obtención de características en imágenes digitales, para conocer las técnicas utilizadas hasta la fecha, con el fin de poder tener un conocimiento global sobre en qué consisten, en qué se basan y cuáles son sus aplicaciones.

En primer lugar, se describirán de manera breve los métodos más utilizados para extracción local de la información. Se continuará con una descripción detalla de algunos métodos de obtención de información global realizados hasta la fecha. De ellos partiremos para realizar modificaciones, realizar pruebas y obtener resultados, que se expondrán a lo largo de este trabajo.

En segundo lugar, se introducirán algunos de los métodos que se han comenzado a desarrollar para las cámaras de eventos, de forma que se muestre de una manera general qué ideas se han planteado para la agrupación de eventos y cómo poder extraer información de ellos.

A continuación, debido a las longitudes que pueden alcanzar los vectores de estado, se mostrarán algunos métodos de reducción de la dimensionalidad, en concreto PCA.

Para finalizar, se describirá la técnica conocida como correlación, muy utilizada en tratamiento de señal y con la que se puede extraer información de desplazamientos que existen entre una imagen con respecto a otra de referencia. Esta técnica aún no se ha utilizado en procesamiento con eventos, por lo que abre un amplio campo de investigación.

2.2 Descriptores de una imagen digital

La visión por computador son todas aquellas técnicas que aplican a una imagen y que nos permiten modificarla (filtrado de ruido, aumento de resolución, …), u obtener información de los elementos que la componen, en la que nos centraremos. Esta extracción de información tiene una gran importancia en robótica, ya que nos permite situarnos en el entorno (métodos de SLAM), reconocer formas, conocer a qué distancia se encuentra un objeto, reducir información…, es decir, nos permite otorgarle autonomía a nuestro robot utilizandolos simultaneamente con técnicas de Inteligencia Artificial.

Una imagen digital o "tradicional" se define como una representación bidimensional a partir de una matriz de números digitales, en la que cada componente corresponde con un píxel del sensor, ver figura 2.1. Dichas imágenes poseen tres canales de color RGB (*Red*, *Green & Blue*) para poder representar una gama amplia de colores, realizando superposiciones entre ellos. En función del número de píxeles, la imagen poseerá un mayor rango dinámico, por lo que se considerarán siempre en este trabajo imágenes monocromáticas en escala de grises de 8 bits, es decir, valores comprendidos en el rango [0,255] por cada píxel.

Figura 2.1 Imagen monocromática

Hasta la fecha, la obtención de información de una imagen o conjunto de imágenes se ha realizado calculando lo que se conoce como *descriptores*, propios de cada imagen y que nos permite tratarla. Se pueden separar en dos clases, *descriptores locales* y *descriptores globales*. Además, este tratamiento se puede realizar en diferentes dominios: el dominio espacial, basado en el uso directo de las intensidades de cada píxel y el dominio frecuencial, realizando cambios sobre el espectro de frecuencias de la imagen.

2.2.1 Descriptores locales

Son aquellos se basan en pequeñas regiones para la extracción de características, pudiendo obtener puntos singulares dentro de estas, puntos que permitan diferenciarlos del resto y relacionar imágenes tomadas desde posiciones distintas. Normalmente, se pretenderán buscar como puntos característicos esquínas que posean los objetos de la imagen, ya que son las regiones más distintivas de un entorno.

En imágenes tradicionales en las que poseemos un conjunto de valores de intensidad, algunos de los métodos más utilizados hacen uso de lo que se conoce como el tensor estructural **T**, una matriz que contiene información de los gradientes de una pequeña región que rodea a uno de los píxeles. Dicha matríz se deduce de la **SSD** (*Sum of Square Differences*) o suma de diferencias al cuadrado, que como la definición indica, no es más que la suma de las diferencias entre cada vecino con el píxel central. Con él se pueden realizar operaciones para conocer como de característico es cada píxel.

$$SSD_q(u,v) = \sum_{p\in\Omega}[\, f(x+u, y+v) - f(x,y)\,]^2 \qquad (2.1)$$

$$T(q) = \sum_{p\in\Omega}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \qquad (2.2)$$

Esta información se traduce en una puntuación por cada uno de los píxeles de la imagen, con el que se filtran aquellos con mayor peso que equivaldrán posiblemente a esquinas. Dependiendo de qué cálculo realicemos con dicha matríz, estaremos tratanto con un método u otro, y será necesario utilizar diferentes criterios para seleccionar los candidatos. Entre ellos destacan el detector de esquinas de Harris [4] y el detector de Noble [5], ambos basados en operaciones sobre el determinante y la traza de la matriz pero aplicados de manera distinta, ecuación 2.3 y 2.4 respectivamente.

$$Score_H = \det[T(q)] - k \cdot tr[T(q)]^2 \qquad (2.3)$$

$$Score_N = \frac{\det[T(q)]}{tr[T(q)]+\epsilon} \qquad (2.4)$$

Tener que calcular constantemente para cada imagen el tensor estructural y aplicar operaciones sobre estos tiene un alto coste computacional, por lo que puede llegar a ser un proceso lento. Por ello, se desarrollaron métodos que reducen considerablemente el coste computacional a cambio de perjudicar un poco la eficacia en la detección. Destaca el método FAST [6], que utiliza también una región entorno a un pixel, pero el objetivo es encontrar una variación en la intensidad con respecto al píxel central, de un número consecutivo de vecinos que pertenecen a una circunferencia de radio R fijada previamente, ver figura 2.2. Cada píxel solo será tratado

si se cumplen una serie de requisites previos. De esta manera, se evita aplicar la detección con aquellos píxeles que no son candidatos, disminuyendo el computo.

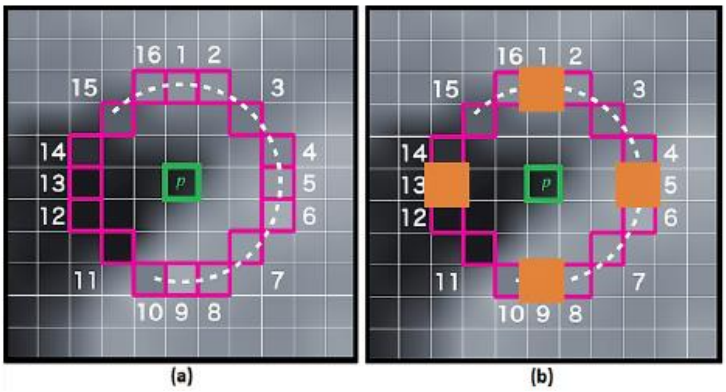


Figura 2.2 Proceso de selección de esquinas usando FAST

Debido a las pequeñas regiones que se utilizan para el cálculo en los métodos anteriores, no se obtiene una vision general de cada esquina dentro de la imagen, provocando un mal funcionamiento en cambios de escala. Por ello surgen métodos como SURF [7] o SIFT [8], que trabajan con diferentes regiones alrededor de cada pixel en busca de obtener aquellas que sean más características.

2.2.2 Descriptores globales

Su funcionamiento persigue la extracción de un único descriptor por cada imagen, que contiene información global del escenario. Estas técnicas poseen ventajas frente a los descriptores mencionados anteriormente, como por ejemplo, la reducción de memoria, aunque es necesario una buena configuración de los parámetros de cada método para obtener una buena relación entre precisión y coste computacional [9].

Algunos de los métodos más utilizados basan su funcionamiento en el uso de descriptores locales, mientras otros extraen directamente información global. Algunos de ellos se han utilizado para la creación de mapas (SLAM) [9] [10] en robots móviles con cámaras omnidireccionales tradicionales:

- **Vectores basados en PCA**. El uso de este método está relacionado directamente con la distribución de intensidades en la imagen y en un post-procesamiento con PCA. Pretende la obtención de vectores reduciendo la información más importante de cada escena con la compresión PCA, en el que la entrada sería la unión de un conjunto de imágenes, pero rotadas un ángulo determinado, como muestra la figura 2.3a. Dichas rotaciones parciales son debidas a la falta de invarianza a rotación que posee PCA. De esta forma, al proyectar sobre el mismo espacio las imágenes recopiladas durante su funcionamiento, obtendríamos un vector similar a los resultantes del conjunto previo. Dicho procesamiento implicaría mucho computo, por lo que se realizan diferentes simplificaciones utilizando la matriz de convarianza **Q** del conjunto construido, ver figura 2.3b.

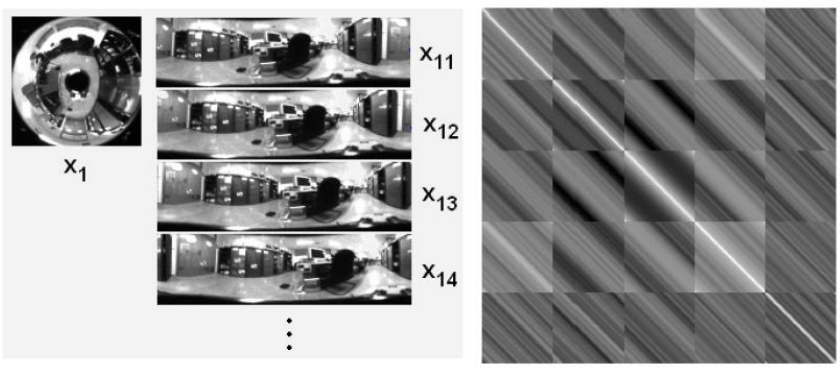


Figura 2.3 a) Conjunto de imágenes rotadas b) Conjunto de matrices de covarianza para PCA

- **Histrogram of Oriented Gradients**. Hace uso del histograma de diferentes regiones de la imagen para poder crear un vector a partir de ellos. Dicho vector será la concatenación de dos vectores generados por la aplicación del método, tanto de manera vertical como horizontal, sobre el resultado de un filtrado LP (*Low Pass*) a la imagen inicial, ver figura 2.4. Además, a dichos vectores se normalizan para evitar la influencia del factor iluminación. Para dividir la imagen se declaran de una serie de parámetros que dependen del sensor y de la carga computacional que se quiera invertir, ya que cuanta más regiones se usen, mayor tamaño tendrá el vector resultante.

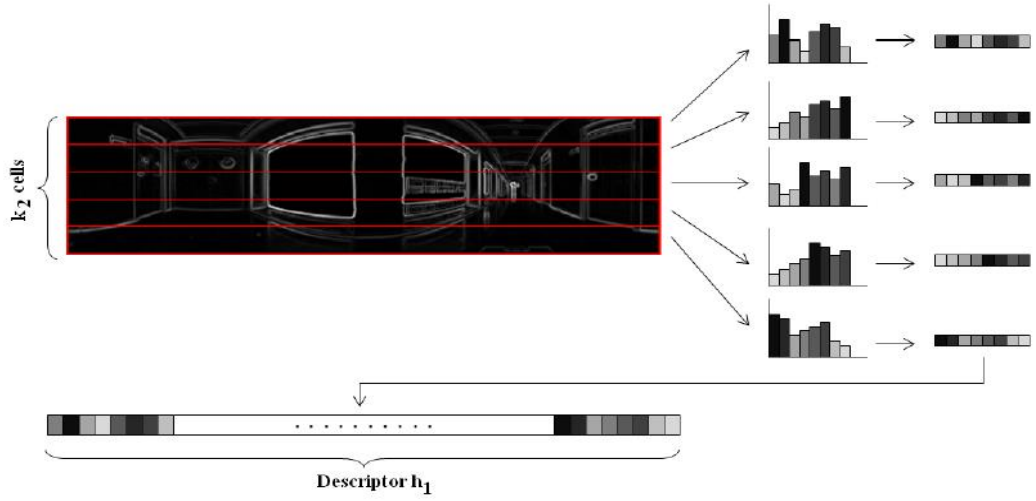


Figura 2.4 Esquemático de aplicación de HOG

- **GIST**. Su propósito es desarrollar un vector que contenga información del entorno a diferentes escalas y con la información de los bordes de la imagen desde diferentes ángulos. Para ello, se obtiene una primera versión de la imagen y luego se realiza un pirámide de imágenes como se muestra en [9], (aunque con dos componentes en esta pirámide es suficiente [10]) reduciendo la escala de la imagen a 0.5Mx0.5N (siendo M el alto de la imagen y N el ancho). Posteriormente, se le aplica un filtro de Gabor para un número discreto de ángulos equiespaciados en el intervalo [0º,180º], obteniendo ese mismo número de imágenes por nivel de la pirámide. Para finalizar y obtener el vector, se aplica las mismas regiones que el método HOG, pero almacenando la media de las intensidades que componen cada rejilla, figura 2.5.

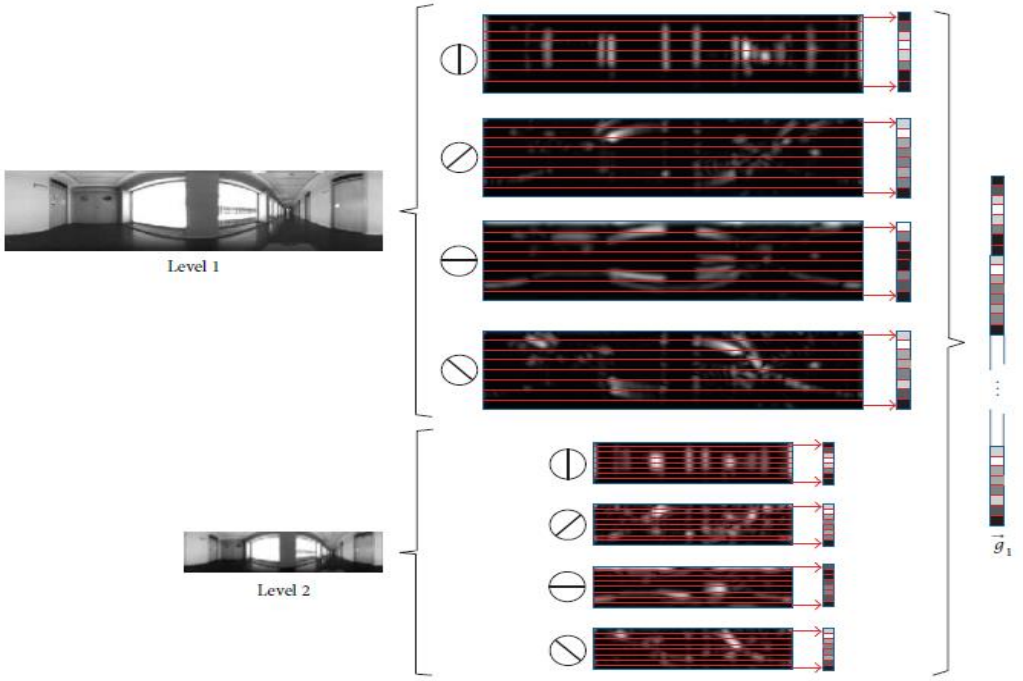


Figura 2.5 Esquemático de aplicación de GIST

- **DFT o Discrete Fourier Transform**. Es un método clásico que hace uso de la transformada de Fourier. Nos muestra características interesantes, ya que nos permite obtener información de como se distribuyen los elementos en la imagen observando la distribución en frecuencias que posee la imagen. Al aplicar la transformada de Fourier a una imagen se obtiene una distribución de números complejos que se puede descomponer en dos matrices diferentes, una que posee la magnitud de la distribución en frecuencia y otra que representa el ángulo. En ellas se almacena información acerca de la situación de las esquinas, etc, y la información de rotación y la escala, respectivamente. Gracias a esto, se puede obtener invarianza a translación y a escala, obteniendo un vector que posea la información de ambas matrices. En este caso [10], se utiliza la DFT unidimensional de las imágenes, obteniendo lo que se nombra como FS (*Fourier Signature*), que almacena la información relevante en bajas frecuencias de la magnitud, permitiendo eliminar aquellas frecuencias altas que no aportan información, reduciendo el tamaño del descriptor, ver figura 2.6.

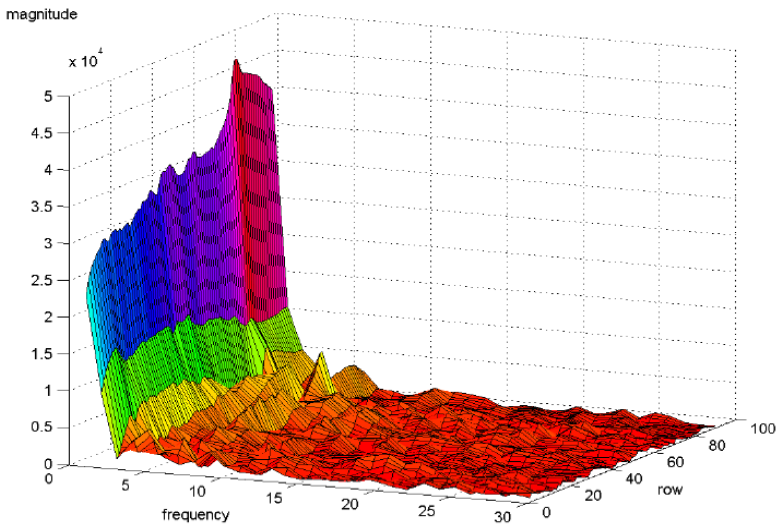


Figura 2.6 FS de una imagen

2.3 Descriptores de eventos

La obtención de descriptores sobre eventos no es un procedimiento trivial y se han desarrollado métodos para conseguir condensar la información y poder tratar con ella. Hasta la fecha para cámaras de eventos, se han adaptado algunos de los métodos locales nombrados para su procesamiento de manera síncrona y asíncrona.

2.3.1 Método síncrono

Para detectar esquinas se realizan operaciones como si de imágenes convencionales se trataran, ya que se recopilan eventos para generar imágenes "artificiales" y así poder realizar las operaciones en búsqueda de características, aplicando directamente los métodos tradicionales mencionados.

Esta recopilación de eventos se pueden hacer de diferentes maneras [1] y depende de la situación y de la aplicación. Algunos de los criterios para la creación de dichas imágenes es la selección de un número concreto de eventos a introducir por imagen, o incluir todos los eventos que pertenezcan a una ventana temporal. Estos procedimientos generan imágenes digitales binarias, en las que solo existen dos valores de intensidad diferentes. Es posible representar los eventos de diferente polaridad para añadir información de la velocidad y la dirección en la que se desplaza la cámara del entorno. Dichas imágenes binarias mostrarán la imagen de contornos de la escena, equivalente a aplicar un filtrado de paso de alta.

Los criterios de selección de eventos comentados poseen ventajas y desventajas, y no existe aún un método eficiente para dicha selección.

2.3.2 Método asíncrono

A diferencia con el caso anterior, está adaptado para el nuevo funcionamiento de la cámara, hacienda uso de lo que se conoce como **SAE** (*Surface of Active Events*) de los nuevos eventos recibidos para poder distinguir los *features* y realizar el procesamiento evento a evento. El **SAE** simplemente hace uso del instante en el que se generan los eventos para formar una representación temporal, en el que cada posición almacena el instante de tiempo en el que se produjo el ultimo evento, es decir, los pixeles con mayor intensidad se habrán generado más recientemente que aquellos con menor intensidad,ver figura 2.7.

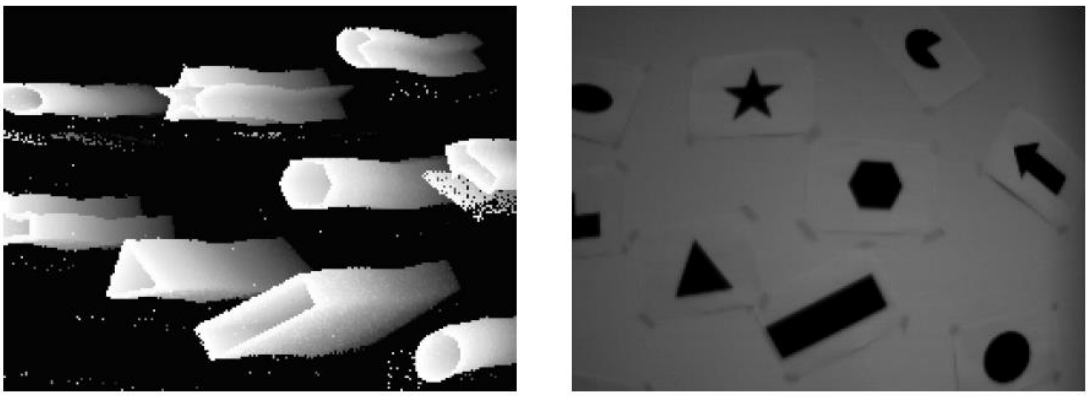


Figura 2.7 Representación del SAE

Tratar con tiempos permite realizar un filtrado de los eventos que llegan, ya que si dos eventos se producen en un intervalo de tiempo demasiado pequeño, estaremos realizando operaciones redundantes que añaden computo innecesario al sistema. Por ello, surgen métodos de filtrado como **eFilter** [11], en el que a partir del SAE, es posible detector si en la posición en la que llega un nuevo evento, la diferencia existente entre el valor temporal del nuevo evento y la almacenado dicha posición no es superior a un determinado umbral. Si dicho umbral no se supera dicho, el evento recibido no se procesará.

Analizando y operando sobre las propiedades de esta nueva representación temporal de los eventos, surgen pequeñas modificaciones de los métodos síncronos, como es el caso de **eHarris** [12]. En este método se realiza el cálculo del tensor estructural sobre una ventana de tamaño **N**x**N** binaria alrededor del nuevo evento. Dicha matríz se obtiene de la binarización de los L valores temporales más recientes. Otro método desarrollado es **eFAST** [13], o *event FAST*, que busca reconocer determinados patrones característicos que posee una esquina y que se prolongan a lo largo del tiempo, utilizando en este caso, a diferencia del método FAST síncrono, un mayor número de circunferencias, ver figura 2.8.

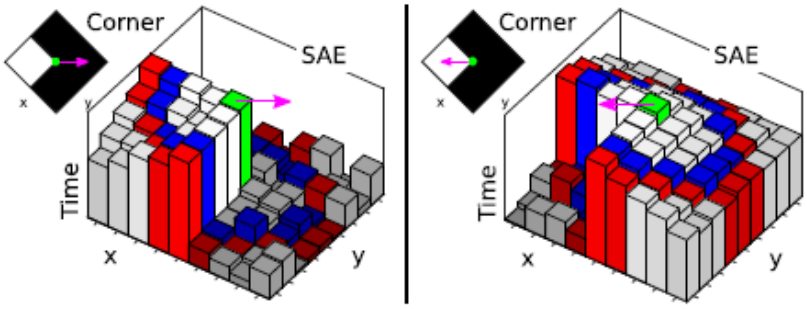


Figura 2.8 Aplicación de eFAST sobre SAE

Además, en muchas situaciones se realizan combinaciones de estos métodos, tanto de manera asíncrona como síncrona, lo que nos permite obtener mejores resultados. Esto ocurre en el caso de **FA-HARRIS** [14], en el que se utilizan dos etapas de filtrado para la detección de esquinas. La primera etapa utiliza eFAST y genera

posibles candidatos, y la segunda aplica un refinamiento a dichos posibles candidatos para filtrar de manera más exacta las esquinas reales.

La diferencia principal entre los métodos asíncronos y los síncronos, es que para el caso de los síncronos se pierde determinadas propiedades de las cámaras de eventos, ya que es necesario crear una síncronía como se realiza en los sensores tradicionales. Aún así, los métodos síncronos son más robustos y que poseen mejores resultados. Para el caso de los métodos asíncronos, son mejores computacionalmente, ya que no operan con todo el SAE, si no con pequeñas regiones alrededor del pixel donde surgió un nuevo evento. La desventaja principal es la generación de ruido, por lo que son necesarios más estudios para obtener métodos más eficientes.

2.4 PCA

El PCA o *Principal Components Analysis*, es una técnica utilizada para describir un conjunto de datos utilizando nuevas variables no correlacionadas. Los componentes se ordenan por la cantidad de varianza que describen haciendo uso de la matriz de covarianza, por lo que esta técnica es útil para reducir la dimensionalidad de un conjunto de datos. Esto último es ventajoso, por ejemplo, en aplicaciones de clasificación, donde las características de las muestras a clasificar son de gran dimensión.

Se tiene el objetivo de encontrar los vectores $\boldsymbol{u}_j$, de dimensión D, tales que maximicen las varianzas muestrales de los vectores de entrada $\boldsymbol{X}_i$ proyectados sobre las direcciones de los vectores $\boldsymbol{u}_j$. En este caso, no es necesario emplear métodos de optimización para la búsqueda de tales vectores, ya que el álgebra lineal aporta una solución eficiente, que es la base de la técnica PCA y hace que su uso sea tan popular:

- Los autovectores de la matriz de covarianza, correspondientes a los autovalores, ordenados en sentido decreciente, indican las direcciones de máxima varianza muestral del conjunto de datos inicial.

La técnica PCA suele emplearse para reducir la dimensión D del espacio original de características a un nuevo valor K. Los K autovectores se agrupan y se denominan Componentes principales.

La elección del parámetro K es crítica y depende de la aplicación. Si se quiere visualizar la dispersión del conjunto de datos original, K debe ser igual a 2 ó 3. Sin embargo, si se pretende buscar un correcto equilibrio entre la información que se mantiene y el número de autovectores que no se tienen en cuenta, una buena opción es hacer uso de lo que se conoce como *Scree Plot*. Como se observa en la Figura 2.9, en el eje horizontal se representan los Componentes Principales (autovectores) y en el eje vertical la varianza, por unidad, asociada a cada Componente principal, que como sabemos se corresponde con los autovalores.

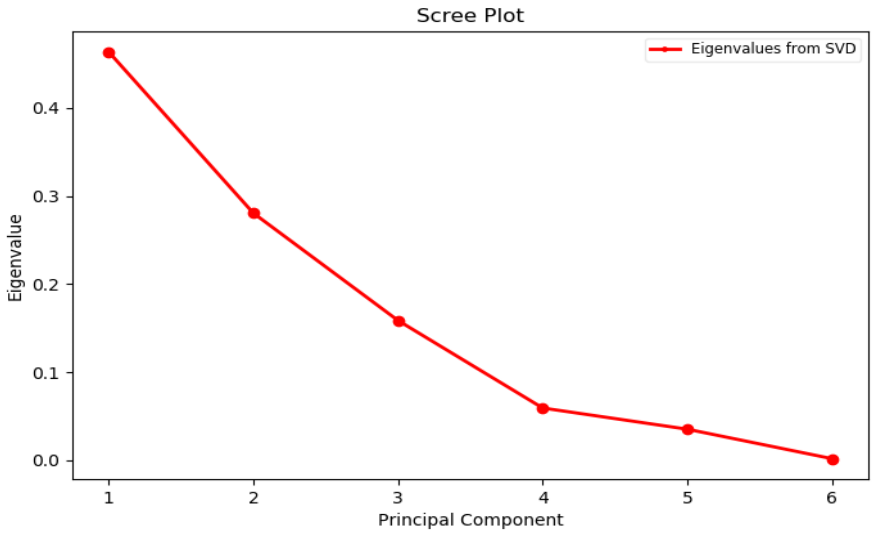


Figura 2.9 Scree plot

2.5 Correlación de imágenes

La correlación es un método que tiene un uso bastante común en procesamiento de señal, ya que nos permite conocer el desfase temporal que existe entre dos señales simplemente hacienda uso del producto de convolución entre ambas. Esta convolución nos permite comparar dichas señales, pero debido a su coste computacional y conociendo las propiedades de la transformada de Fourier, resulta más sencillo operar en el dominio de la frecuencia. De esta manera, y extrapolando a un entorno bidimensional, podemos conocer el desplazamiento que existe entre 2 imágenes.

Para ello, como se comenta en [15] tras haber calculado la DFT de las imágenes se obtiene el CPS (*Cross Power Spectrum)* de las dos señales y que aplicando la antitransformada de Fourier, producirá una matríz con el tamaño de las imágenes originales, en el que los posibles candidatos a desplazamiento tendrán almacenado un valor más alto.

Para el caso ejemplo de tener las figuras 2.10 y 2.11 desplazadas un total X = 25 píxeles e Y = 37 píxeles.

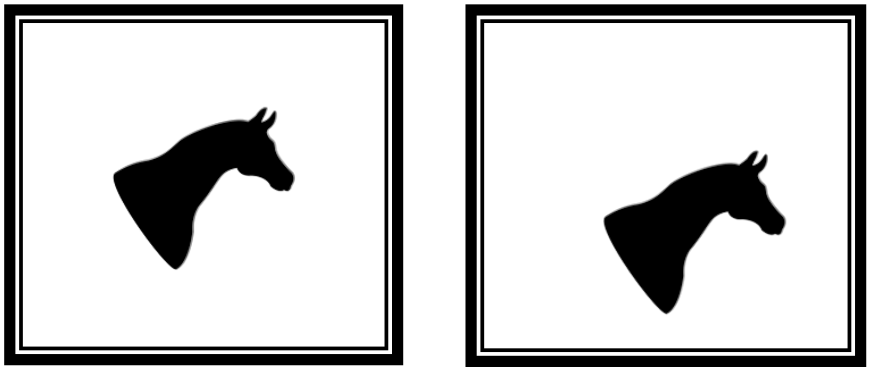


Figura 2.11 Imagen ejemplo sin traslación **Figura 2.10** Imagen ejemplo trasladada

Obtendríamos una matriz como la siguiente:

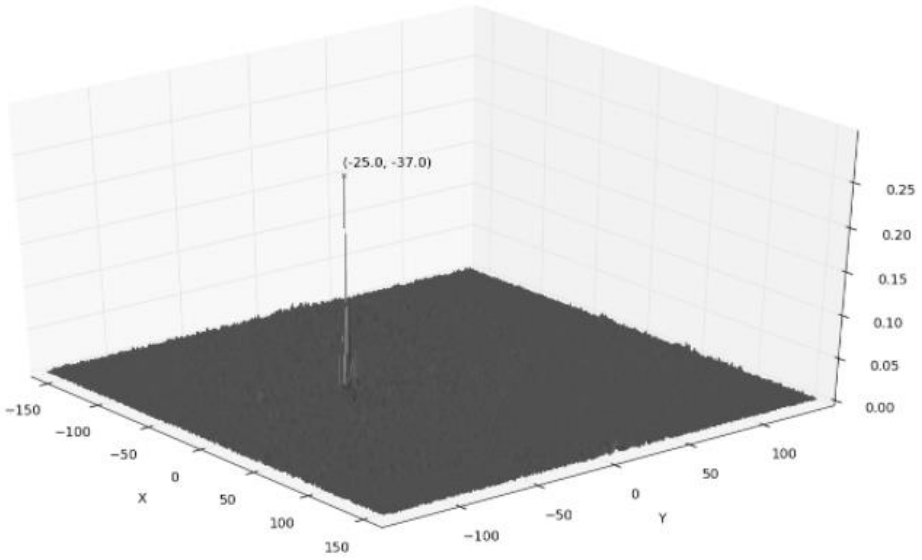


Figura 2.12 Matriz de pesos obtenida de aplicar correlación

En ella se aprecia claramente un pico en la posición que indica la traslación que existe entre las dos imágenes comparadas. Debido a la resolución de un pixel que posee la operación, puede llegar a ser impreciso, por lo que existen métodos que trabajan en el entorno de subpixel, para obtener mayor precisión, utilizando una función cuadrática [16] o matrices de pesos gaussiana [17] alrededor del valor más alto.

Este mismo método se puede aplicar para la obtención de rotación y cambios de escala, realizando diferentes operaciones de filtrado antes del procesamiento frecuencial, ya que se puede obtener información redundante. Además, se realiza un cambio, y en vez de utilizar la transformada de Fourier, se utiliza la transformada de Fourier-Mellin, que añade términos de la transformada de Mellin.

2.6 Conclusiones

Tras realizar el estudio de diferentes métodos, podemos concluir con que ya se han desarrollado a día de hoy métodos robustos sobre eventos, pero no haciendo uso de descriptores globales ni en el dominio frecuencia.

El uso de estos descriptores globales, teniendo en cuenta el carácter temporal y la polaridad de los eventos, nos pueden otorgar información tanto espacial y temporal, combinando algunos de los procedimientos comentados, llegando a se capaces de conocer, por ejemplo, en que zona es más redundante el movimiento del robot, pudiendo centrarnos en dicha zona para la obtención de mayor información, o conocer la velocidad a la que se mueven los objetos del entorno.

En este proyecto nos centraremos sobre todo en el uso de tres de los métodos comentados HOG, GIST y DFT. Sus propiedades y principios de funcionamiento pueden permitir extraer de una manera eficiente información de los eventos sobre imágenes artificiales generadas con alguno de los métodos comentados y que se desarrollarán en el presente trabajo de fin de grado. Además, debido a la naturaleza binaria y su falta de variedad en intensidades de este tipo de imágenes, los métodos elegimos permiten modificaciones para reducir el computo y etapas innecesarias.

3 AGRUPACIÓN DE EVENTOS

3.1 Introducción

En este capítulo se realizará una explicación detalla de la primera etapa de obtención de descriptores. Se agruparán los eventos recibidos por la cámara en diferentes escenarios, de manera que consiga de la mejor forma posible una representación fiel del entorno, pudiendo aplicar así métodos de descripción global. Posteriormente, se expondrán propuestas de filtrado de la agrupación o durante la agrupación, buscando eliminar el ruido introducido por el sensor.

La selección de un buen criterio en la agrupación de información es clave, de ella dependerá la precisión y los errores que se cometerán en etapas posteriores del procesamiento, pudiendo llegar a provocar la falla del proceso completo.

Todas las soluciones desarrolladas se han testeado en primer lugar en un caso más sencillo con figuras y posteriormente se ha trasladado a escenarios más complejos. Las imágenes que se mostrarán de las figuras y de los escenarios serán sin tener en cuenta la polaridad, para apreciar más claramente la forma de cada una de ellas y la distribución eventos en la imagen. Dicha propiedad se usará en algunos descriptores en el capítulo 4.

3.2 Métodos de agrupación

Como se ha comentado, un evento es un cambio de intensidad detectado en cada píxel del sensor con respecto a un cierto umbral, siguiendo la forma:

$$\Delta I(x_k, y_k, t_k) = I(x_k, y_k, t_k) - I(x_k, y_k, t_k - \Delta t_k) \quad (3.1)$$

$$\Delta I(x_k, y_k, t_k) = p_k U \quad (3.2)$$

Siendo $\Delta \boldsymbol{t_k}$ el tiempo entre el nuevo evento y el último que se produjo en dicha posición, $\boldsymbol{p_k}$ la polaridad y **U** el umbral. El valor del umbral posee intrínsecamente la cantidad de ruido que permite pasar, de forma que para valores muy bajos de **U**, se obtendrá mucho ruido, y para el caso opuesto, se perderá información real.

Esta serie de eventos provocarán lo que se conoce como flujo de eventos, que se puede representar de forma temporal como una nube de puntos que describen los objetos que aparecen en el entorno, tal y como se muestra en la figura 3.1. Se puede tratar **evento a evento** o realizando **paquetes de eventos**. Para nuestro problema en particular, desarrollaremos diferentes métodos de agrupación en paquetes que nos permitan obtener descriptores de dicha nube de puntos.

La agrupación en paquetes se basa en obtener imágenes de eventos realizando una proyección de la nube de puntos obtenidas sobre el plano de la imagen, utilizando diferentes ventanas para condensar la información. No es una tarea simple ya que existen una serie de consideraciones a tener en cuenta.

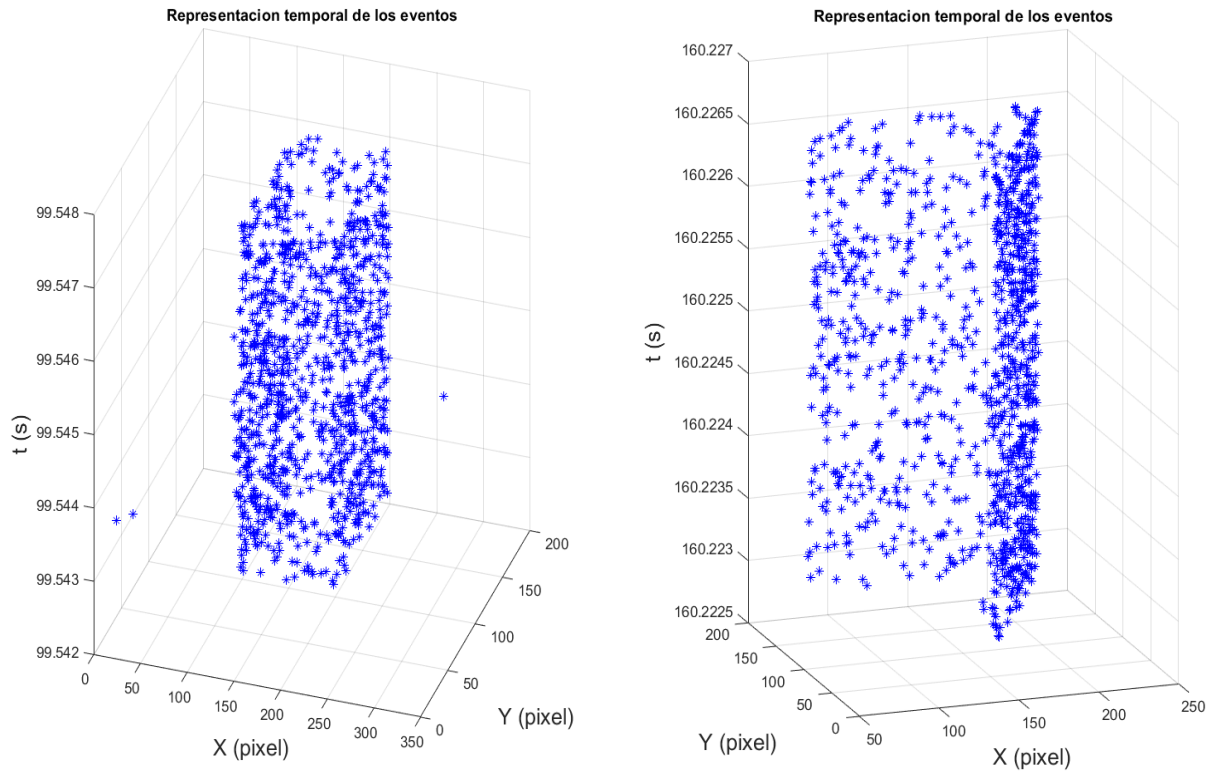


Figura 3.1 Flujo de eventos recogidos por el sensor

3.2.1 Consideraciones previas

La agrupación de dichos eventos no es trivial, ya que se incorporan diferentes problemas debido al propio funcionamiento de la cámara. Cada uno es un caso diferente y se tiene que solucionar de manera particular, por lo que es necesario realizar una serie de consideraciones:

- **Dirección de movimiento**: Debido a la detección de cambios de intensidad, si se quisiera detectar una figura, existe la posibilidad de que la cámara se desplace en la dirección de uno de sus lados. Esto provocaría que la figura en cuestión no apareciera completa. La cámara representa la imagen de contornos de la figura en la dirección de movimiento, equivalentemente al cálculo de gradiente en imágenes tradicionales. Este efecto se puede apreciar en la figura 3.2.
- **Velocidad de movimiento**: La velocidad con la que se realiza el movimiento puede producir un mayor desplazamiento espacial de los eventos, por ejemplo, para una misma franja temporal fija, los eventos aparecerán más dispersos en el marco del sensor, y será necesario reducir la ventana temporal para que el entorno sea apreciable. Por tanto, el umbral elegido dependerá de la velocidad de movimiento.
- **Elementos de la escena**: Aquellas zonas del medio en el que nos encontremos que posean mayor contraste, serán las que producirán mayor cantidad de eventos, por lo que será necesario tenerlos en cuenta.
- **Ruido**: El ruido del sensor aumenta con la cantidad de eventos que se producen, es decir, a mayor cantidad de eventos en un periodo pequeño de tiempo, mayor cantidad de ruido implícito aparecerá en el flujo de información, por lo que es necesario tenerlo en cuenta.

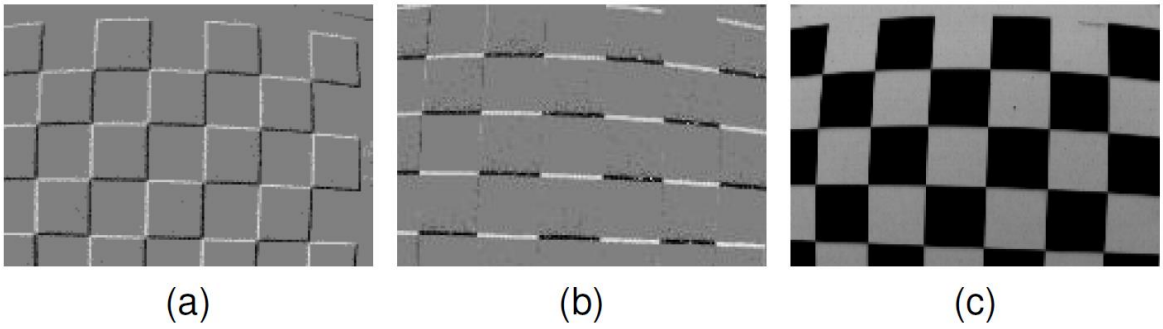


Figura 3.2 Problemas en la representación de eventos

En general, los parámetros se eligen a día de hoy experimentalmente, ya que no existe un criterio fijo para poder agrupar los eventos para cualquier situación de manera eficiente y que solucione todos los problemas. Todo depende de los factores comentados y de la aplicación.

3.2.2 Agrupación espacial

En este método se hace uso de ventanas de una cantidad de eventos fija, de forma que se van añadiendo a la imagen hasta completar una cantidad **N**. De esta forma, no se tiene en cuenta el tiempo en el que se produjeron los eventos y se pierde dicha información, pero en el caso de que haya instantes en los que no se produzcan eventos, el bloque esperará y no enviará imágenes.

Es posible que en ciertas situaciones el número de eventos elegidos sea insuficiente para la representación completa de la escena e instantes en los que existe un exceso de eventos, emborronando la imagen y haciendo difícil su compresión.

Este efecto se puede apreciar en los tres escenarios, figura 3.3, en los que para el caso de *"Testbed"* y *"Soccer"* el número de eventos es suficiente para visualización de la escena, pero que provoca distorsión en *"Hills"*.

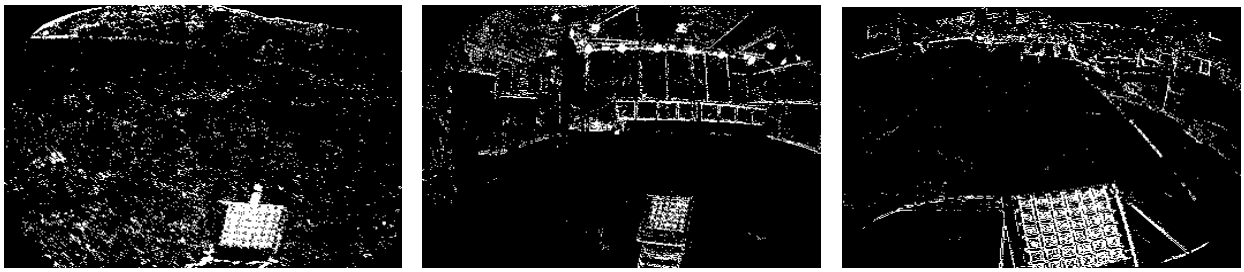


Figura 3.3 Imágenes de escenarios reales creadas con ventana espacial de 100000 eventos

Algoritmo

Código 3.1 Agrupación espacial

```
MIENTRAS contador < umbral
  Esperar evento
  Añadir evento a matriz de imagen f
FIN MIENTRAS
```

3.2.3 Agrupación temporal

Este método hace uso de ventanas temporales, de forma que se agrupan los eventos que pertenecen a un cierto intervalo de tiempo, pudiendo ser variable la cantidad de eventos por imagen. Por ello, posee más equivalencia con el procedimiento de una cámara tradicional, que tiene una frecuencia fija para obtener imágenes, pero con la ventaja de poder elegir el ratio de disparo. El problema es que pueden producirse imágenes en las que la cantidad de eventos sea insuficiente para reconocer la escena. Dicho problema es debido a una falta de movimiento relativo entre los elementos de la escena y el sensor.

Dicho efecto se puede ver en la figura 3.4, en la que la imagen de la izquierda muestra los instantes previos al inicio del movimiento, para los que el ornitóptero permanece prácticamente inmóvil, y en el que claramente se puede apreciar una falta de eventos. Esto hace imposible el reconocimiento de la escena en comparación con la imagen de la derecha, generada justo cuando comienza el vuelo.

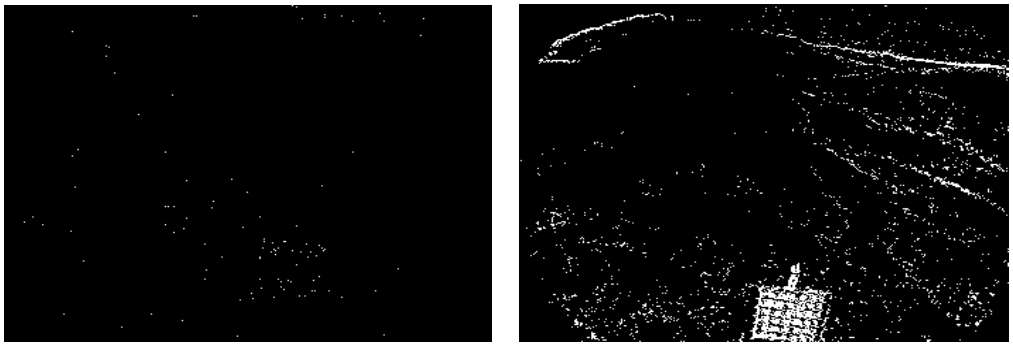


Figura 3.4 Imágenes de *"Hills"* en instantes diferentes de tiempo con ventana temporal de 200 ms

Algoritmo

Código 3.2 Agrupación temporal

```
MIENTRAS incre_t < umbral
  Esperar evento
  Añadir evento a matríz de imagen f
FIN MIENTRAS
```

3.2.4 Agrupación espacial por votos

Es una versión modificada de la agrupación espacial, y está enfocada a la detección de figuras geométricas con el fin de conseguir visualizar la figura completa, eliminando algunos de los problemas comentados como consideraciones. De esta forma, realizamos una agrupación de eventos sin tener en cuenta el tiempo transcurrido, si no que utilizamos una matriz de votaciones del tamaño de las imágenes, figura 3.5, en la que llevamos una cuenta de las veces que se repite un evento en cada píxel. Una vez que se alcance un cierto umbral, se tomará la imagen resultante de agrupar todos los eventos producidos en dicho intervalo.

Se realiza la suposición de que cuando se repita un evento en la misma posición **n** veces, es muy probable que el movimiento de la figura en el sensor haya sido suficiente para que aparezca la figura aparezca completa.

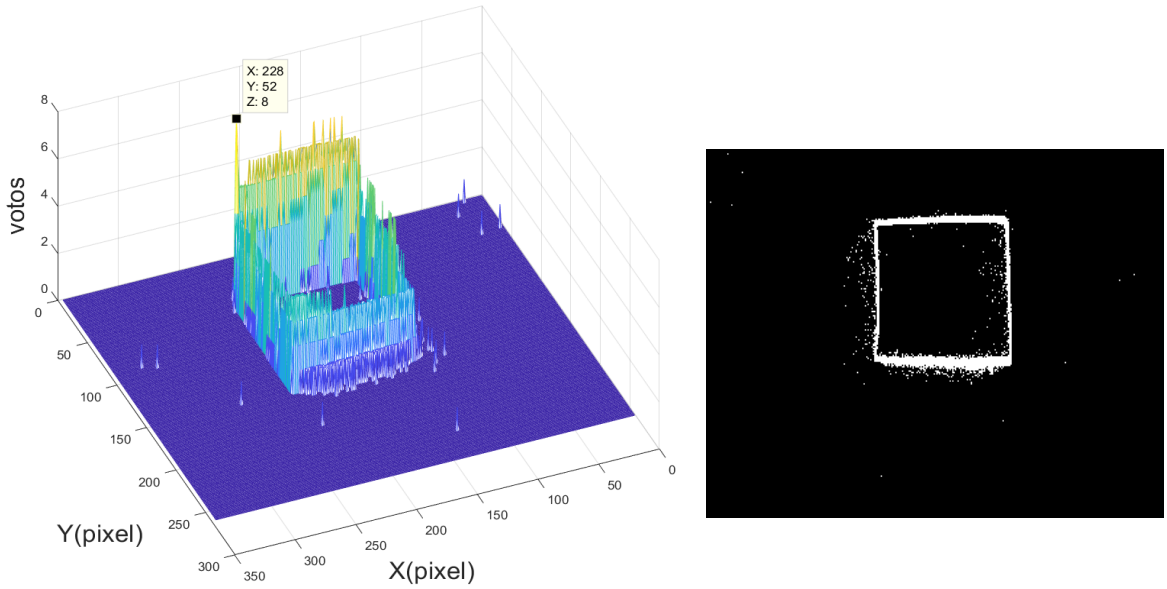


Figura 3.5 Resultado de aplicar votos en "*square*" a) Matriz de votos b) Imagen para umbral de 8

Algoritmo

Código 3.3 Agrupación espacial por votos

```
MIENTRAS max(votos) < umbral
  Esperar evento
  Añadir evento a matríz de imagen f
  Añadir voto en votos(Yevento,Xevento)
FIN MIENTRAS
```

Debido a la suposición realizada de repetición, generalmente la cantidad de eventos es muy variable en cada imagen, pudiendo provocar que se distorsione la figura con lados más gruesos de lo que realmente son, e incluso, existirán casos en los que se siga mostrando la imagen incompleta. Aún así, se reduce el número de veces en los que este último efecto sucede.

Este método de agrupación no genera buenos resultados en casos en los que existan mucha variedad de elementos, lo que puede producir que los eventos se centren en una zona en particular porque se supere el umbral y no se muestre completamente la escena.

3.3 Filtrado de eventos

Como se ha explicado, la elección del parámetro **U** que interviene en el umbral de cambio de intensidad es crítico, ya que de él dependerá la cantidad de ruido que se generará debido al propio ruido intrínseco en los componentes electrónicos que forma cada receptor. Por tanto, se considerará que el ruido es un ruido aleatorio y esporádico, es decir, se genera de manera ocasional y de forma aleatoria en la distribución de píxeles que contiene la imagen.

Dicho ruido es posible que interfiera en el correcto funcionamiento de algunos descriptores, por lo que es necesario realizar un estudio de posibles métodos de filtrado que sean capaces de reducirlos en cierta medida.

A continuación, se expondrán métodos utilizados en imágenes convencionales para mostrar su funcionamiento en imágenes de eventos y se propondrán métodos nuevos que utilizan las propiedades que ofrece la cámara, todos ellos aplicados sobre escenarios reales, ya que por lo general, todos poseen un resultado similar para casos sencillos (figuras).

En la realización de filtrado, aquellos filtros que requieren máscaras se han elegido de radio unidad. Esto se debe al alto contraste de la escena, lo que provoca un comportamiento demasiado agresivo para radios mayores.

3.3.1 Filtro de mediana

Método no lineal que aplica máscaras a cada píxel y realiza la operación mediana entre todos los elementos que pertenecen a dicha matriz. Utilizado generalmente en imágenes tradicionales para eliminar ruido *"salt & pepper"* (sal y pimienta) que posee relación con el tipo de ruido que obtenemos al generar imágenes de eventos, ver figura 3.6.

Para el caso de la tabla 3.1, el valor central sería el pixel a analizar. Para ello, se calcula la mediana de los valores que pertenecen a dicha matriz y se sustituye por el píxel analizado. En este caso particular, se sustituiría por 0.

Tabla 3.1 Matriz de ejemplo

0	0	0
0	255	0
0	255	0

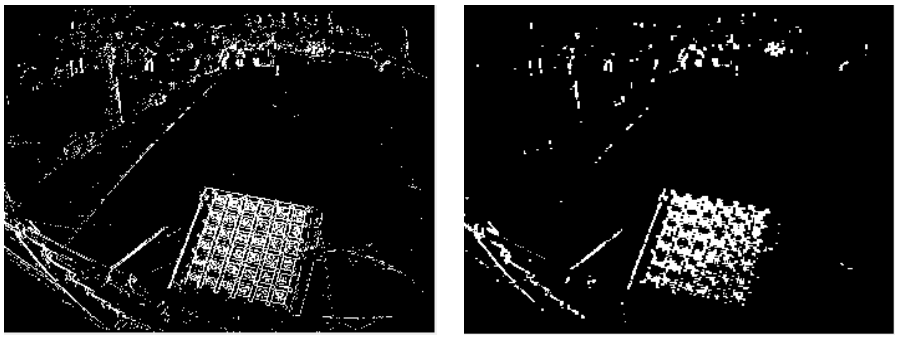


Figura 3.6 Resultado de aplicar filtro de mediana en *"Soccer"* a) Imagen original b) Imagen filtrada

3.3.2 Suavizado gaussiano

Método lineal basado en el uso de una máscara con distribución gaussiana para realizar una suma ponderada del entorno de vencidad del píxel a tratar. Se utiliza en imágenes tradicionales para "suavizar" el ruido, pero si se incrementa demasiado el tamaño de la máscara provoca difuminado y por tanto, pérdida de detalles.

Una distribución gaussiana no es más que una distribución de probabilidad, es decir, la suma de todos los valores tiene que ser igual a uno. Posee forma simétrica y acampanada, ver figura 3.7, por lo que los pesos más altos se encontrarán en el centro de la distribución (media **μ**) e irán bajando a medida que nos alejamos de dicho valor. Dicha pendiente dependerá de la desviación típica **σ,** siguiendo la ecuación 3.3. Así, aumentamos la importancia de los valores centrales de dicha matriz.

$$w(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (3.3)$$

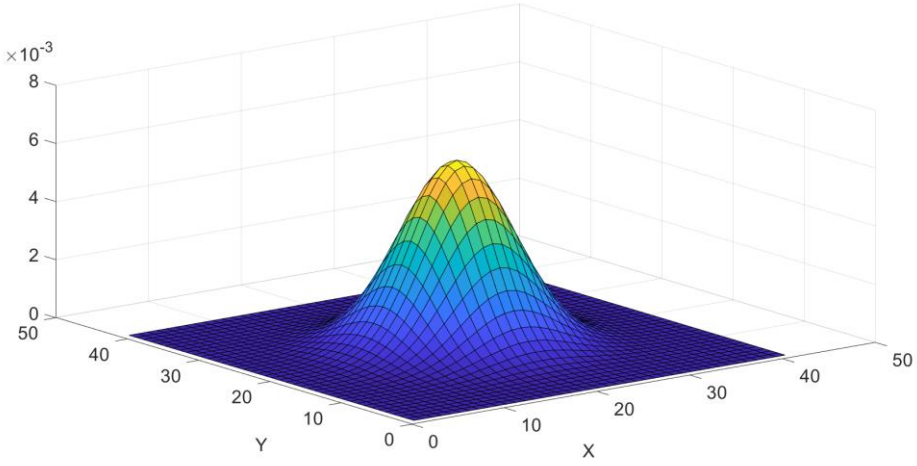


Figura 3.7 Distribución gaussiana

Su uso en imágenes de eventos se ha extrapolado a un funcionamiento binario, utilizando un umbral para seleccionar aquellos píxeles que posean valores más altos tras un primer filtrado gaussiano, de forma que se evite el difuminado de la imagen. El resultado se puede apreciar en la figura 3.8.

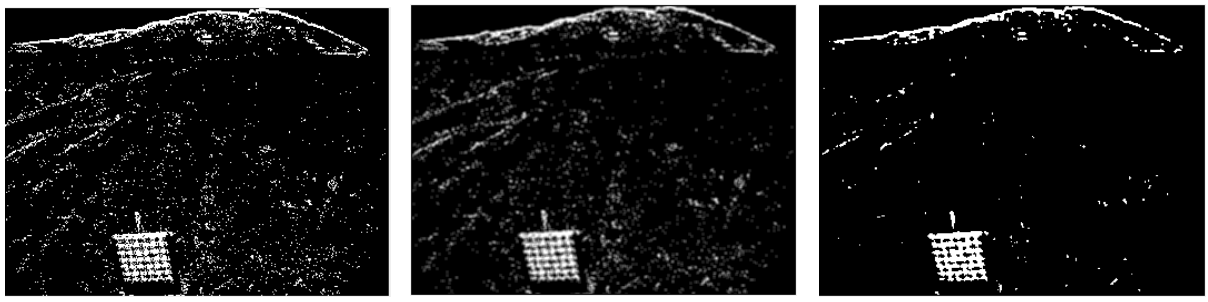


Figura 3.8 Resultado de aplicar filtro Gaussiano en *"Hills"* para umbral = 127 y máscara de radio unidad a) Imagen original b) Imagen filtrada c) Imagen umbralizada

3.3.3 Erosionado

Método morfológico utilizado en imágenes tradicionales para aplicación de métodos no lineales. En particular, el erosionado sustituye el valor del píxel tratado por el menor de todos los vecinos que pertenecen a una cierta máscara (normalmente circular). Eso provoca un proceso de "erosión" de los contornos de las figuras que aparezcan que permite eliminar puntos espúreos. Normalmente, se utiliza en conjunto con otros métodos sobre plantillas binarias.

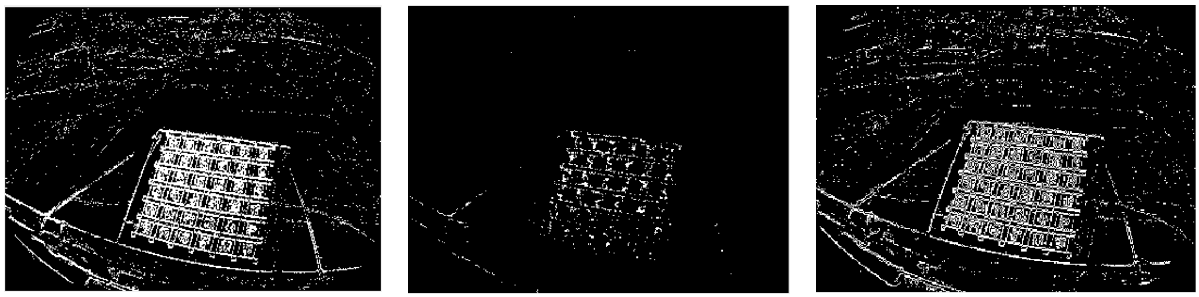


Figura 3.9 Resultado de filtrado espacio-temporal en "*Soccer"* para máscara circular de radio 1 a) Imagen original b) Imagen filtrada c) Ruido

3.3.4 Filtrado temporal

Unas de las propiedades de las cámaras de eventos es la capacidad de conocer los instantes de tiempos en los que se produjo cada evento, y que es interesante para la obtención de información temporal.

En ello se basa los métodos que se expondrán a continuación. Estos utilizan el **SAE** en busca de conocer aquellos lugares donde se han producido eventos con anterioridad de forma que, se pueda reconocer el ruido y eliminarlo de la imagen. En ambos métodos, se realizará una inicialización previa del **SAE** de forma que se tengan suficiente valores para poder comenzar a aplicar los métodos de filtrados y no se produzcan errores.

3.3.4.1 Suavizado Gaussiano temporal

Conociendo el funcionamiento del filtro gaussiano, es posible realizar un suavizado temporal considerando el entorno de vecindad temporal de cada pixel y, posteriormente, aplicar un umbral para representar exclusivamente aquellos puntos que poseen temporalmente un valor alto, obteniendo una imagen binaria a partir del **SAE**.

Para la realización de dicho **SAE**, se añadirán un número fijo de eventos por iteración y se considerará una ventana temporal para realizar los cálculos, de forma que no se tenga en cuenta aquellos puntos con demasiada antigüedad y que pueden producir emborronado y errores.

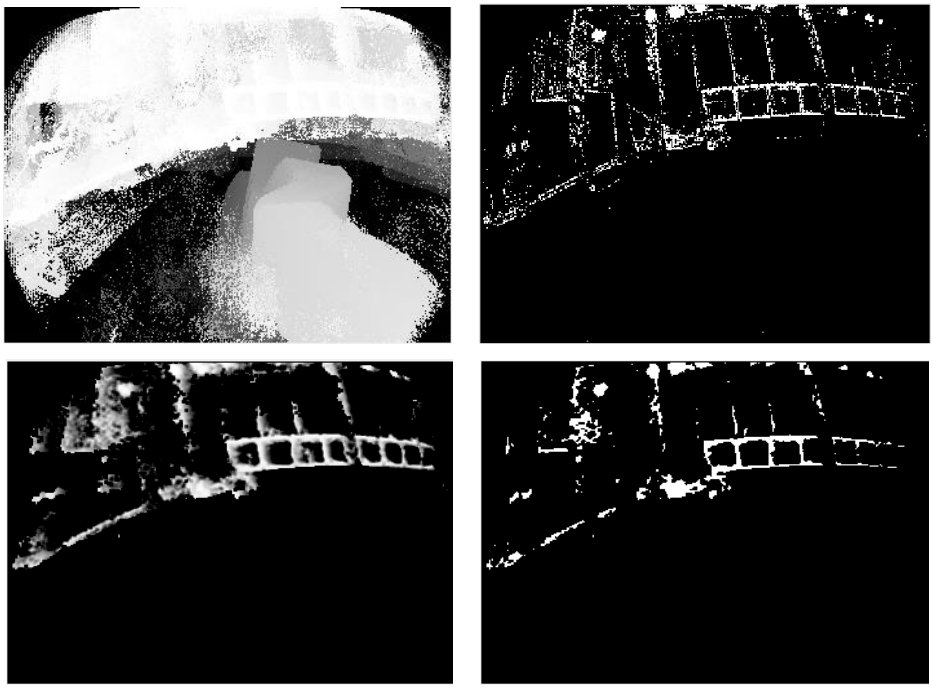


Figura 3.10 Resultado de filtrado espacio-temporal en "*Testbed"* para máscara circular de radio 1 a) SAE b) Imagen original c) SAE parcial suavizado d) SAE umbralizada

La imagen resultante como se puede apreciar en la figura 3.10, posee un difuminado de los elementos debido al suavizado, perdiendo detalles, pero consiguiendo eliminar el ruido.

3.3.4.2 Correlación espacio-temporal

Teniendo en cuenta la aleatoriedad del ruido, es posible realizar la consideración temporal de que, aquellos eventos generados por ruído, poseerán una distancia temporal mayor entre sus vecinos que aquellos que se generan realmente de manera correcta, es decir, aquellos eventos generados por movimiento real en el sensor poseerán valores temporales más cercanos [18]. Por ello, se realiza la comparación:

$$t_e - t_{ant} < d \quad (3.4)$$

En el que $\boldsymbol{t_e}$ es el instante de tiempo del nuevo evento, $\boldsymbol{t_{ant}}$ es el instante de tiempo del vecino más reciente y **d** es el umbral temporal elegido para realizar el filtrado. El entorno tomado será una ventana 3x3 del **SAE** construido hasta el momento y centrada en la posición $(\boldsymbol{x}, \boldsymbol{y})$ perteneciente al nuevo evento entrante. Los resultados obtenidos, figuras 3.11 y 3.12, son buenos tanto para el caso de figuras y escenarios

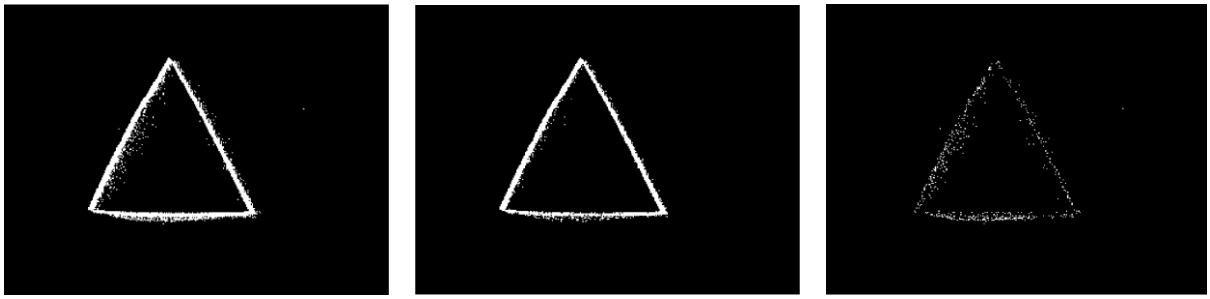


Figura 3.11 Resultado de filtrado espacio-temporal en "*triangle"* para d = 1ms a) Imagen original b) Imagen filtrada c) Ruido

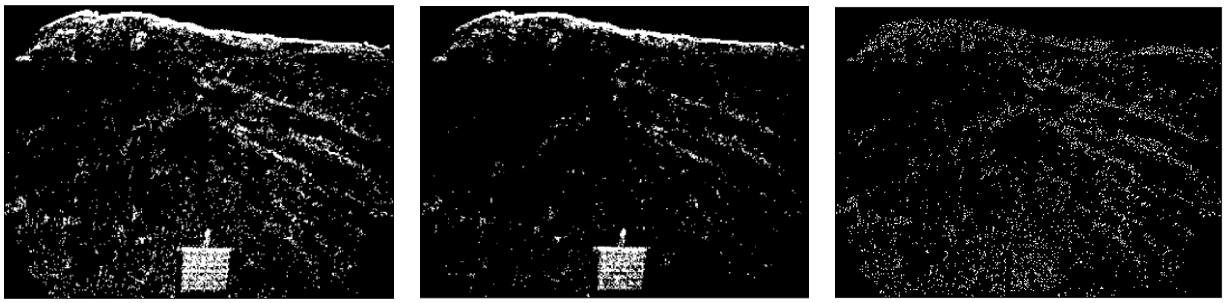


Figura 3.12 Resultado de filtrado espacio-temporal en "*Hills"* para d = 50 ms a) Imagen original b) Imagen filtrada c) Ruido

Algoritmo

Código 3.4 Correlación espacio-temporal

```
Esperar evento
Comprobamos distancia temporal con sus 8 vecinos
SI evento cumple distancia
  evento es valido
FIN SI
```

3.3.5 Filtrado por votos

Observando las propiedades y el funcionamiento de asociar zonas de la imagen con cantidad de eventos como se ha utilizado en la agrupación por votos, es posible extrapolar su funcionamiento como filtro.

Para ello, es necesario generar una imagen por el método espacial, de forma que se posea una cantidad de eventos constantes y fija en cada imagen, De esta manera, se obtendrá una representación de la imagen como ocurría con cuadrados y triángulos, pero en el que el máximo es variable en cada instante y dependerá de la escena, figura 3.13.

La idea es considerar que todos aquellos puntos que hayan generado únicamente un voto son, por tanto, ruido producido por el sensor y que es necesario eliminar ya que no aporta información útil al conjunto. La aplicación del método permite reducir la cantidad de eventos aproximadamente en un 20% del total de los eventos iniciales. Además, de esta manera, representamos realmente la información más relevante de la escena que se mostrará independientemente del ángulo de visión de la cámara. Los resultados se pueden ver en la figura 3.14.

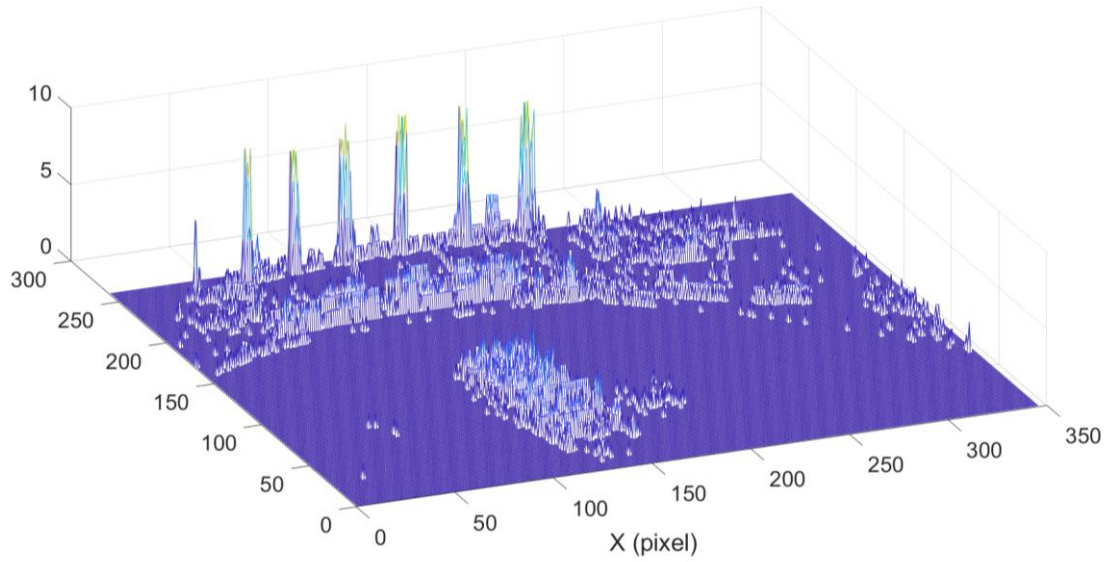


Figura 3.13 Matriz de votos para filtrado

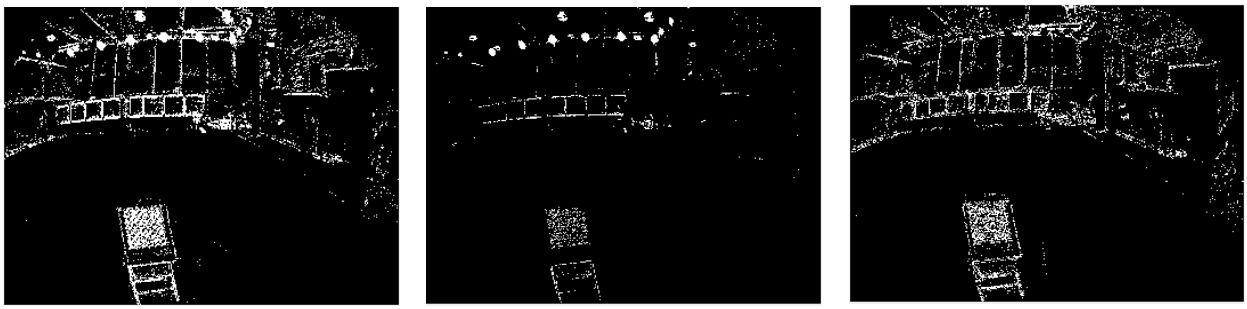


Figura 3.14 Resultado de filtrado por votos en "*Testbed"* a) Imagen original b) Imagen filtrada c) Ruido

3.4 Conclusiones

En cuanto a la agrupación de eventos, las tres soluciones citadas consiguen su objetivo de mostrar la escena en una imagen digital binaria, siendo capaces de distinguir a simple vista objetos y figuras dentro de ellas. Por el contrario, ninguno de ellos consigue solucionar todos los problemas comentados y cada uno tiene ventajas y desventajas frente a los otros.

En nuestro caso particular, se ha optado por utilizar la ventana espacial, debido a que el funcionamiento de los descriptores globales hace necesario que las imágenes a tratar aparezcan completas, cosa que no se puede conseguir siempre usando ventanas temporales. Los umbrales utilizados para la generación han sido de 5000 eventos para cuadrados y triángulos, y 10000 eventos para escenarios reales.

Finalmente, con respecto al filtrado de ruido se ha realizado una tabla comparativa en la que se muestran los tiempos de ejecución. Se ha aplicado cada método a las 100 primeras imágenes obtenidas de *"square"* por el método de agrupación espacial con ventana de 6000 eventos, y se ha realizado la media del conjunto.

Analizando la tabla 3.2, podemos concluir que los métodos más eficientes son la erosión, el filtrado espacio-temporal y el filtrado por votación. Aún así, debido a su funcionamiento tan agresivo, se descarta el uso de erosionado, restringiendo su uso para aplicaciones específicas. Por ello, los métodos tradicionales utilizados para imágenes de eventos no son interesantes, debido a sus altos tiempos de ejecución y sus resultados que equivalen o empeoran a los adaptados a cámaras de eventos.

Tabla 3.2 Comparación de tiempos de ejecución

Método de filtrado	**Tiempo de ejecución (ms)**
Mediana	779.111224
Gaussiana espacial	150.732383
Erosión	2.070457
Gaussiano temporal	155.063792
Espacio-temporal	10.529855
Votos	0.620246

4 DESCRIPCIÓN DE IMÁGENES

4.1 Introducción

En este capítulo se expondrán los tres métodos de descripción global HOG, GIST y DFT utilizados en imágenes de eventos basados en estudios realizados sobre imágenes tradicionales [9], proponiendo modificaciones para su adaptación a las nuevas cámaras de eventos.

Los descriptores globales surgieron con la necesidad de obtener información de una imagen completa y poder clasificar elementos que se encuentren en su interior. Nace como solución a la falta de información global que aportaban los descriptores locales desarrollados hasta la fecha. Muchos de los utilizados hoy en día son, de hecho, la unión de un conjunto de descriptores locales concatenados y con pequeñas modificaciones para conseguir cierta robustez.

El objetivo es conseguir extraer información espacio-temporal que resulte útil para una etapa de procesamiento posterior, como se desarrollará en el TFG de Francisco Javier Gañán Onieva en el que, utilizando la DFT, se realizará una clasificación de diferentes escenas.

Se mostrarán las bases teóricas y las propiedades de cada uno de ellos, finalizando con unas conclusiones donde se comentarán aspectos a destacar.

4.2 HOG

HOG o *Histogram of Oriented Gradients* [19] basa su funcionamiento en la descripción de la imagen utilizando el gradiente, aprovechando de manera eficiente dicha información con la división la matriz de gradientes en celdas de pequeño tamaño distribuidas de manera equiespaciada. De esta manera se obtienen histogramas locales que proporcionan las direcciones de los contornos y que permiten la detección y distinción de objetos.

Dichos histogramas locales, se combinan para obtener un vector de características que describa de manera global la imagen de contornos tanto en magnitud, como en orientación.

4.2.1 Cálculo de gradiente

La primera etapa del procesamiento, como se ha comentado, comienza con el cálculo del gradiente de la imagen. El gradiente no es más que un cambio de intensidad con la dirección donde dicho cambio es máximo, poseyendo, por tanto, carácter vectorial y dos componentes esenciales, la magnitud del cambio y la dirección.

Para calcular dicho gradiente se obtienen en primer lugar sus componentes cartesianas $\boldsymbol{G_x}$ y $\boldsymbol{G_y}$, que no son más que los cambios de intensidad en las direcciones de los ejes cartesianos que definen la imagen. De esta forma, se obtienen la dirección $\boldsymbol{A_{ij}}$ y el módulo $\boldsymbol{G_{ij}}$ mediante operaciones con dichas proyecciones (ecuaciones 4.1, 4.2 y 4.3).

$$\begin{cases} Gx = I(i+1,j) - I(i-1,j) \\ Gy = I(i,j+1) - I(i,j-1) \end{cases} \quad (4.1)$$

$$A_{ij} = \arctan\frac{Gy}{Gx} \quad (4.2)$$

$$G_{ij} = G_x + G_Y \quad (4.3)$$

En la práctica, el cálculo del módulo se realiza aproximándose suma de ambas componentes como se muestra en la ecuación 4.3, reduciendo el computo.

Dicha operación se realiza sobre todos los píxeles de la imagen, utilizando una serie de máscaras de tamaños variables y la operación convolución. Algunas de las máscaras más conocidas son las de *Sobel* y *Prewitt*, aunque para nuestro caso práctico utilizaremos una máscara sencilla como la mostrada en la tabla 4.1, con su equivalente traspuesta para el cálculo de la dirección X.

Tabla 4.1 Máscara para el cálculo de gradiente en dirección Y

0	-1	0
0	0	0
0	1	0

4.2.2 Cálculo de histograma

Una vez obtenida la matriz de gradientes, comienza su división en celdas para realizar el cálculo de histogramas locales. La elección del tamaño de estas celdas es variable, aunque según el estudio que se sigue en [19] el uso de ventanas de tamaño 6x6 son las que mejores resultados otorgan, por lo que será el tamaño elegido, ver figura 4.1.

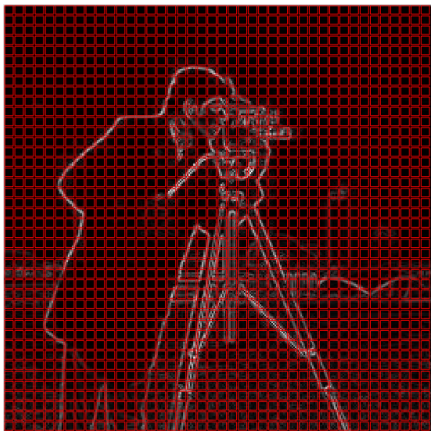


Figura 4.1 Imagen de contornos dividida en celdas de 6x6

El rango de ángulos a utilizar también es un parámetro variable para el cálculo de los histogramas, aunque un valor típico es discretizar en 9 valores el rango [0º, 180º], es decir, dividimos la orientación en 9 fragmentos de 20º sin considerar signo, como se puede ver en la figura 4.2.

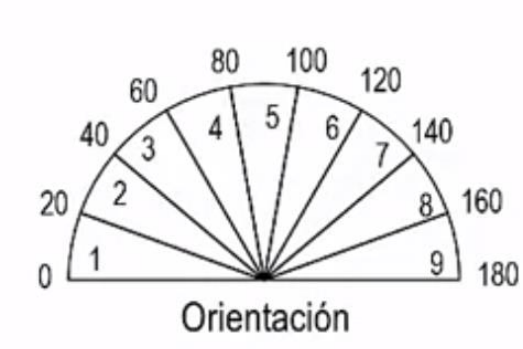


Figura 4.2 Discretización de orientaciones para HOG

Para lo que finalmente se calcula el histograma, ecuación 4.4, que no es más que la suma acumulada de los valores de magnitud en cada ángulo discreto. La longitud total de cada histograma será igual al número total de intervalos elegido:

$$h(k) = \sum_{(x,y)\in C} w_k(x,y) \cdot G(x,y) \quad (4.4)$$

Siendo **C** la celda tomada, **k** cada uno de los intervalos discretos y $\boldsymbol{w_k}$ un peso que depende de la dirección que posea cada elemento, siendo 1 cuando la dirección de dicho punto pertenece al intervalo **k** y 0 en caso contrario.

4.2.3 Consideraciones

Este procedimiento de obtención de histogramas es simple, provocando que los diferentes gradientes solo intervengan al intervalo y a la celda a la que pertenecen. Dicha simpleza puede provocar que pequeñas variaciones en la imagen original produzcan variaciones significativas en el descriptor final.

Para solucionar este problema se realiza una modificación de los pesos $\boldsymbol{w_k}$ siguiendo la ecuación 4.5 y añadiendo los pesos $\boldsymbol{w_x}$ y $\boldsymbol{w_y}$, ecuaciones 4.6 y 4.7, consiguiendo que cada elemento de gradiente pueda intervenir en más de una celda y en más de un intervalo diferente, y por lo tanto, consiguiendo robustez.

$$w_k = \max\left(0, 1 - \frac{|\theta(x,y) - \theta_k|}{\delta\theta}\right) \quad (4.5)$$

$$w_{ij}^x(x,y) = \max\left(0, 1 - \frac{d_{ij}^x}{\delta x}\right) \quad (4.6)$$

$$w_{ij}^y(x,y) = \max\left(0, 1 - \frac{d_{ij}^y}{\delta y}\right) \quad (4.7)$$

Siendo $\boldsymbol{\delta x}$ y $\boldsymbol{\delta y}$ las distancias entre celdas horizontales y verticales respectivamente, y $\boldsymbol{d_{ij}^x}$ y $\boldsymbol{d_{ij}^y}$ la distancia del píxel (x, y) a la celda (i, j), ver figura 4.3.

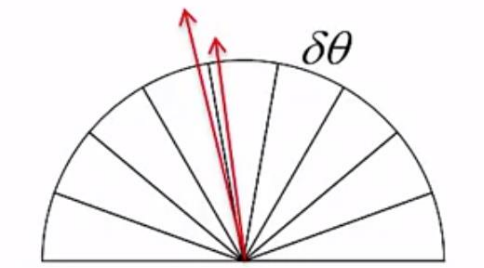


Figura 4.3 Influencia de orientaciones próximas a rangos discretos

4.2.4 Cálculo de descriptor

Con el objetivo de obtener el mayor número de invarianzas posibles, se realiza una normalización de los histogramas locales usando bloques. De esta manera se consiguen invarianzas a cambios en la iluminación de la escena. Estos bloques son agrupaciones de un número de celdas fijo de tamaño bxb, siendo **b** típicamente 2.

La normalización consiste en concatenar los histogramas pertenecientes a cada bloque y dividir entre el módulo del vector resultante:

$$v' = \frac{v}{|v| + \varepsilon} \quad (4.8)$$

La constante $\boldsymbol{\varepsilon}$ simplemente se añade para evitar las divisiones por 0 que pueden existir en bloques donde las intensidades sean constantes y por tanto, la magnitud del vector sea nula.

Dichos vectores finalmente se concatenan por bloques, realizando el cálculo de cada uno de ellos de manera que exista un solapamiento de la información (celdas), que ayudará a la robustez del descriptor final. Este solapamiento se presentará de manera diferente debido a la normalización independiente de cada bloque.

4.2.5 Adaptación a imágenes de eventos

Como se ha comentado en capítulos anteriores, las imágenes de eventos obtenidas son directamente imágenes de contorno de la escena en cuestión, por lo que para conocer la orientación del gradiente, sería necesario la aplicación de una segunda derivada que presentará dobles bordes, ver figura 4.4.

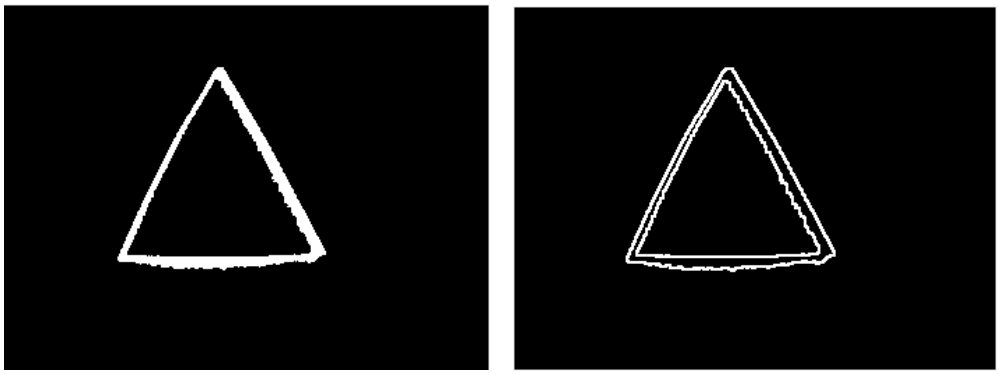


Figura 4.4 Diferencias entre primera derivada y segunda derivada en imágenes de eventos

Por definición, la segunda derivada, ecuación 4.9, no posee carácter vectorial ya que es un escalar, por lo que el uso del descriptor en dichas condiciones no concuerda con su propósito inicial. Además, a esto se le añade la perdida de texturas y de variedad de gradientes al ser una imagen binaria.

$$\Delta f \doteq \vec{\nabla} \cdot \overrightarrow{\nabla f} = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \qquad (4.9)$$

El hecho de que el sensor sea invariante a cambios de iluminación, permite que existen ciertas consideraciones en el método que se pueden omitir ahorrando carga computacional, como por ejemplo, la normalización de los vectores y por tanto la necesidad de utilizar bloques de agrupación de celdas.

Aún así, la aplicación directa del método nos da información de la figura y escena en cuestión, caracterizándola de manera parecida al caso tradicional, pero con pérdida de información como se puede apreciar en la figura 4.5. En dicha figura, se ha realizado la imagen de contornos de una imagen tradicional con el fin de poseer una equivalencia con los resultados obtenidos de agrupación de eventos.

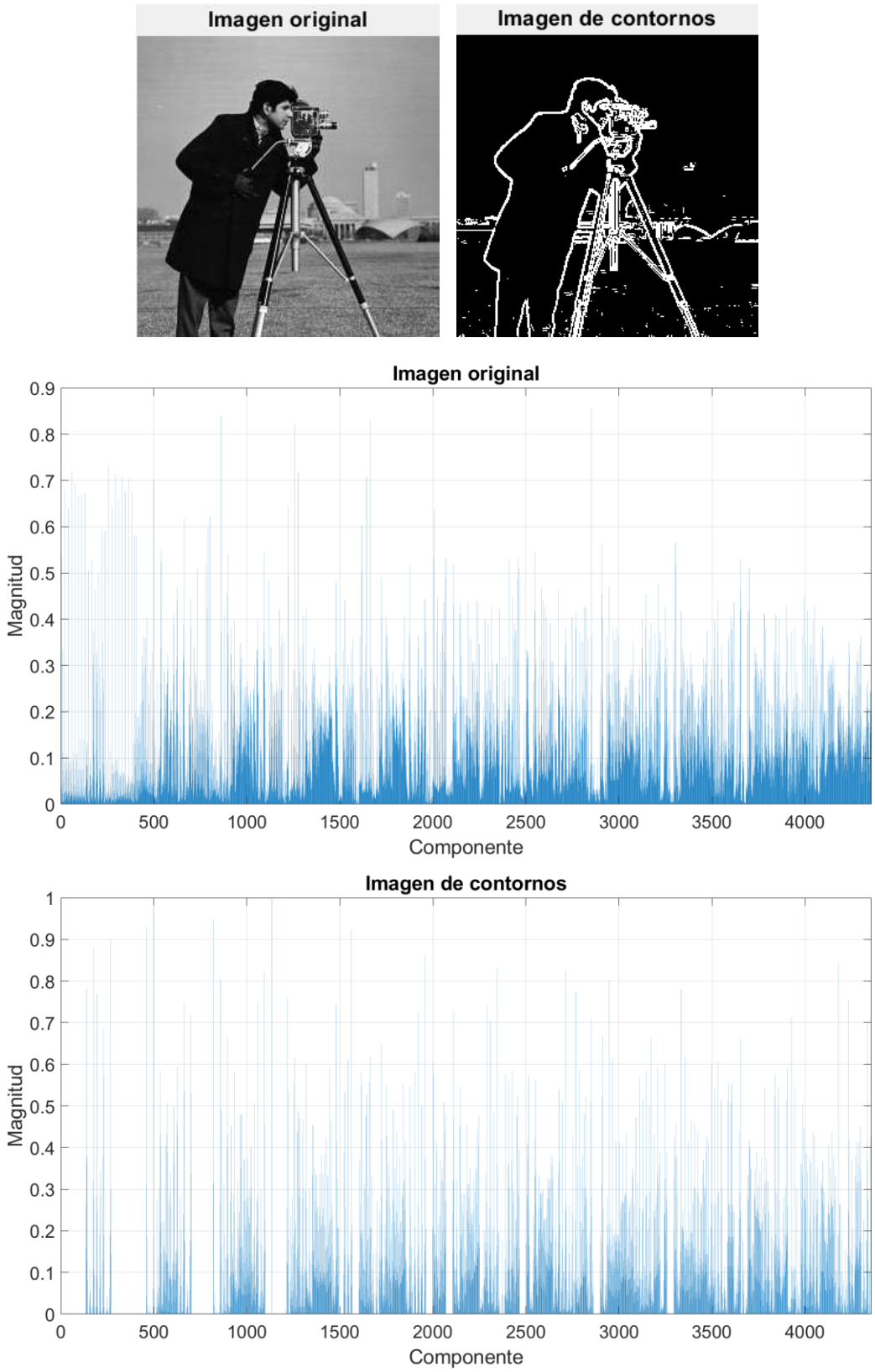


Figura 4.5 Resultado de aplicar HOG a una imagen tradicional y una imagen de contornos

Por todo ello, una posible aplicación del método es su uso sobre el SAE, ya sea el SAE global sobre el intervalo total de uso del sensor o un SAE parcial utilizando ventanas temporales. Para ello, se realizará un escalado de los valores temporales a valores de intensidad de la imagen.

La idea consiste en ser capaces de detectar el espacio donde se producen los eventos, mediante el contorno temporal que se almacena al desplazarse los objetos en el sensor. A su vez, es posible almacenar la dirección de gradiente del movimiento producido, como el equivalente a las texturas de las imágenes tradicionales, que permitirá en una etapa posterior de procesamiento, conocer la dirección y velocidad de movimiento.

Esta adaptación otorga información, no tan solo espacial como se ha realizado hasta la fecha, si no también temporal, ver figura 4.6.

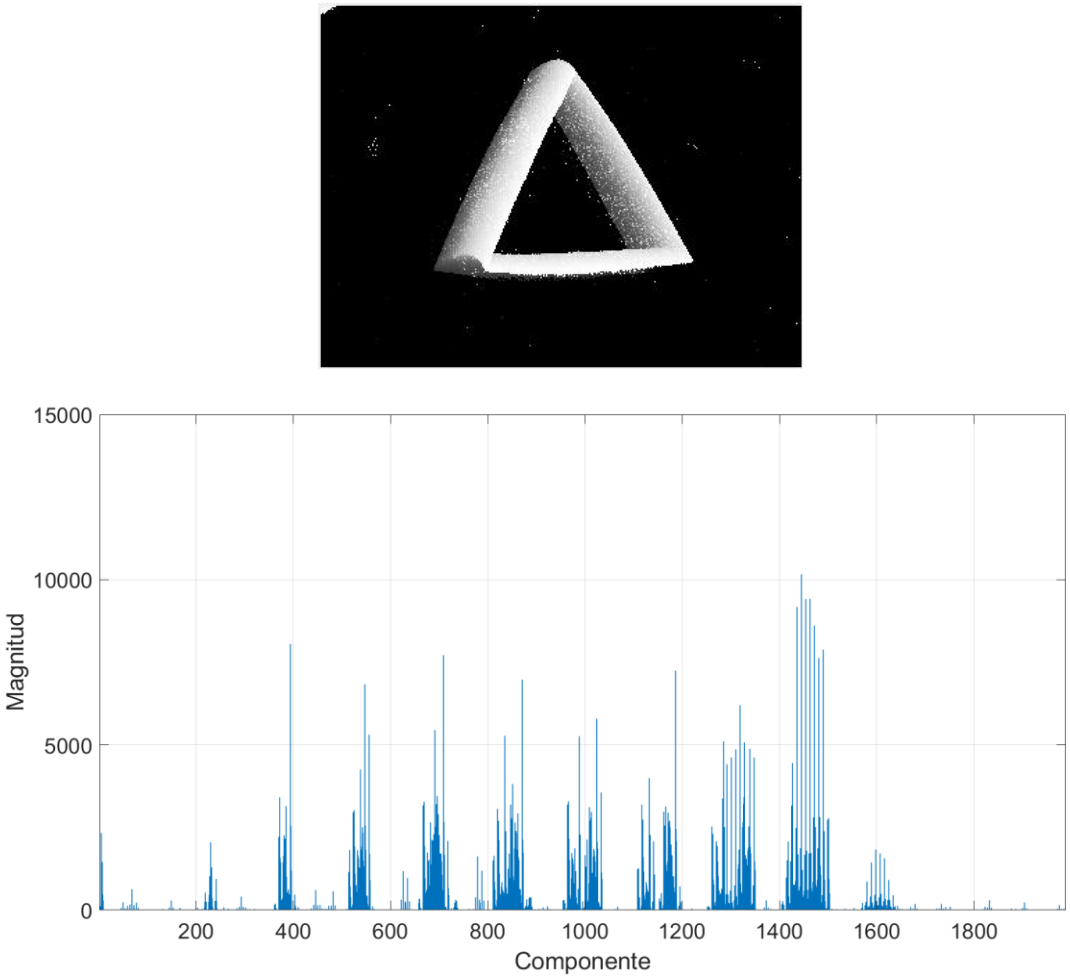


Figura 4.6 Resultado de aplicar HOG al SAE

Algoritmo

Código 4.1 HOG modificado

```
PARA cada píxel
   PARA cada celda
      PARA cada valor discreto angular
         Calculamos pesos y añadimos valor a histograma
      FIN PARA
   FIN PARA
FIN PARA
```

4.3 GIST

GIST, descrito por primera vez en [20] se basa en la habilidad del ser humano en la extracción de ciertas regiones caracterizadas por su color o textura, permitiendo obtener información de ciertos parámetros que definen cada imagen (naturalidad, franqueza, aspereza, expansión, robustez…). Su propósito esencial es la clasificación de escenarios a partir de dicha información.

Para ello existen diferentes versiones del descriptor modificadas para problemas particulares. Por tanto, tomaremos la versión desarrollada en [9] realizando una explicación detalla del proceso de obtención de características, y posteriormente, se mostrará las modificaciones para la adaptaciones en imágenes de eventos.

4.3.1 Pirámide de imágenes

La primera etapa para la creación del descriptor es obtener diferentes niveles de la imagen construyendo una pirámide. De esta forma, aplicaremos el método sobre diferentes escalas del escenario, otorgando más información al conjunto global.

Dicha pirámide está compuesta por un primer nivel, donde se encuentra la imagen original de entrada, y número **x** de nuevos niveles, formados por un suavizada mediante un filtro LP más un escalado que reduce a la mitad de las dimensiones de la imagen del nivel superior, 0.5Mx0.5N. En la figura 4.7 se muestra una pirámide de dos niveles.

Figura 4.7 Diferentes niveles de la pirámide GIST

4.3.2 Filtrado de Gabor

Para poder añadir información de la orientación de los bordes en la escena, se utiliza **m** filtrados de Gabor en el rango [0º,180º], permitiendo resaltar en cada uno de estos filtrados el conjunto patrones visuales que caracterizan a cada escenario.

Una función de Gabor [21] es la unión de una función sinusoidal y una función gaussiana en un espacio bidimensional, siguiendo la distribución de la ecuación 4.10.

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} e^{i2\pi u_0 x} \quad (4.10)$$

Para poseer un conjunto de funciones de Gabor que posean diferentes orientaciones y longitudes de ondas se realiza el cambio a $\boldsymbol{f_{pq}(x,y)}$:

$$f_{pq}(x,y) = \alpha^{-p} f(x',y') \quad (4.11)$$

Donde sigue el cambio de variables:

$$x' = \alpha^{-p}(x\cos\theta_q + \mathrm{y}\sin\theta_q) \quad (4.12)$$

$$y' = \alpha^{-p}(y\cos\theta_q - \mathrm{x}\sin\theta_q) \quad (4.13)$$

$$\text{con } \theta_q = \frac{n\pi}{k}$$

Siendo **p** el parámetro que permite dilatar o contraer la máscara y **q** el parámetro que permite rotarla, consiguiendo una familia de máscaras diferentes, ver figura 4.8.

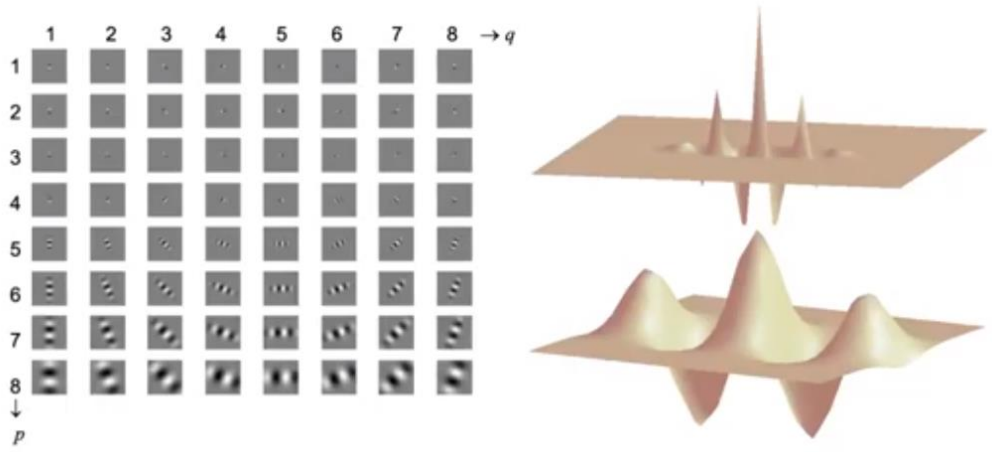


Figura 4.8 Familia de funciones Gabor

La aplicación de este filtro dará como resultado una matriz de números complejos que se podrán dividir por magnitud y por fase, de la que se toma únicamente la magnitud.

4.3.3 Reducción por celdas

Al igual que en HOG, una forma de reducir la información obtenida es la agrupación de la información en celdas. Estas celdas, se situarán de manera equiespaciada en toda la imagen filtrada y aportarán como información, la media de todos los valores incluidos en ellas, añadiendose al vector de características.

Dicha media se realiza para agrupar los valores de coincidencia con la máscara de Gabor tras aplicar la convolución a la imagen inicial, de forma que los valores más altos representarán la similitud de dicha zona con la máscara utilizada. La coincidencia producida dependerá de la forma de las texturas y los contornos de la imagen, pudiéndose extraer dicha información.

4.3.4 Adaptación a imágenes de eventos

El uso de GIST en imágenes de eventos nos permitirá definir la imagen según la orientación de sus bordes, pero se perderá la virtud de poder interpretar características como la textura de la imagen, que está ausente en imágenes de eventos.

Realizando el método GIST sobre imágenes tradicionales y sobre imágenes de contorno es posible observar lo comentado.

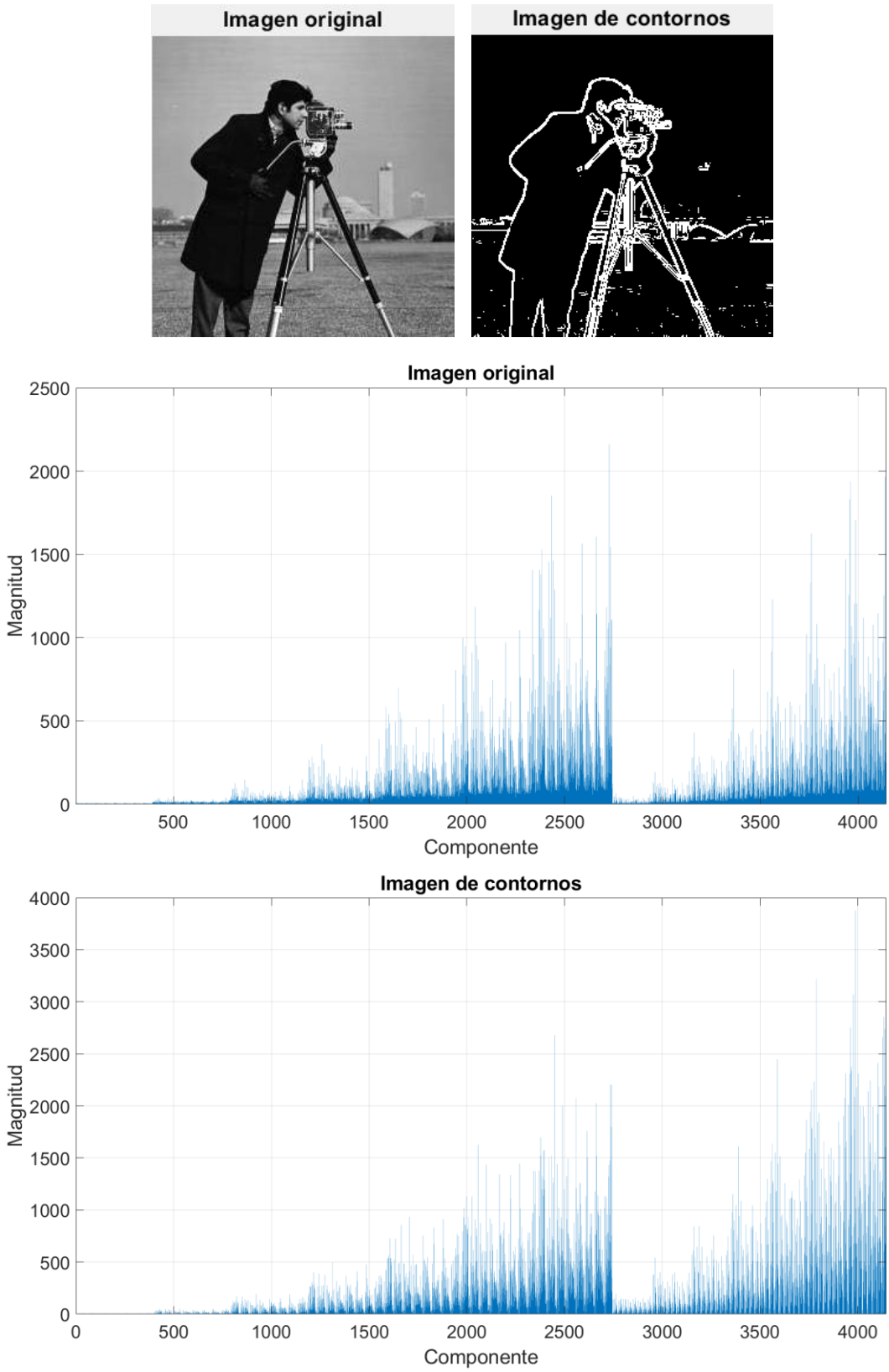


Figura 4.9 Resultados de aplicar GIST en imágenes tradicionales y de contorno

En la figura 4.9 se pueden apreciar las diferencias principales al aplicar el método sobre ambas imágenes a simple vista, teniendo la imagen de contorno menos información que las imágenes tradicionales como era de esperar. Los vectores se han obtenido para celdas de tamaño 50x50 píxeles, 8 ángulos diferentes para Gabor, 7 longitudes de onda diferente y 2 niveles en la pirámide.

Es posible realizar una normalización del vector basada en HOG, normalizando parcialmente el vector resultante de cada subimagen de Gabor. Este cálculo permite que se homogenicen los valores del vector, además de la posibilidad de formar información redundante aumentando robustez la descripción final.

Invarianzas del descriptor GIST

Existe una posible modificación del vector de características para otorgarle invarianza a traslación y hacerlo más robusto en la detección.

Dicha modificación es la eliminación de la separación por celdas, aplicando directamente la media sobre la imagen filtrada. Esto permite conseguir un único valor por cada una de las imágenes filtradas, otorgando al descriptor invarianza a traslación, ver figura 4.10. Para conseguir invarianza a rotación, se puede aplicar el cambio a polares como se explicará en la sección 4.4.

Para añadir más información al conjunto final, se amplia el rango de ángulos y el número de longitudes de ondas para el filtro de Gabor.

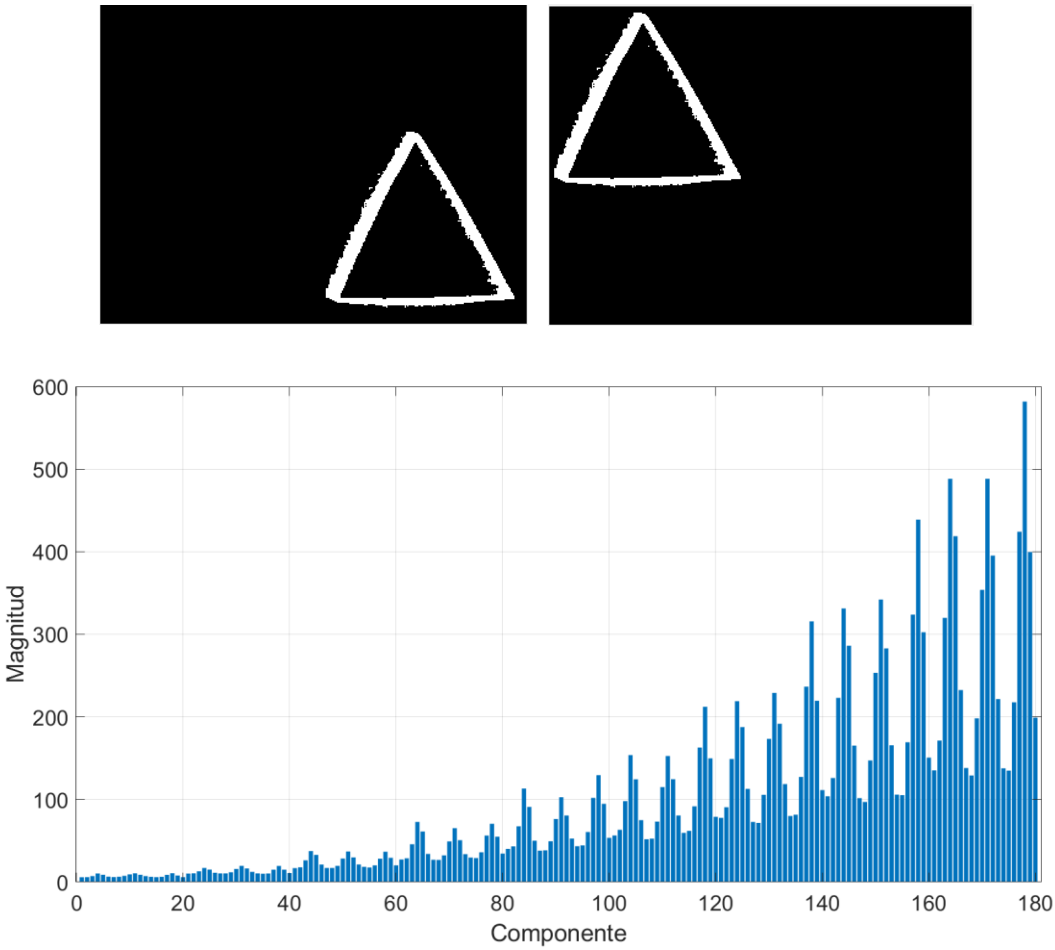


Figura 4.10 Resultado de invarianza en GIST

La diferencia fundamental entre usar ambos métodos es el nivel de detalle detectado en la escena. Si solo se usa una celda, la información captada es la general de toda la escena, mientras que si se utilizan celdas más pequeñas, es posible reconocer detalles más específicos dentro de ella.

Reducción de componentes y computo

La propuesta realizada para reducir la longitud del vector es utilizar un único nivel de pirámide, ya que el cambio de escala de la figura y su filtrado tiene como propósito utilizar la información de las texturas de la imagen, que como ya se ha comentado, está ausente en imágenes de contorno. Esta simplificación reduce notablemente el cómputo y el número de componentes a aproximadamente la mitad. Aún con la reducción de información planteada, es posible reconocer la orientación de los bordes de figura gracias a Gabor, que aporta bastante información de la escena.

Además, aplicando una única celda como modificación para obtener invarianza se consigue reducir también la longitud del vector notablemente.

A estas reducciones se le añade variar los parámetros de celda, ángulos de filtrado y longitudes de onda para el filtrado de Gabor, a costa de reducir la información del conjunto final.

Algoritmo

Código 4.2 GIST modificado

```
PARA cada longitud de onda
  PARA cada ángulo
    Calcular correlación entre filtro e imagen
    Realizar media de la matriz resultante
  FIN PARA
FIN PARA
```

4.4 DFT

La DFT o *Discrete Fourier Transform* es un tipo de transformación que nos permite cambiar del dominio temporal al dominio frecuencial, descomponiendo una señal en una suma infinita de senos y cosenos a diferentes frecuencias, amplitudes y fases. Dicha transformación es una modificación de la FT o *Fourier Transform* (ecuación 4.14) realizada para la operación de señales discretas en el tiempo, ecuación 4.15.

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi wt} dt \qquad (4.14)$$

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-i\frac{2\pi}{N}kn} \; para\; k = 0, 1, \dots, N-1 \qquad (4.15)$$

La diferencia principal es el cambio de una integral infinita a un sumatorio finito, donde la suma se realiza sobre todos los componentes de la señal, y por tanto se obtiene un número finito de frecuencias, siendo **N** el número de componentes de la señal discreta y **1/N** el valor de la primera componente frecuencial.

La DFT tiene un amplio uso en tratamiento de señales debido a la gran cantidad de propiedades que posee y la información que nos otorga de la señal a tratar, pudiendo realizar operaciones con la información espectro como, por ejemplo, el filtrado de frecuencias no deseadas.

Estas propiedades se extrapolan al dominio bidimensional, pudiendo aprovecharlas para la aplicación en imágenes digitales entendiéndolas como funciones discretas bidimensionales:

$$F(u,v) = \frac{1}{M \cdot N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} f(x,y) e^{-i2\pi\left(\frac{ux}{M} + \frac{uy}{n}\right)} = A(u,v) e^{i\theta(u,v)} \qquad (4.16)$$

Siendo **M** el número de columnas de la imagen y **N** el número de filas. El resultado obtenido de la transformación es una matriz compleja que se puede descomponer en magnitud y en fase, que se aprovecharán para descripción de la imagen.

Existe una versión que permite obtener la DFT de una manera más eficiente computacionalmente conocida como FFT o *Fast Fourier Transform*, que será el implementada el trabajo.

4.4.1 Propiedades

Algunas de las propiedades más importantes de la DFT son:

- **Separabilidad.** Para la obtención de la DFT bidimensional es posible separar la operación realizando primero la transformada de Fourier unidimensional para las filas y posteriormente para las columnas. Suponiendo una imagen NxN:

$$F(u,v) = \frac{1}{N}\sum_{u=0}^{M-1} e^{-i2\pi\frac{ux}{N}} \sum_{v=0}^{N-1} f(x,y)e^{-i2\pi\frac{uy}{n}} \quad (4.17)$$

- **Traslación.** El desplazamiento de los objetos dentro de la imagen no provocarán cambios en la magnitud, únicamente en la fase:

$$\mathcal{F}\{f(t-t_0)\} = e^{-iwt_0}F(w) \quad (4.18)$$

- **Rotación.** La rotación de la imagen un ángulo cualquiera, provocará la misma rotación en la magnitud de la transformada.
- **Periodicidad.** La transformada de Fourier es simétrica con respecto a su centro (M/2,N/2), por lo que es posible trasladar las magnitudes al centro y situar en él la frecuencia fundamental.
- **Teorema de convolución.** Dicho teorema explica que realizar una convolución en el dominio temporal es equivalente a un producto en el dominio de la frecuencia:

$$\mathcal{F}\{f_1(t) * f_2(t)\} = F_1(w) \cdot F_2(w) \quad (4.19)$$

  Por lo que, la aplicación de máscaras en imágenes tiene su equivalencia con filtros frecuenciales.

4.4.2 Adaptación a imágenes de eventos

Debido a su comportamiento, la DFT se aplica de manera idéntica a imágenes de eventos, obteniendo en el dominio de la frecuencia la información que ha producido la cámara.

El resultado obtenido es el equivalente de aplicar la transformada de Fourier a una imagen de contornos, provocando que el espectro en frecuencia reduzca sus valores de magnitud en frecuencias bajas, dispersando la potencia total en un rango más amplio de frecuencias. Este efecto no provoca que se pierda la información esencial de la imagen, ya que se puede observar en la figura 4.3 que la distribución de los valores más altos de magnitud conservan el mismo patrón.

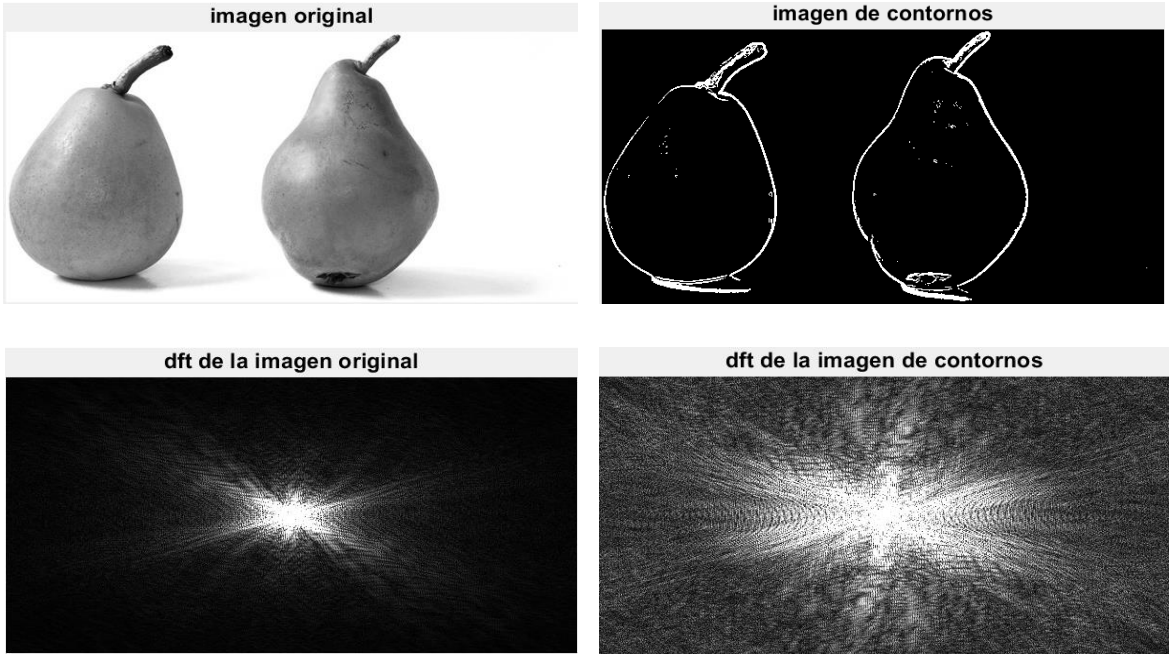


Figura 4.11 Diferencias en espectro de frecuencias entre imagen monocromática y de contornos

La propuesta realizada para la descripción es, por tanto, la concatenación de la matriz de magnitud en un único vector, que poseerá como longitud MxN siendo estas, las dimensiones de cada imagen de entrada. Aplicando dicho método sobre imágenes de eventos se puede observar lo comentado:

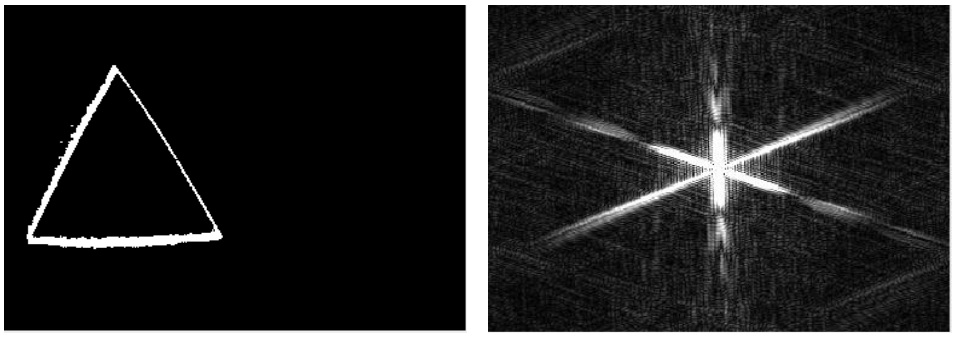


Figura 4.12 Espectro de frecuencias para una imagen de eventos con *"triangle"*

Invarianzas del descriptor DFT

Gracias a las propiedades comentadas en la sección 4.4.1 que posee la transformación al dominio de la frecuencia, se consiguen una serie de invarianzas que permiten que el descriptor sea más robusto ante ciertos cambios en la imagen original.

La más notoria es la invarianza a traslación, que permite que un objeto que se sitúe en cualquier parte de la imagen produzca el mismo vector de características.

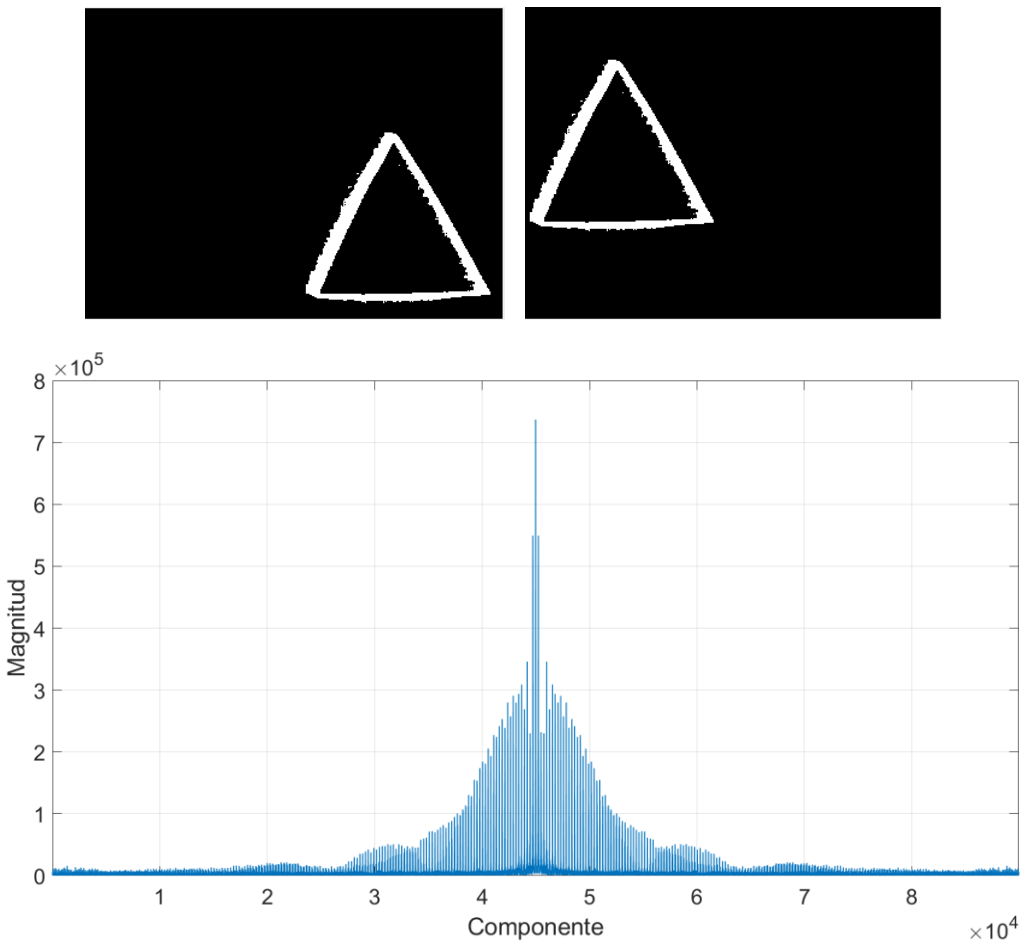


Figura 4.13 Invarianza de descriptor DFT

Dicha invarianza a traslación es posible transformarla en invarianza a rotación y escala, realizando un cambio a coordenadas polares que provocará que un giro de un objeto en la imagen cartesiana, se convierta en una traslación en coordenadas polares.

Las coordenadas polares es un cambio de variables que permite describe una imagen por el ángulo y la distancia de cada píxel con respecto al origen de coordenadas siguiendo las ecuación 4.16 y 4.17.

$$\rho(x,y) = \sqrt{(x-x_c)^2 + (y-y_c)^2} \quad (4.20)$$

$$\theta(x,y) = \tan^{-1}\frac{y-y_c}{x-x_c} \quad (4.21)$$

Siendo $\boldsymbol{x_c}$ y $\boldsymbol{y_c}$ el punto de referencia para la obtención del radio, siendo en este caso particular el centro de la imagen (M/2, N/2).

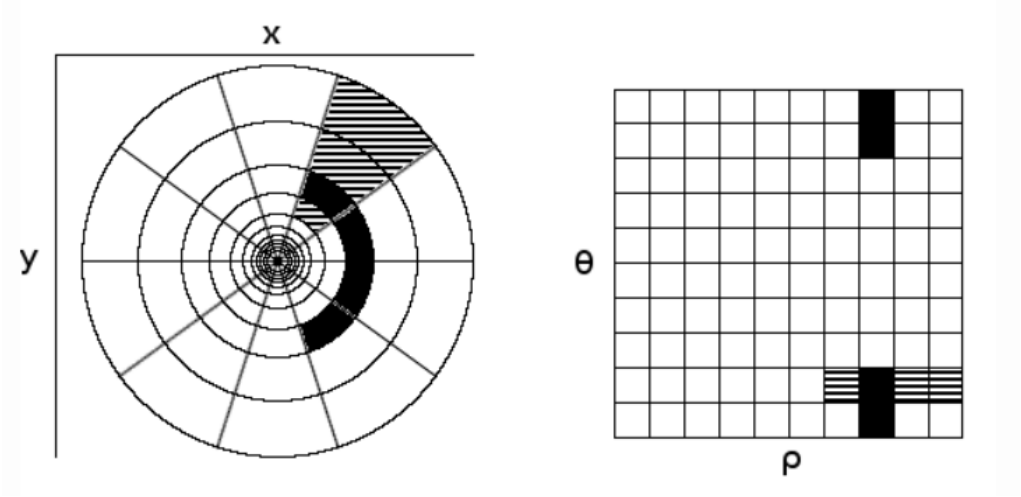


Figura 4.14 Descripción gráfica de cambio a coordenadas polares

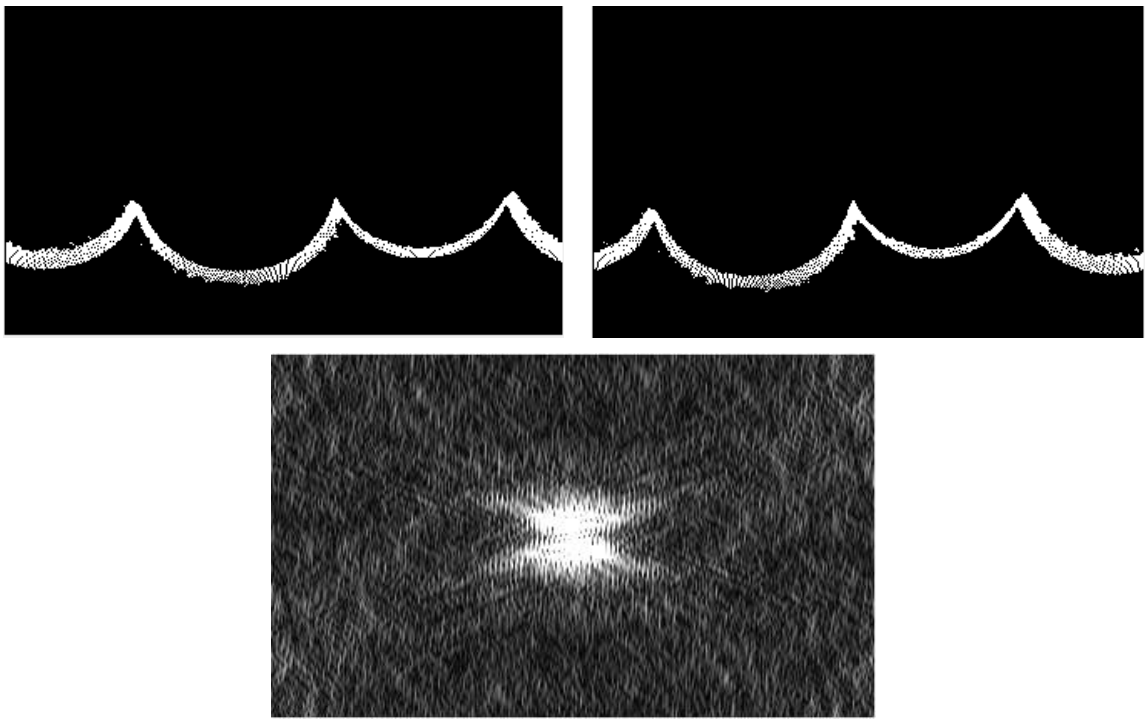


Figura 4.15 DFT aplicada a imágenes de un triángulo centrado en el origen rotado un ángulo θ

En el caso en el que se quisiera conocer la posición de los objetos en la imagen porque esta fuera importante, sería necesario añadir la información angular tal y como se ha realizado con la magnitud, ya que en ella se almacena la información espacial.

Evaluando el vector obtenido, se puede apreciar que la componente de continua del espectro (componente 0) es el número de eventos que posee la imagen, por lo que particularizará el vector en función de la cantidad de eventos utilizados para la agrupación.

Para reducir este efecto se divide la matriz de magnitud por dicha componente, "normalizando" el vector de características, y haciéndolo invariante al método de agrupación utilizado.

$$A_f = \frac{A}{A_{DC}} \qquad (4.22)$$

Reducción de componentes y computo

Como hemos mencionado, el vector obtenido posee MxN componentes, lo que puede llegar a ser una cantidad excesiva para algunas aplicaciones en tiempo real. Nuestro caso particular de procesamiento a bordo del ornitóptero, debido a su tamaño, no permite soportar elementos demasiado pesados, por lo que es indispensable reducir el cómputo y poder incorporar procesadores simples.

Analizando el vector obtenido, es posible realizar una serie de modificaciones para reducir su longitud sin influir en exceso en la información implícita. En primer lugar, evaluando el entorno en frecuencias, es posible realizar una reducción de aquellas componentes situadas en frecuencias altas, que por lo general, no poseen valores de magnitud elevados y no aportan información la imagen. En segundo lugar, el vector resultante posee simetría par con respecto a la frecuencia de continua, por lo que es posible eliminar la información redundante.

Uniendo ambas observaciones es posible reducir el vector final en torno a un 35% del conjunto de datos inicial.

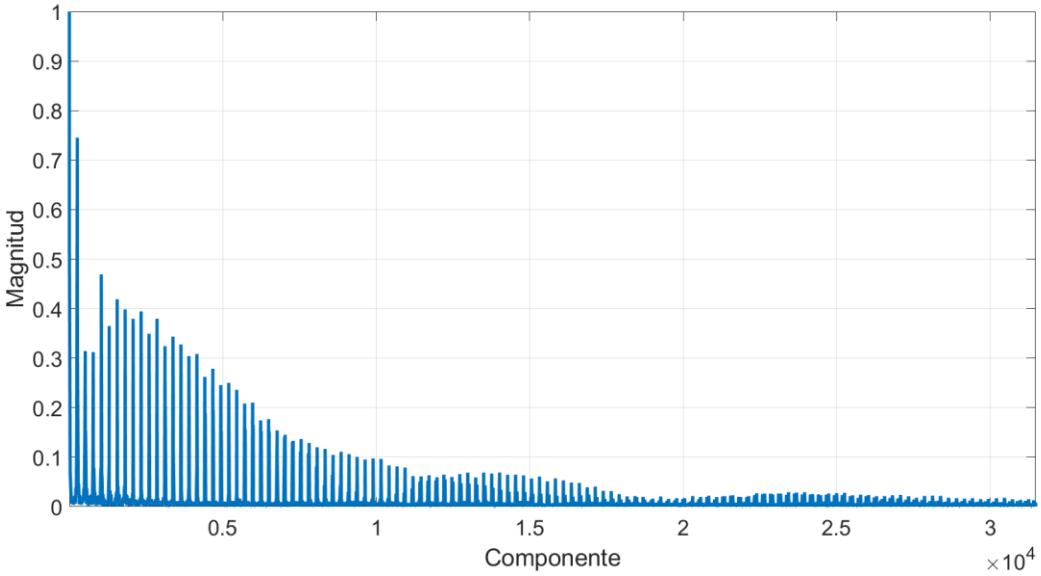


Figura 4.16 Vector reducido de características usando DFT para *"triangle"*

Otra posible reducción del vector y del coste computacional del sistema es aplicar la transformada unidimensional a filas o columnas como se aplica en la literatura [22] [9], consiguiendo dos espectros de frecuencias diferentes denominados FS o *Fourier Signature*.

Dichos espectros se puede combinar eliminando las componentes de alta frecuencia para obtener un nuevo vector de características diferente, pero con la misma información frecuencial. En esta nueva versión del descriptor se perdería la invarianza a traslación, lo que puede ser útil en algunas aplicaciones

4.4.3 Aplicaciones para su uso

Reconocimiento de escenas

El vector resultante contiene la información esencial de la escena distribuida en diferentes rangos de frecuencias, y con las diferentes invarianzas comentadas, permitiendo que sea posible distinguir entre diferentes escenarios, únicamente utilizando dicho descriptor. La idea es conseguir distinguir los tres escenarios de ensayos *"Testbed"*, *"Soccer"* y *"Hills"* utilizando un clasificador SVM o *Support Vector Machine* que consiga separar espacialmente el conjunto de imágenes obtenidos de cada vuelo. Esto aportará la capacidad al robot de distinguir dónde se sitúa.

Además, es posible conocer no solo la información del escenario, sino también el sentido de movimiento de la cámara, es decir, se puede conocer la dirección y el sentido de movimiento del ornitóptero. Esto es posible sencillamente añadiendo la polaridad de cada evento al conjunto de imágenes, realizando una distinción de color entre cada polaridad, figura 4.17, siendo los píxeles positivos el color más intenso "255" y los píxeles de polaridad negativa igual a "127".

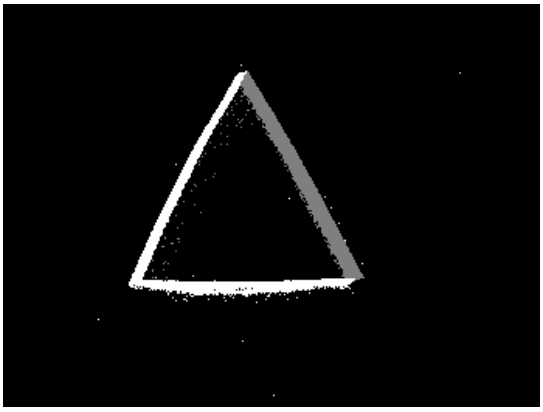


Figura 4.17 Imagen de eventos con polaridad sobre *"triangle"*

Correlación de fase

La correlación, ecuación 4.23, es una operación similar a la convolución que permite obtener el desplazamiento temporal que hay entre dos señales.

$$f(x)\ o\ g(x) = \sum_{m=0}^{M-1} f(m)g(x+m) \quad (4.23)$$

Para ello se realiza una comparación entre ambas, desplazando una de ellas sobre la otra, y consiguiendo un vector que contendrá su máximo en el lugar de mayor coincidencia entre ellas, que puede interpretarse como un desfase o desplazamiento.

Aplicado sobre el dominio de la frecuencia obtenemos que dicha operación se transforma en un producto de conjugados:

$$\mathcal{F}\{f_1(x)\ o\ f_2(x)\} = F_1^*(w) \cdot F_2(w) \quad (4.24)$$

El teorema se traslada al campo bidimensional obteniendo una matriz comparativa entre imágenes. Basándonos en la propiedad de la **traslación**, ecuación 4.25, es posible despejar el término que posee el desplazamiento entre ambas, obteniendo la ecuación 4.26.

$$F_2(w_x, w_y) = e^{-i2\pi(w_x t_x + w_y t_y)} F_1(w_x, w_y) \quad (4.25)$$

$$e^{-i2\pi(w_x t_x + w_y t_y)} = \frac{F_1(w_x,w_y)F_2^*(w_x,w_y)}{|F_1(w_x,w_y)F_2(w_x,w_y)|} \quad (4.26)$$

Este nuevo termino resultante se conoce como CPS o *Cross Power Spectrum* y nos permite calcular la transformada de Fourier de una función delta de Dirac en la posición del desplazamiento producido.

Finalmente, se realiza la transformada de Fourier inversa para conocer la información del desplazamiento, ver figura 4.18.

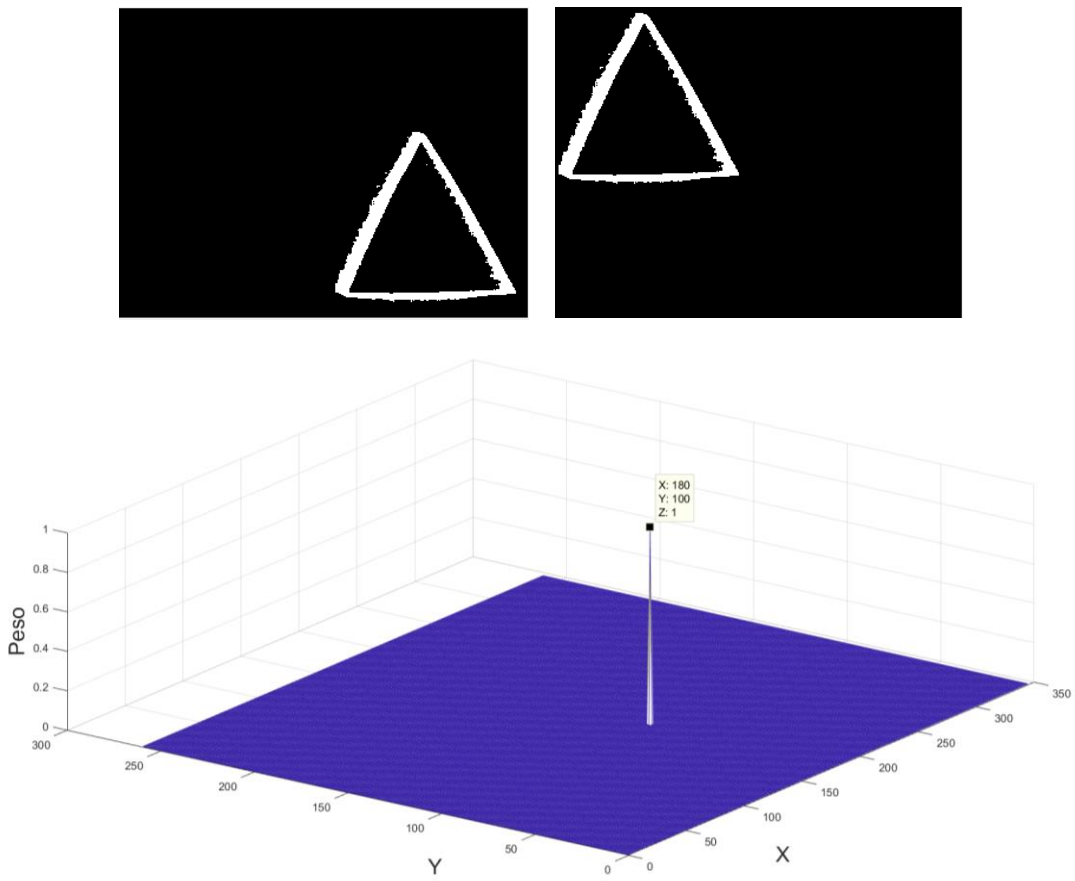


Figura 4.18 Resultado de aplicar correlación de fase en imágenes iguales trasladadas

4.5 Conclusiones

Por lo general, todos los descriptores desarrollados poseen la capacidad de describir las imágenes de eventos, caracterizándola de manera diferente cada uno de ellos. Aún así, y como era de esperar, las nuevas imágenes aportan menos información que las imágenes tradicionales por la falta de intensidades.

Basándonos en el método original, es posible crear variaciones de cada uno de ellos, obteniendo vectores de características completamente diferentes y tiempos de ejecución también variables. Por ello, se han realizado varias versiones de cada descriptor utilizando las adaptaciones mencionadas en cada uno de ellos, para los que se tomará el más interesante para realizar pruebas en el capítulo 5 y comprobar así su funcionamiento.

5 EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

5.1 Introducción

En este capítulo se realizarán pruebas sobre las diferentes modificaciones propuestas en el capítulo anterior para cada descriptor, con el fin de comprobar las robustez que poseen al caracterizar cada imagen de manera global.

Para considerar que un vector de características es bueno, es necesario que su dispersión intraclase sea baja y su dispersión interclase sea alta, facilitando la labor de clasificación. Todo con el menor coste computacional posible. Por ello, se realizarán las siguientes pruebas para comprobar de manera simple lo comentado.

Dichas pruebas se dividirán principalmente en diferenciar los vectores de características para:

1. Imágenes similares.
2. Imágenes similares con ruido, traslación y rotación.
3. Imágenes diferentes.

Se mostrarán un conjunto de gráficas con el vector resultante para cada una de las imágenes elegidas y tablas donde se mostrará el módulo de la diferencia entre vectores y el ángulo que forman, siguiendo las ecuaciones 5.1 y 5.2 respectivamente. Esta última prueba se realizará para conocer como primera aproximación la proximidad entre vectores para cada par de imágenes.

$$error = \left|\vec{A} - \vec{B}\right| \tag{5.1}$$

$$\theta = \arccos\left(\frac{\vec{A}\cdot\vec{B}}{|\vec{A}||\vec{B}|}\right) \tag{5.2}$$

Siendo $\vec{A}$ y $\vec{B}$, los vectores de carácterísitcas para dos imágenes diferentes, figura 5.1.

Consideraremos imágenes similares a aquellas que pertenecen al mismo conjunto de datos e imágenes diferentes a aquellas que pertenecen a conjuntos diferentes. Dichos conjuntos se agruparan en dos sets formados por los de datos *"triangle"* y *"square"*, figura 5.2, y *"Testbed"*, *"Soccer"* y *"Hills"*, figura 5.3. Ambos sets se han generado usando el método de agrupación espacial con 5000 eventos y 10000 eventos respectivamente. Además, el primer conjunto ha sido filtrado para utilizando el filtrado de votos.

Finalmente, se realizará una comparativa de los resultados obtenidos y una conclusión de cuál de ellas posee un mejor comportamiento tras este primer análisis.

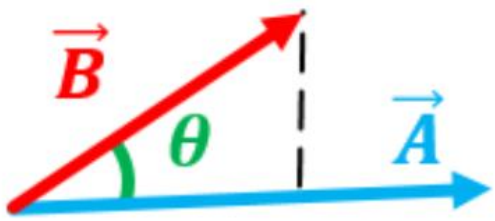
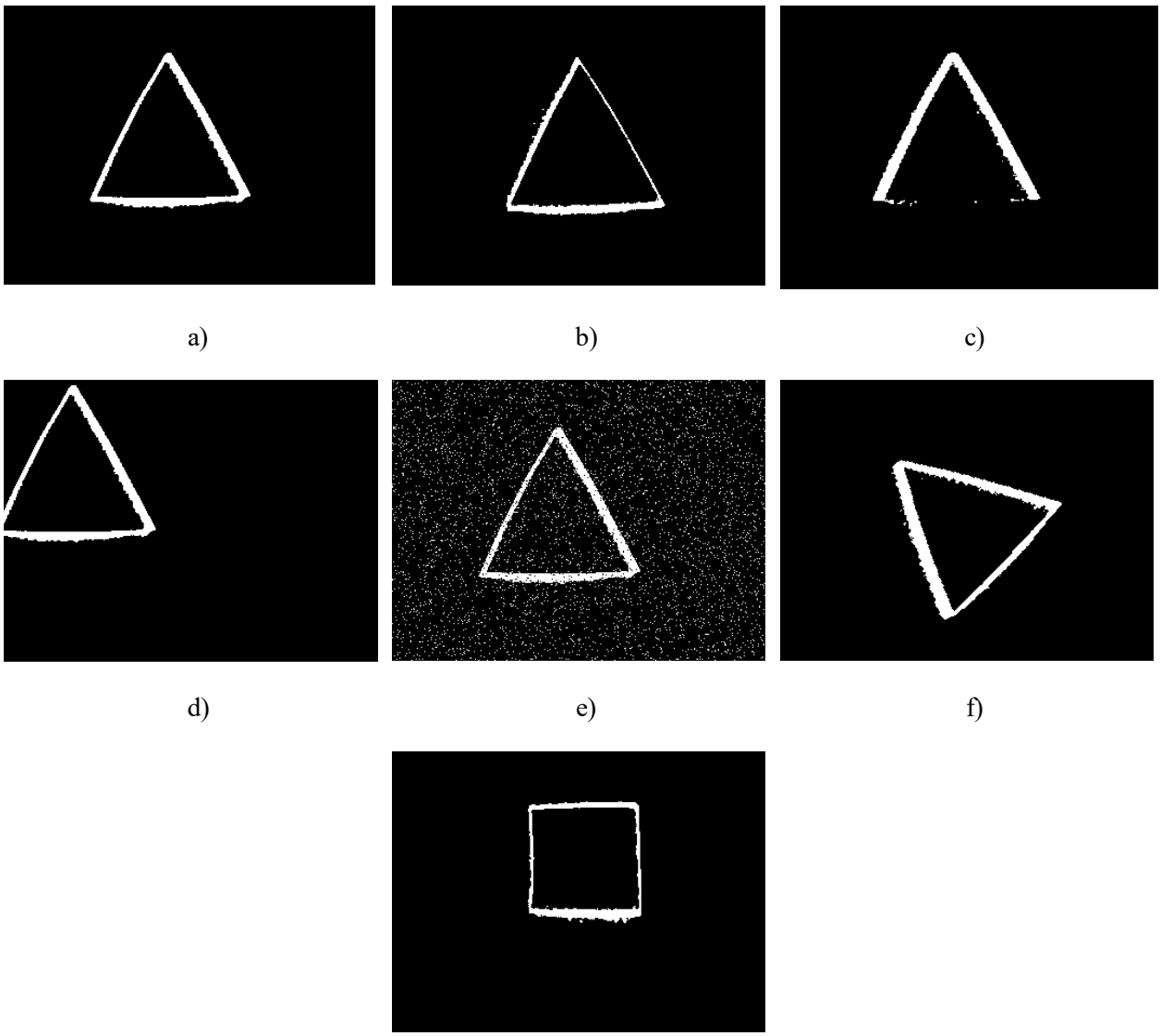


Figura 5.1 Comparación de vectores

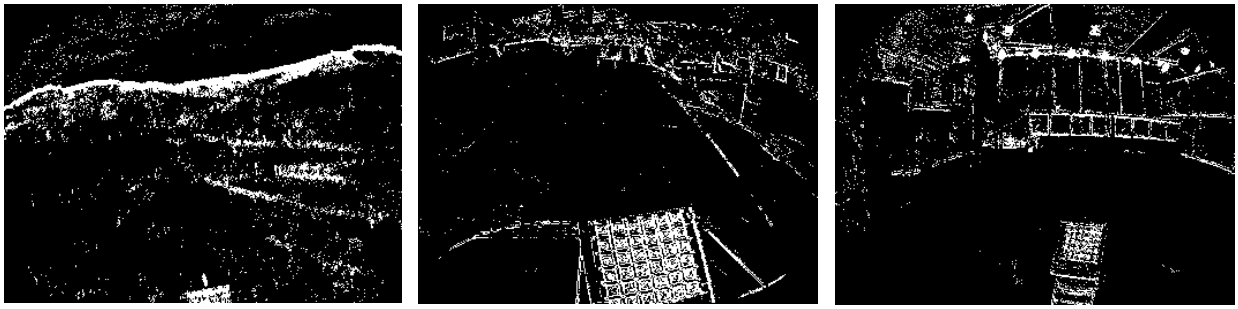


a) b) c)

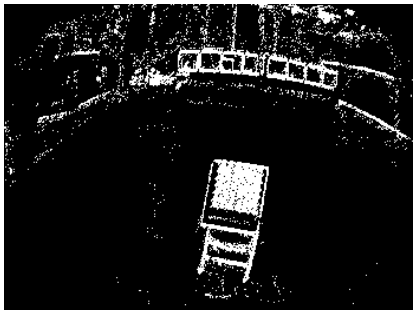
d) e) f)

g)

Figura 5.2 Set de figuras geométricas a) Figura original b) Figura generada en instante diferente c) Figura incompleta d) Imagen trasladada e) Imagen ruidosa f) Figura rotada g) Figura diferente



a) b) c)



d)

Figura 5.3 Set de escenarios reales a) *Hills* b) *Soccer* c) *Testbed 1* d) *Testbed 2*

5.2 Experimentos con HOG

A continuación, se mostrarán los diferentes vectores obtenidos de aplicar el método de HOG. Como se comentó en su desarrollo, existen diferentes modificaciones posibles para realizar, por lo que los resultados obtenidos se han centrado en el uso de HOG sin normalización y sin uso de bloques. Únicamente se ha utilizado la división por celdas para conseguir los resultados.

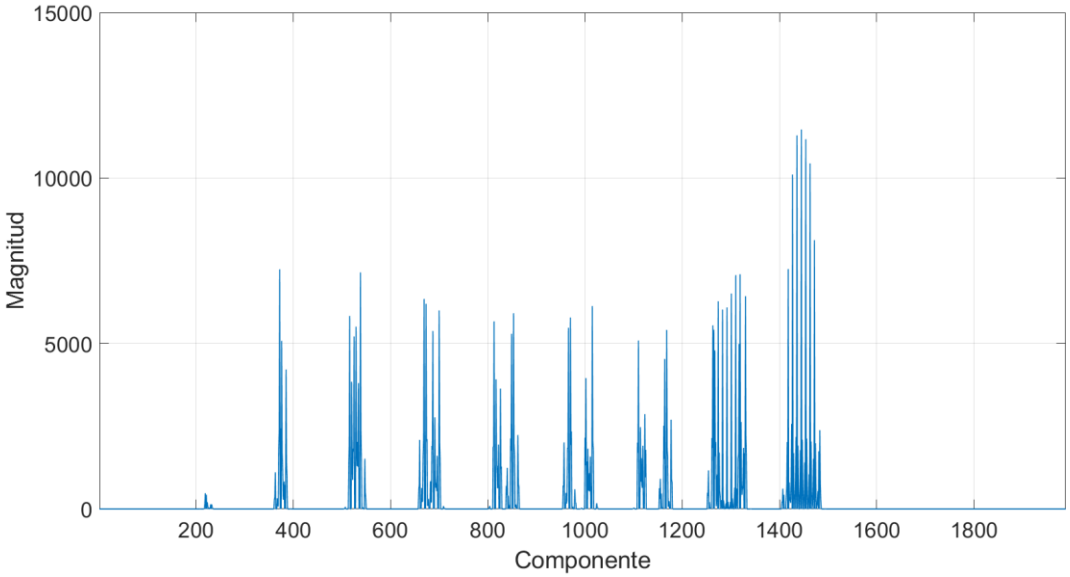


Figura 5.4 Resultado de aplicar HOG a *"triangle"* centrado

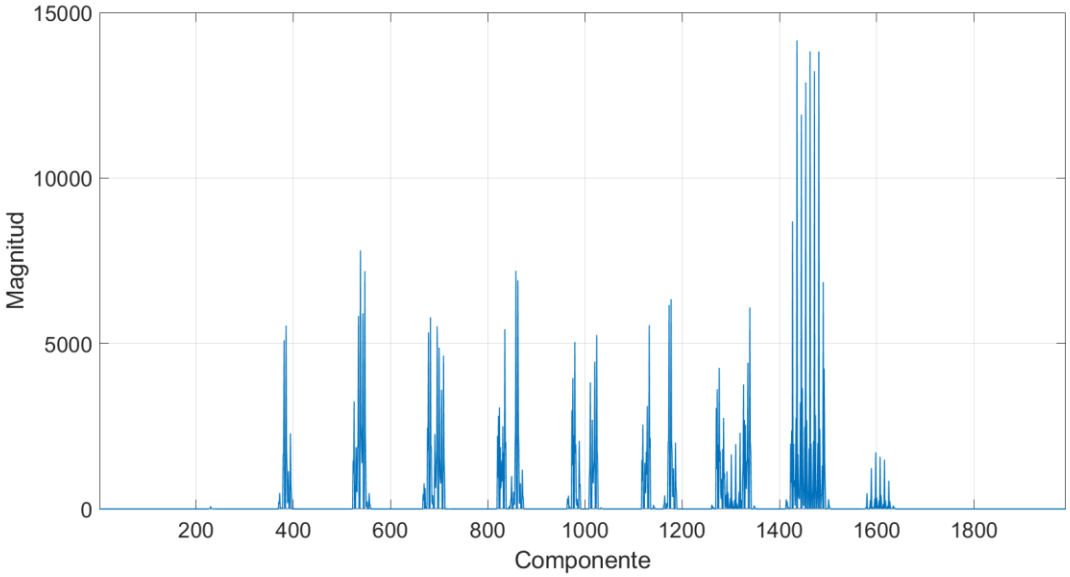


Figura 5.5 Resultado de aplicar HOG a *"triangle"* centrado generado en instante diferente

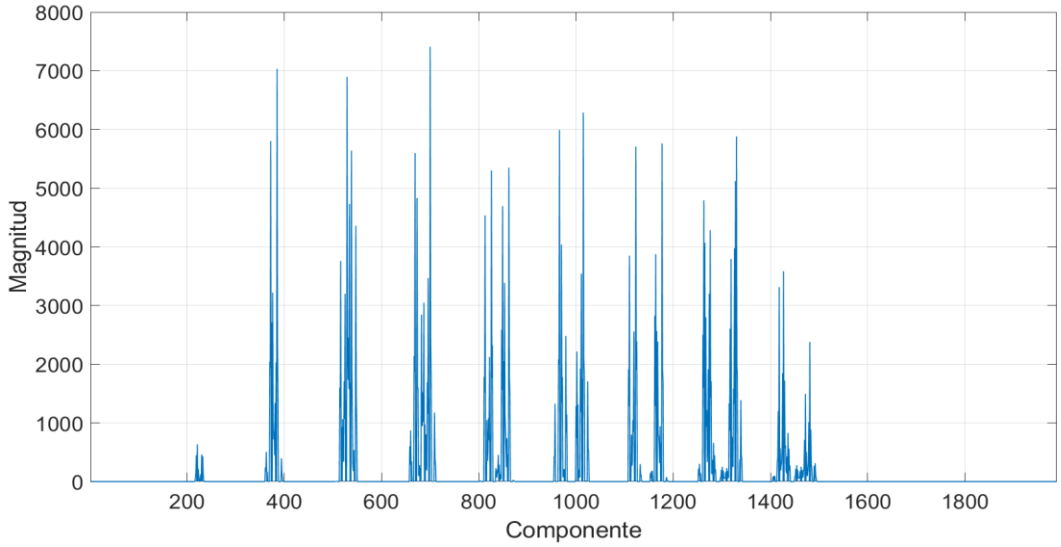


Figura 5.6 Resultado de aplicar HOG a *"triangle"* incompleto

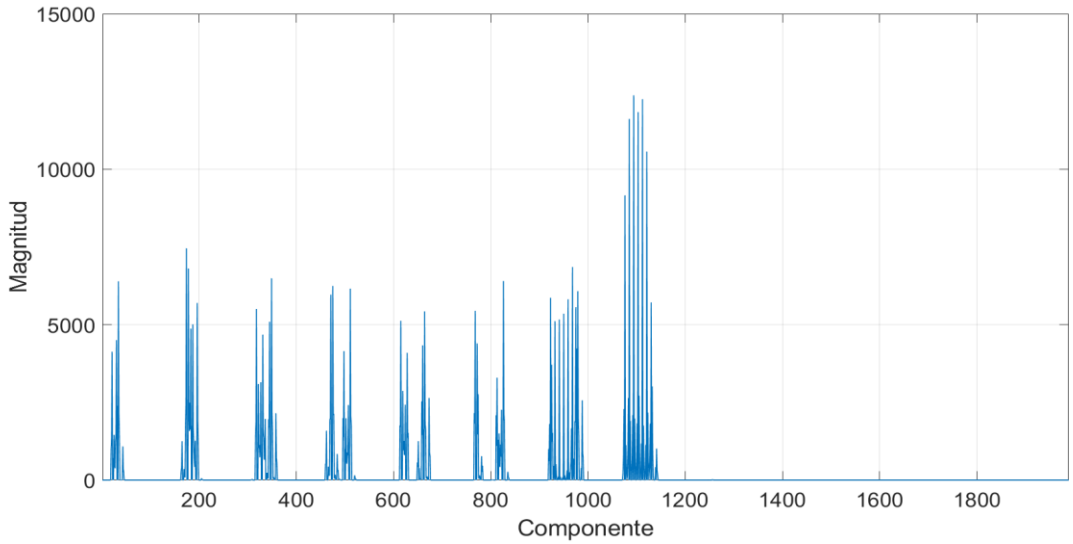


Figura 5.7 Resultado de aplicar HOG a *"triangle"* trasladado

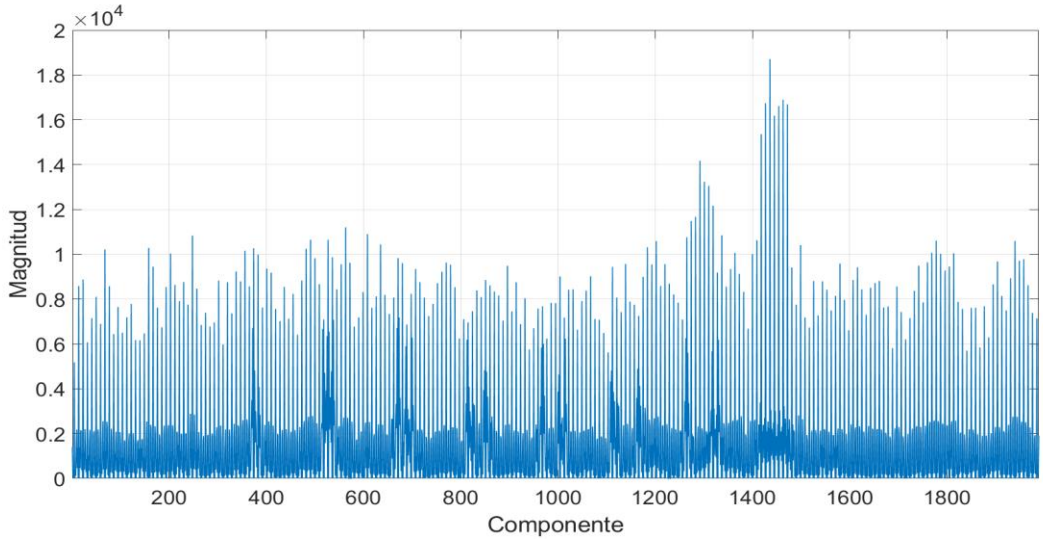


Figura 5.8 Resultado de aplicar HOG a *"triangle"* con ruido

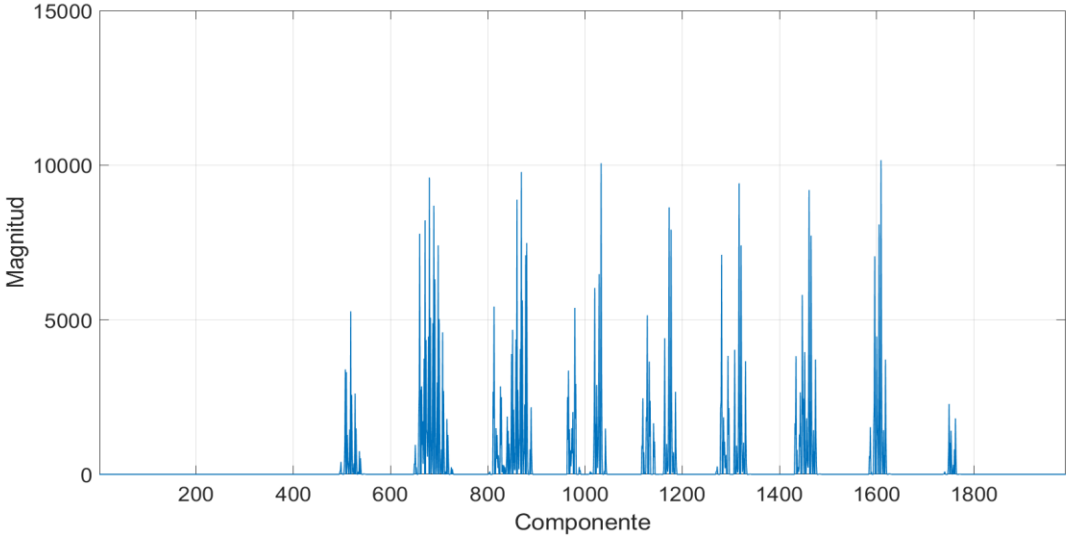


Figura 5.9 Resultado de aplicar HOG a *"triangle"* rotado

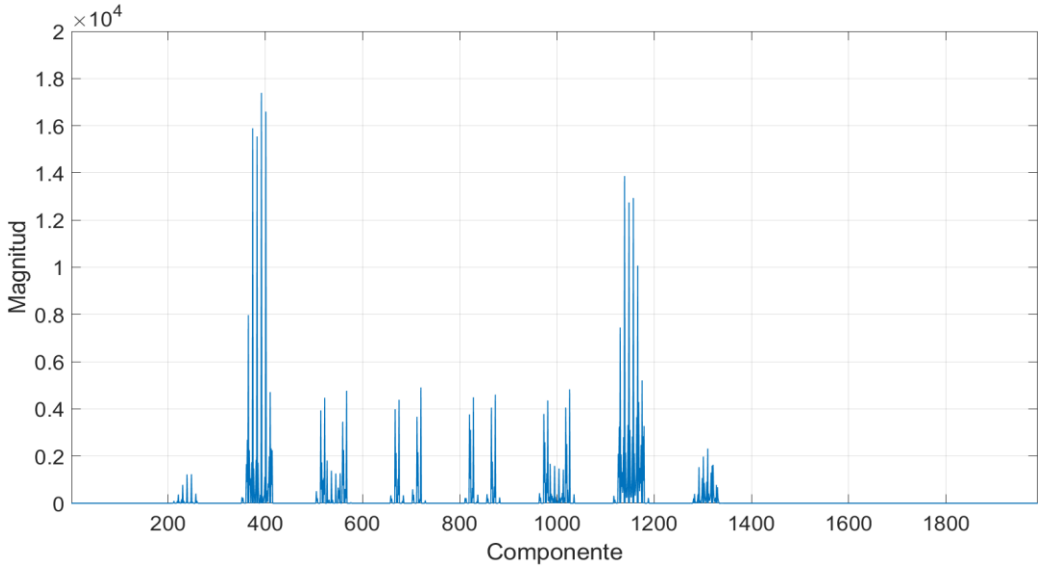
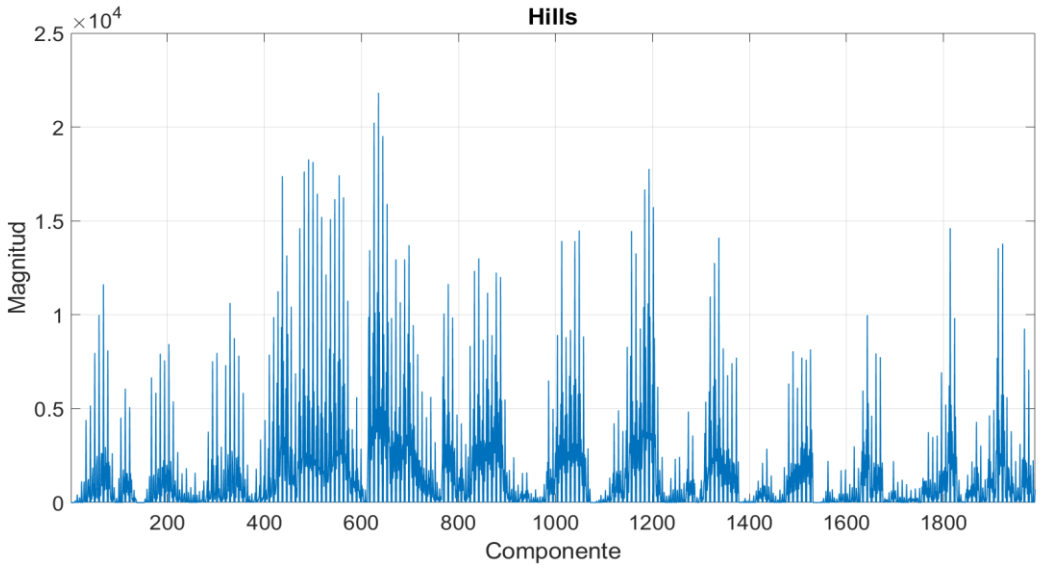


Figura 5.10 Resultado de aplicar HOG a *"square"*



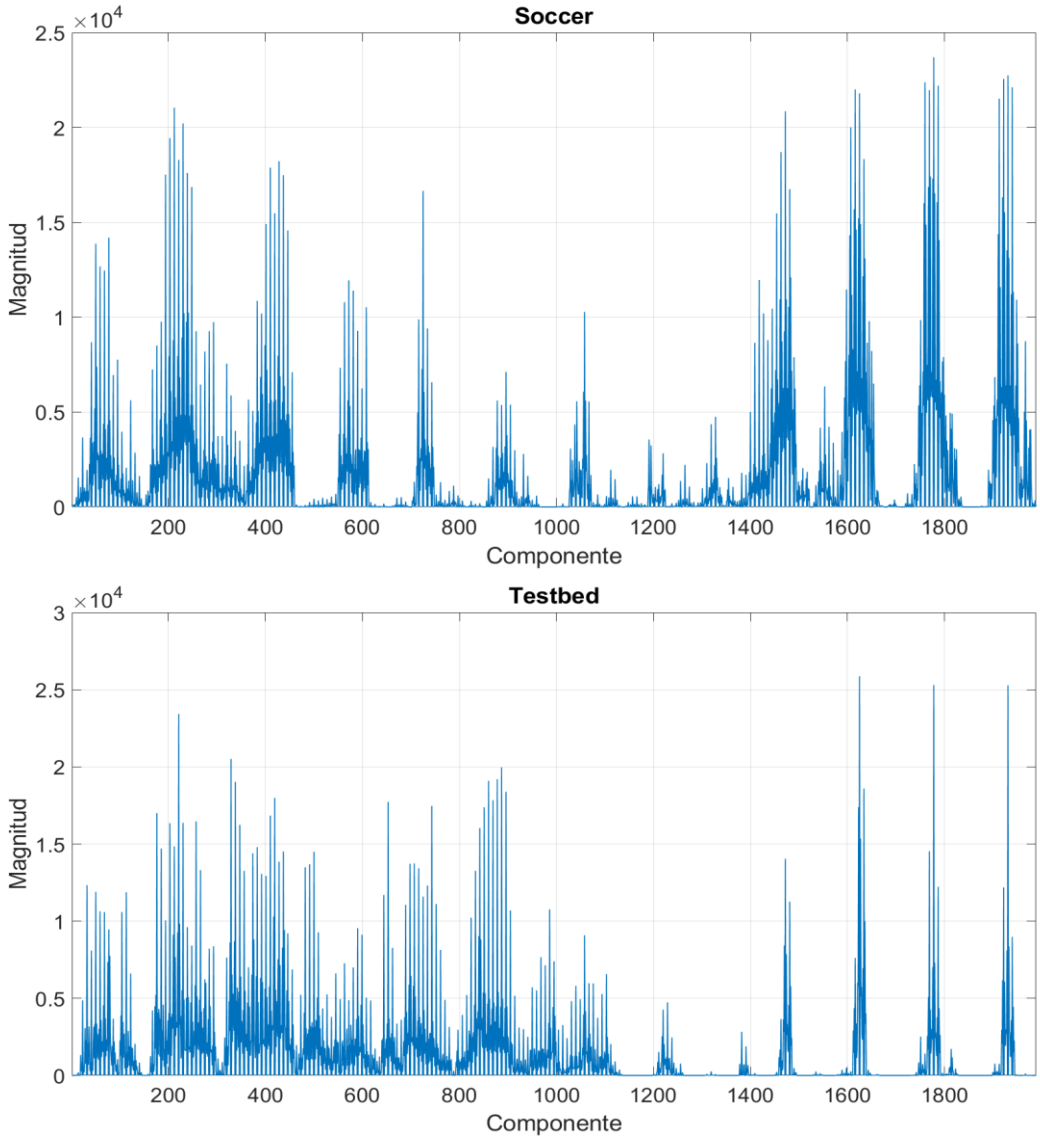


Figura 5.11 Resultado de aplicar HOG a escenarios reales

SAE

Se ha realizado un prueba adicional en este descriptor para comprobar su funcionamiento sobre el SAE de una ventana temporal de 700 ms, donde se puede apreciar un movimiento diferente entre ambos, ver figura 5.12.

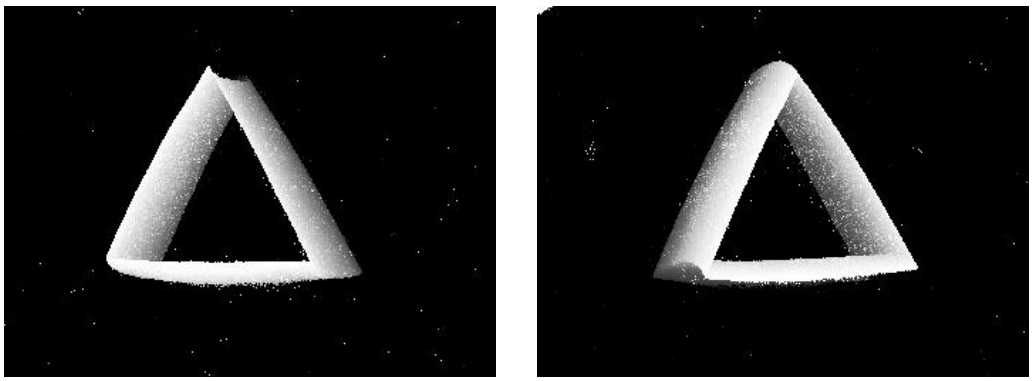


Figura 5.12 SAE de triángulo para instantes diferentes a) SAE 1 b) SAE 2

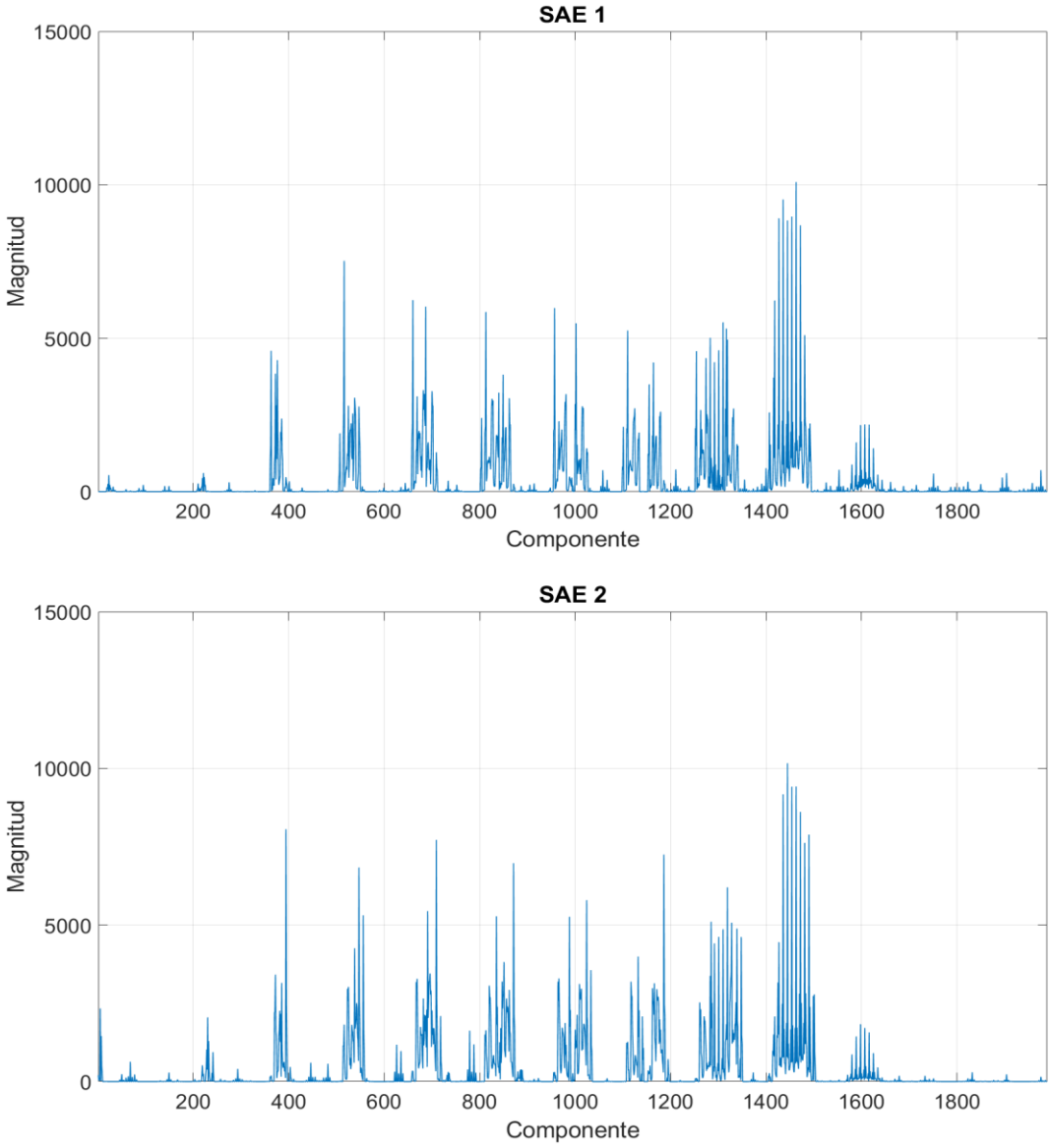


Figura 5.13 Resultado de aplicar HOG a SAE

5.2.1 Análisis de HOG

Como podemos apreciar en los resultados, para pruebas simples donde no existen rotaciones ni traslaciones, figuras 5.4, 5.5 y 5.6, el método cumple su función y reconoce el contorno de la figura.

Aunque no posee invarianza a rotación ni traslación, para el caso de traslaciones, comparando las figuras 5.3 y 5.6, el vector resultante posee la misma distribución de magnitudes, pero trasladada entre sus componentes. Esto hace posible identificar la figura y reconocer el desplazamiento relativo entre ellas.

En cuanto a robustez frente a ruido, el descriptor sometido a un caso extremo afecta en todo el rango de valores de manera significative, como se puede apreciar en la figura 5.8, por lo que es indispensable aplicar un prefiltrado.

Aplicado sobre escenarios reales, donde en caso muy excepcionales ocurrirán procesos de rotación, los vectores son claramente diferentes, lo que permitirá la futura clasificación de los escenarios, ver figura 5.13.

Comparando los vectores usando el modulo de la diferencia y el ángulo formado por ambos, tabla 5.1, se comprueba la falta de invarianza a rotación y traslación que posee el proceso de extracción de componentes. A pesar de esto, se obtienen buenos resultado sobre figuras incompletas. Sobre escenarios, table 5.3, el error obtenido para las diferentes comparaciones es parecida, incluyendo al mismo scenario, para lo que se obtiene resultados ligeramente más bajos. Esto significa que se situan equiespaciados en el espacio y se produce mucha dispersión.

Finalmente, en cuanto al SAE, el descriptor detecta el mismo contorno principal, donde se produce el mayor cambio de intensidades y por tanto, mayor magnitude en el descriptor, ver figura 5.13. Aún así es posible distinguir pequeñas siluetas que pertenecen al rastro temporal que deja el movimiento de la figura. Dichas siluetas muestran que es posible incluir información temporal a los descriptores.

Por lo general, la dispersión en el espacio es bastante alta, pero se comprueba que existe mayor diferencia para casos completamente distintos.

Tabla 5.1 Comparación de descriptores HOG en figuras geométricas

Comparativa	**Error**	**Ángulo (º)**
Original – Instante diferente	39872.63	54.63
Original – Incompleta	31831.15	42.50
Original – Trasladada	63893.52	85.88
Original – Rotada	58505.55	72.07
Original – Ruidosa	133869.31	61.70
Original – Fig. Diferente	63091.28	81.55
SAE 1 – SAE 2	37245.05	47.19

Tabla 5.2 Comparación de descriptores HOG en escenarios reales

Comparativa	**Error**	**Ángulo (º)**
Testbed 1 – Testbed 2	118724.59	46.66
Hills – Soccer	175533.25	66.96
Hills – Testbed	139177.86	55.20
Soccer – Testbed	134102.60	48.50

5.3 Experimentos con GIST

Al igual que HOG, para el caso de GIST existen varias modificaciones del método original, debido a la pérdida de información por texturas. La idea es utilizar un único nivel en la pirámide y, debido a que el tamaño de la celda depende de la aplicación, en nuestro caso particular de caracterización global de la escena se usará únicamente una celda, que añade invarianza a traslación pero perdiendo detalles a pequeña escala.

Para la obtención de vectores con esta versión se han utilizado 9 longitudes de ondas diferentes, y 20 divisiones angulares.

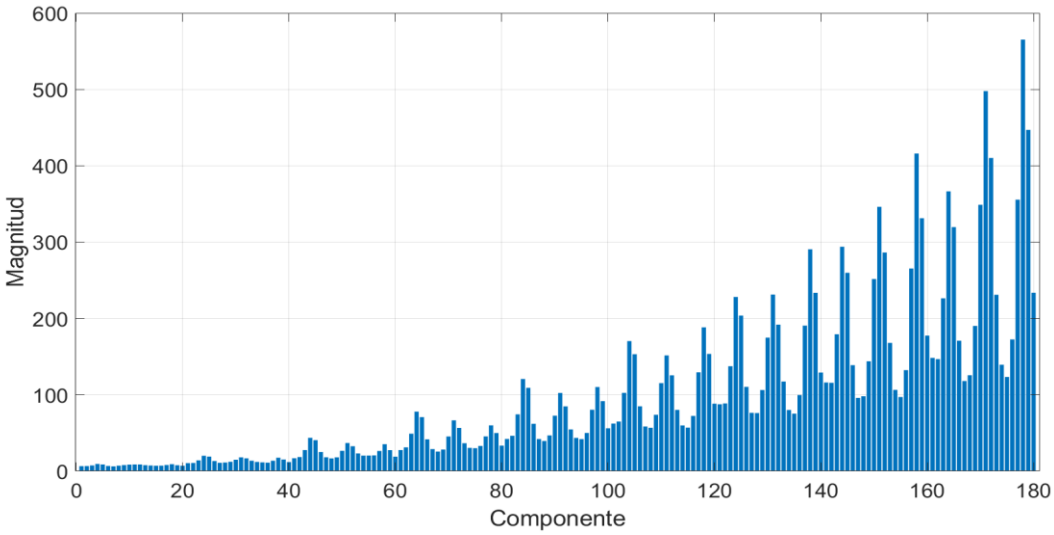


Figura 5.14 Resultado de aplicar GIST a *"triangle"* centrado

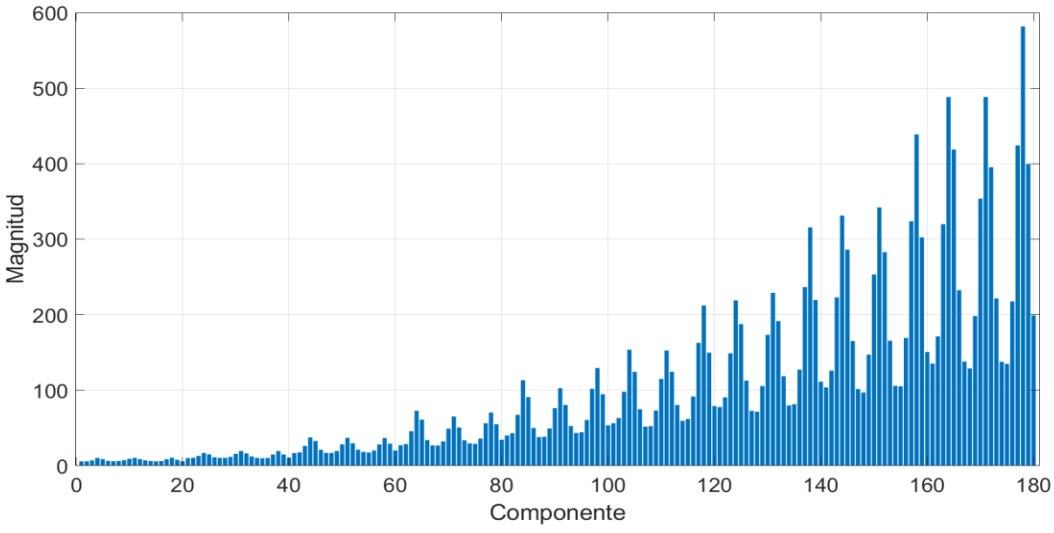


Figura 5.15 Resultado de aplicar GIST a *"triangle"* centrado generado en instante diferente

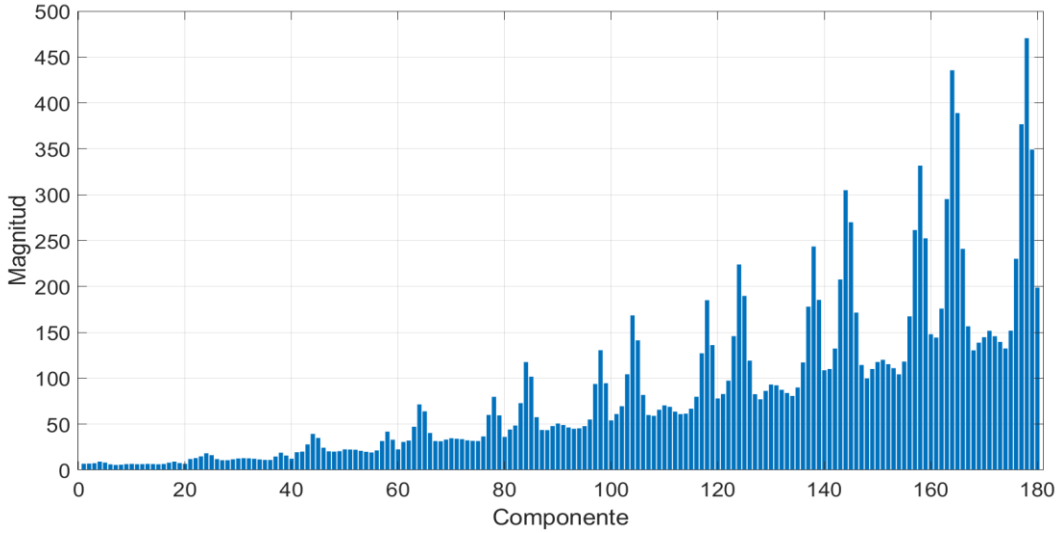


Figura 5.16 Resultado de aplicar GIST a *"triangle"* incompleto

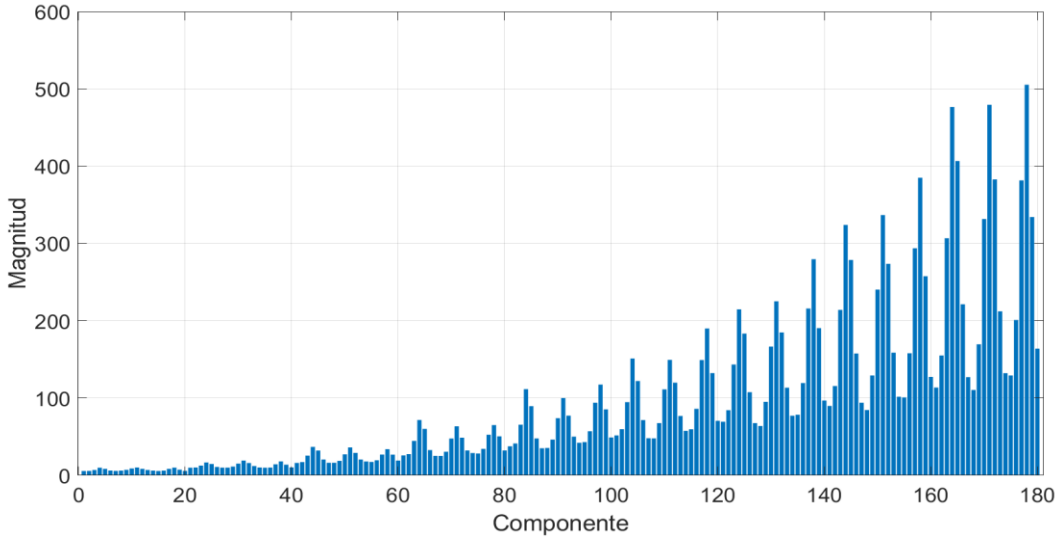


Figura 5.17 Resultado de aplicar GIST a *"triangle"* trasladado

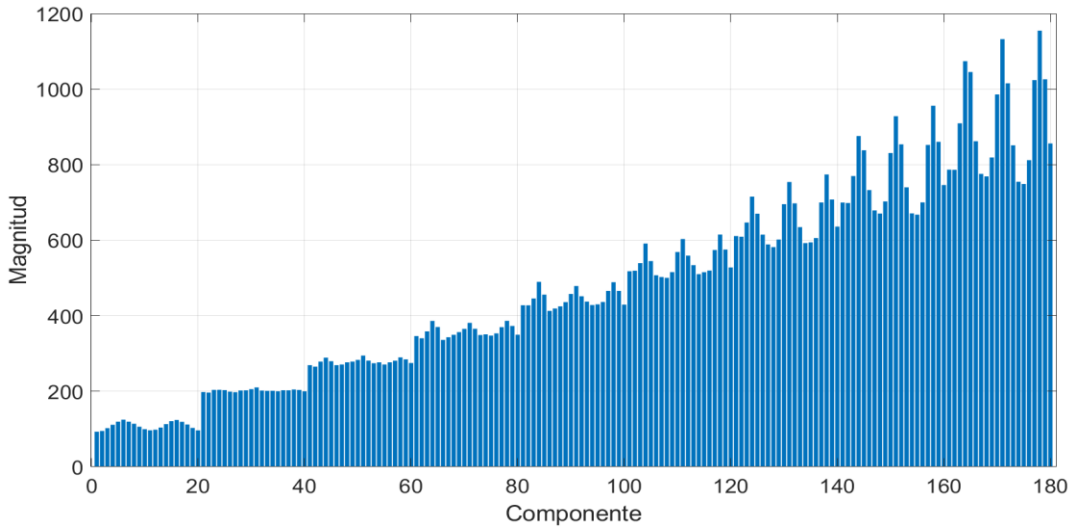


Figura 5.18 Resultado de aplicar GIST a *"triangle"* con ruido

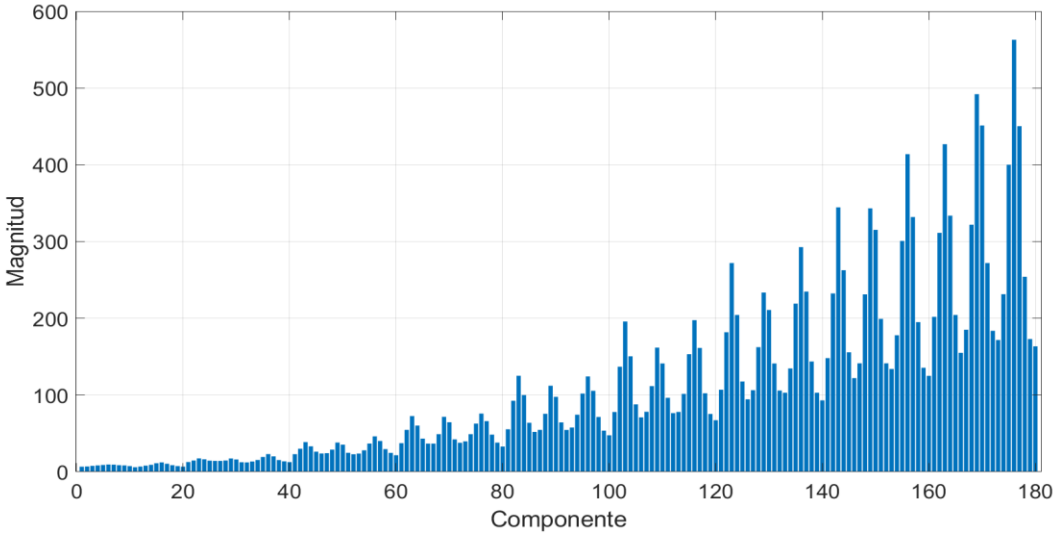


Figura 5.19 Resultado de aplicar GIST a *"triangle"* rotado

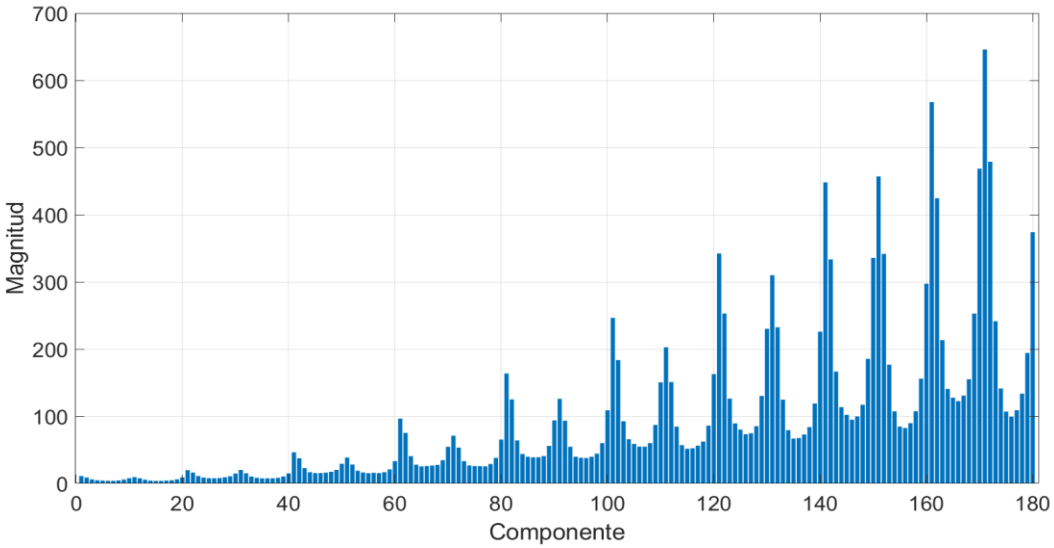


Figura 5.20 Resultado de aplicar GIST a *"square"*

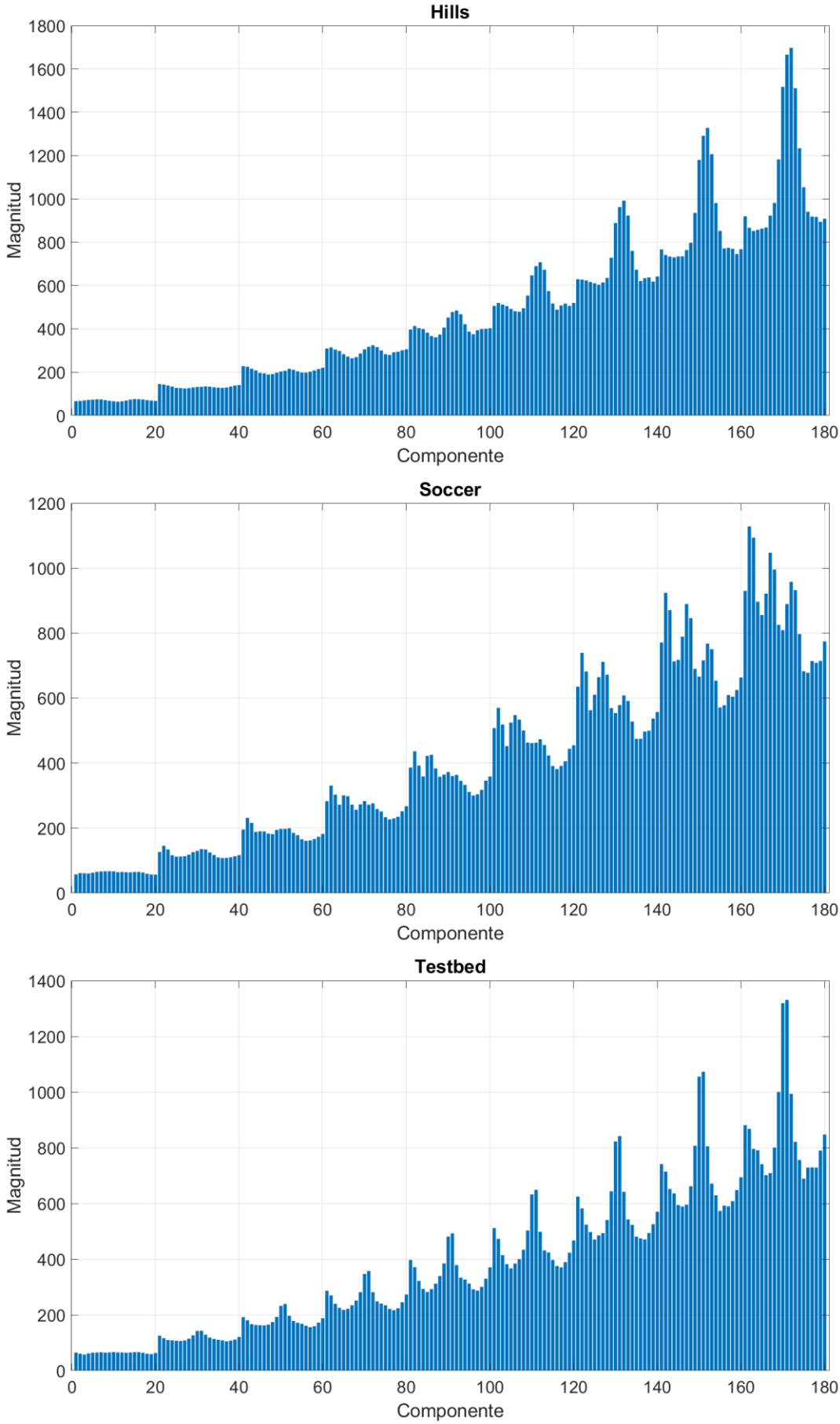


Figura 5.21 Resultado de aplicar GIST a escenarios reales

5.3.1 Análisis de GIST

Viendo los resultados obtenidos, en general, todos presentan la misma distribución de componentes, en la que va aumentando la magnitud a medida que aumenta la longitud de onda del filtro, y se van acentuando aquellas posiciones angulares donde tiene mayor equivalencia con la máscara de Gabor utilizada. Por ello, es posible reducir aún más el descriptor y tomar únicamente los valores finales de este, correspondientes a la mayor longitud de onda utilizada, dónde se acentúan más las características, ver figura 5.22.

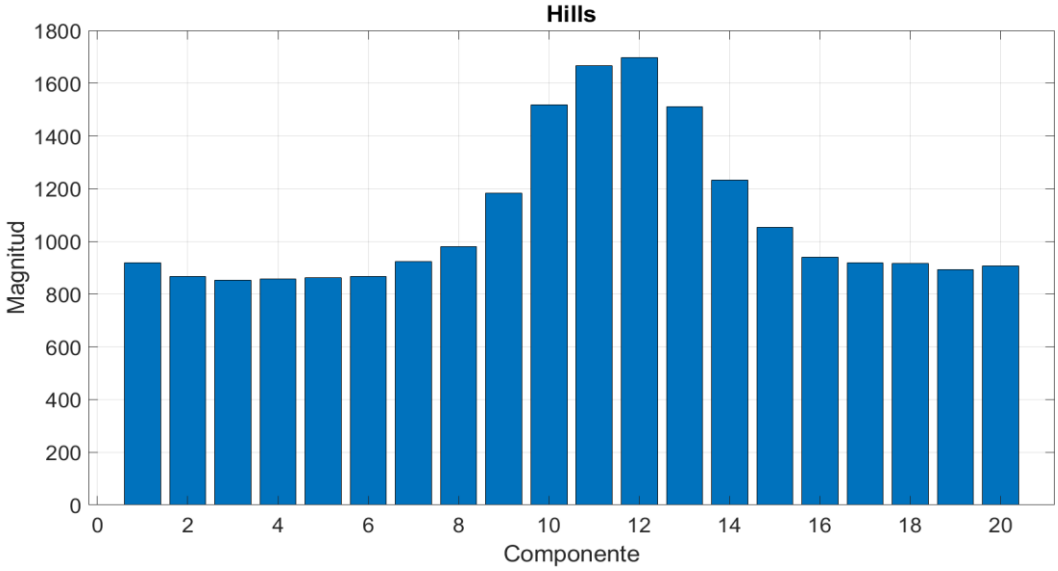


Figura 5.22 Descriptor GIST reducido

Esta versión, como ya se mostró en el desarrollo posee invarianza a traslación y la particularidad de que para rotaciones, se presenta un desplazamiento circular de las componentes dentro de cada longitud de onda. Esto facilita la labor de reconocimiento de formas, ya que cada escena presentará una distribución diferente y particular. De hecho, para el caso de la figura incompleta, se presenta la falta de uno de los tres picos que caracterizan al triángulo, pero manteniendo dicha distribución, ver figura 5.12.

Con respecto al ruido, figura 5.18, es posible distinguir la silueta característica del triángulo, aunque sumergida en un ruido blanco en todas las componentes del vector. Para un caso de ruido menos severo, como encontraremos en una situación real, su comportamiento es bastante robusto y no sería necesario filtrar previamente la imagen.

Cada imagen produce una respuesta diferente en el descriptor, mostrando un correcto funcionamiento, ver figuras 5.14 y 5.20. A esto, se le añade que se puede comprobar con la comparación de vectores, tablas 5.3 y 5.4, que la cercanía entre triángulos es mayor que para el caso de comparar con una figura diferente y entre escenarios. Para el caso de la misma figura rotada se obtienen valores relativamente altos de separación, pero no implica que no sea separable.

Tabla 5.3 Comparación de descriptores GIST en figuras geométricas

Comparativa	**Error**	**Ángulo (º)**
Original – Instante diferente	420.83	7.20
Original – Incompleta	665.68	17.08
Original – Trasladada	173.41	2.97
Original – Rotada	1151.11	31.32

Original – Ruidosa	5479.28	25.22
Original – Fig. Diferente	1354.13	39.45

Tabla 5.4 Comparación de descriptores GIST en escenarios reales

Comparativa	**Error**	**Ángulo (º)**
Testbed 1 – Testbed 2	923.10	7.49
Hills – Soccer	2351.24	14.70
Hills – Testbed	1960.46	9.05
Soccer – Testbed	1449.82	12.82

5.4 Experimentos con DFT

Para la obtención de resultados utilizando la DFT, se ha realizado la reducción de componentes por simetría y eliminando un 30% del descriptor que pertenecen a frecuencias altas que no aportan información real. Además, se ha incluido la reducción de magnitud con la división por la componente de continua.

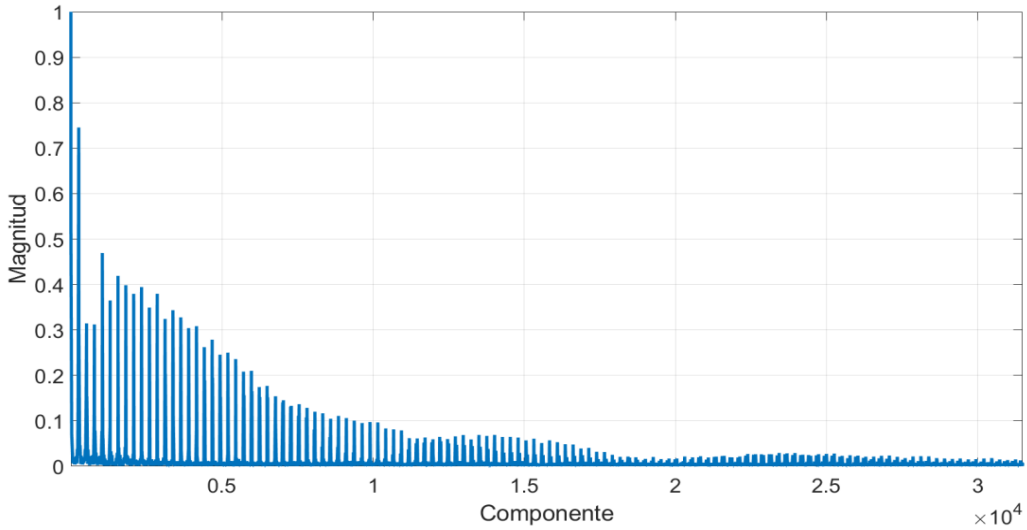


Figura 5.23 Resultado de aplicar DFT a *"triangle"* centrado

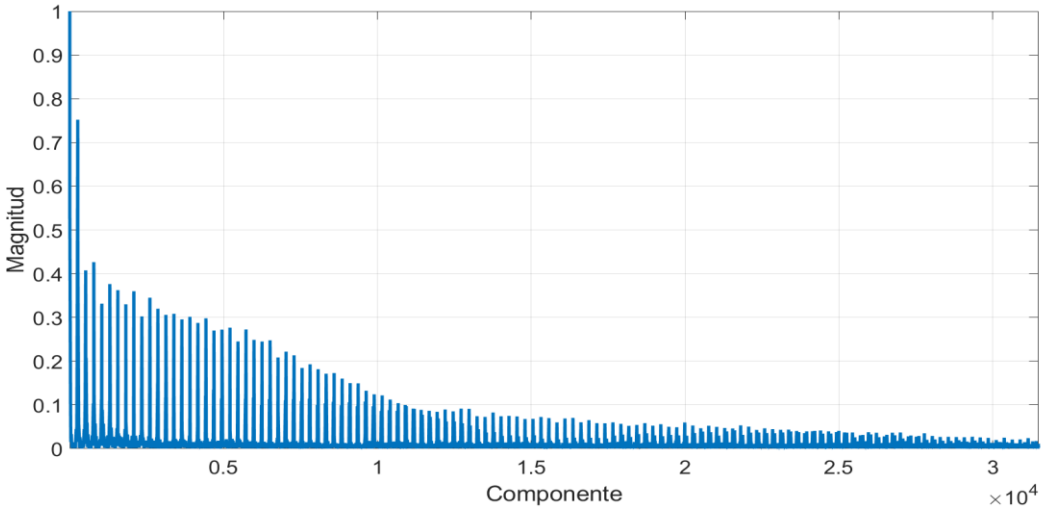


Figura 5.24 Resultado de aplicar DFT a *"triangle"* centrado generado en instante diferente

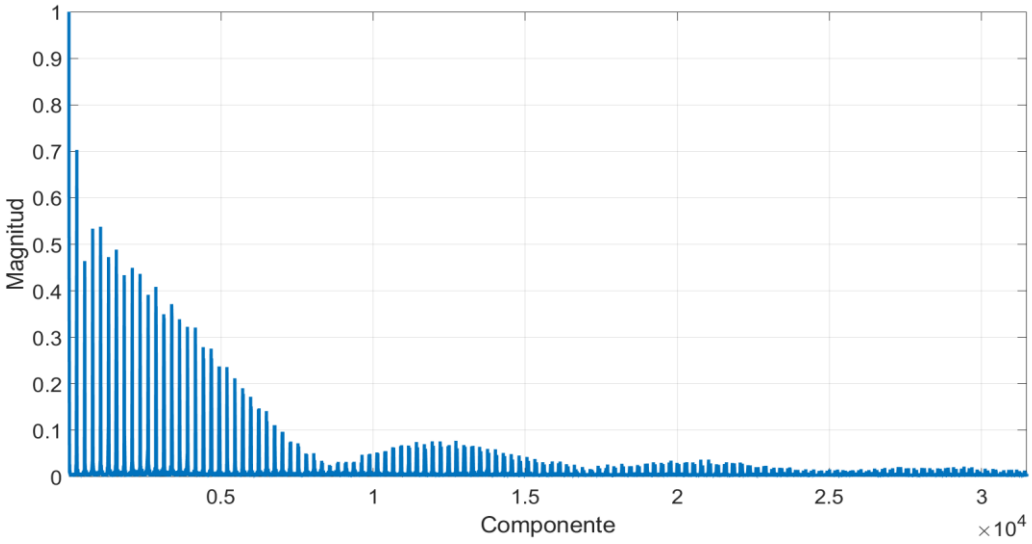


Figura 5.25 Resultado de aplicar DFT a *"triangle"* incompleta

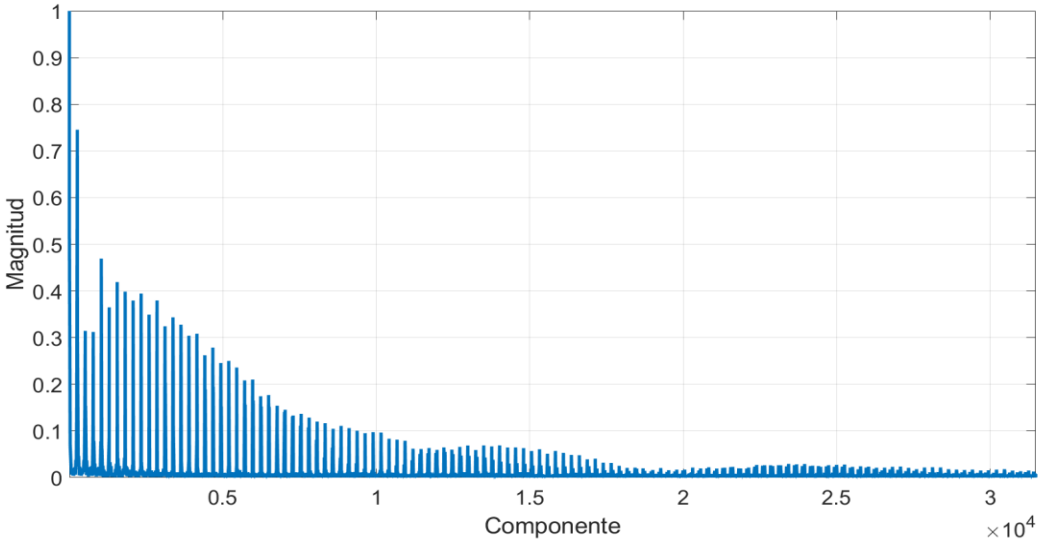


Figura 5.26 Resultado de aplicar HOG a *"triangle"* trasladado

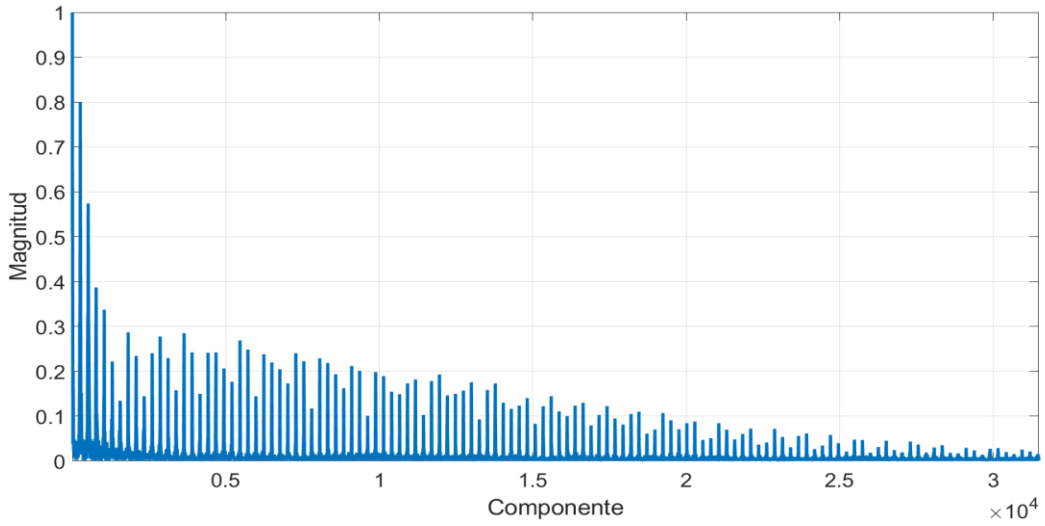


Figura 5.27 Resultado de aplicar DFT a *"triangle"* ruidoso

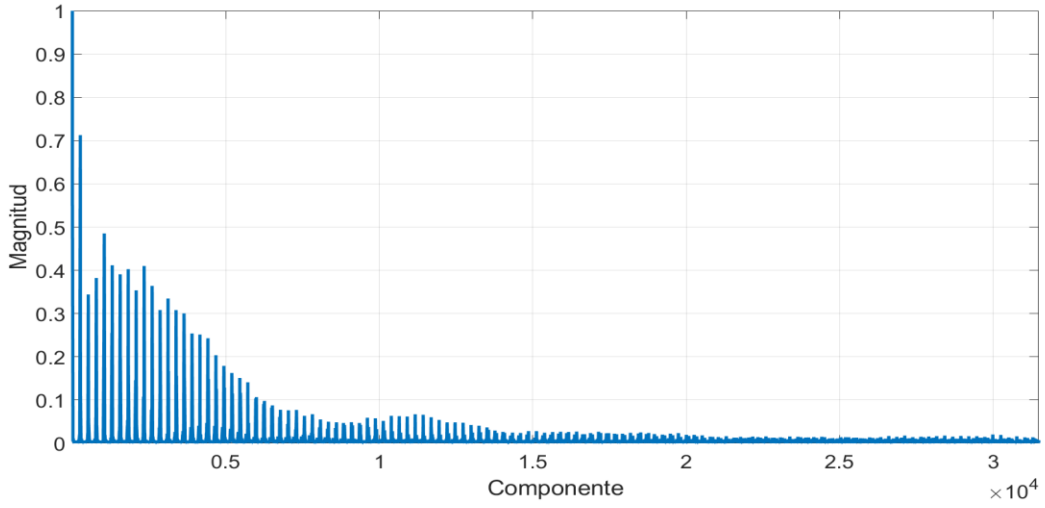


Figura 5.28 Resultado de aplicar DFT a *"triangle"* rotado

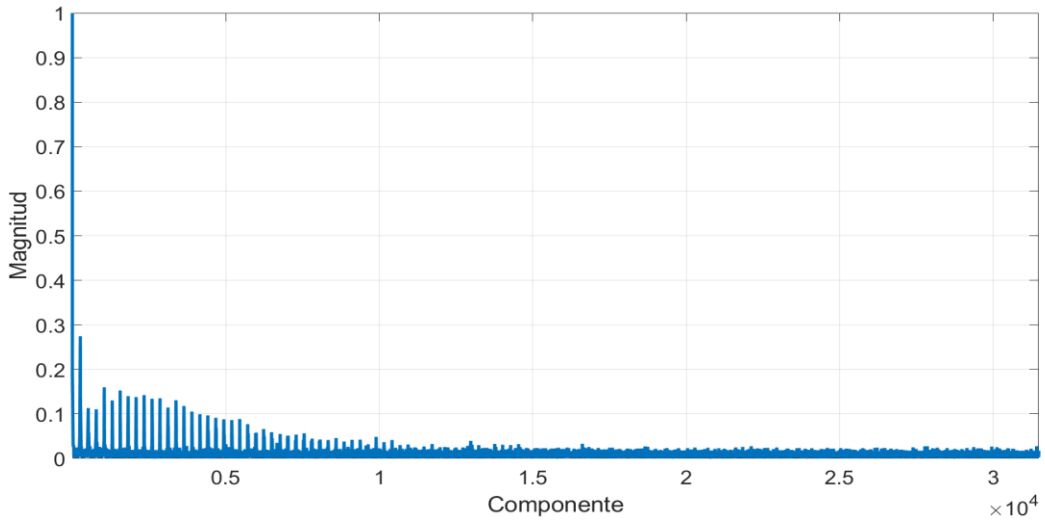


Figura 5.29 Resultado de aplicar DFT a *"square"*

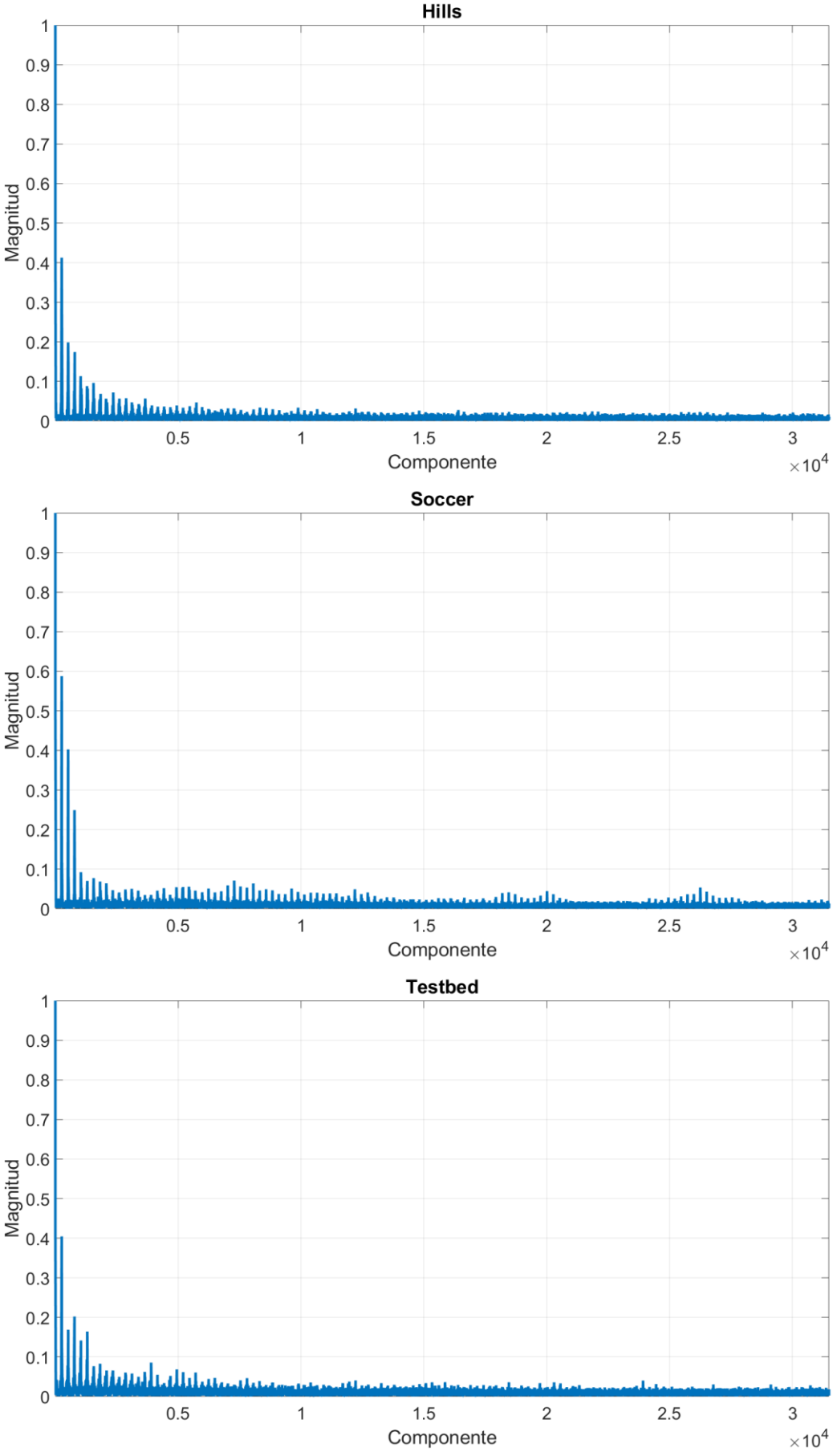


Figura 5.30 Resultado de aplicar DFT a escenarios reales

Consideración de polaridad

Añadir la polaridad a cada imagen permite obtener información de la dirección de movimiento del objeto, aspecto muy interesante a tener en cuenta y que es posible detectar. Para ello se ha tomado una imagen adicional con polaridad para comprobar la respuesta del descriptor y si es posible distinguir el resultado.

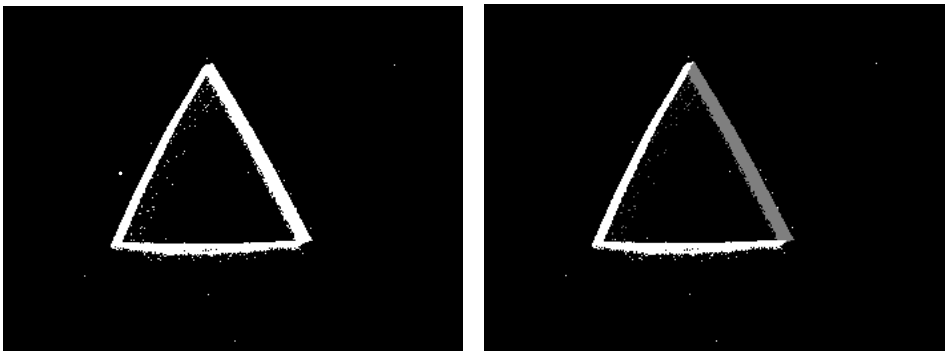


Figura 5.31 Imagen obtenida de *"Triangle"* con y sin considerar polaridad

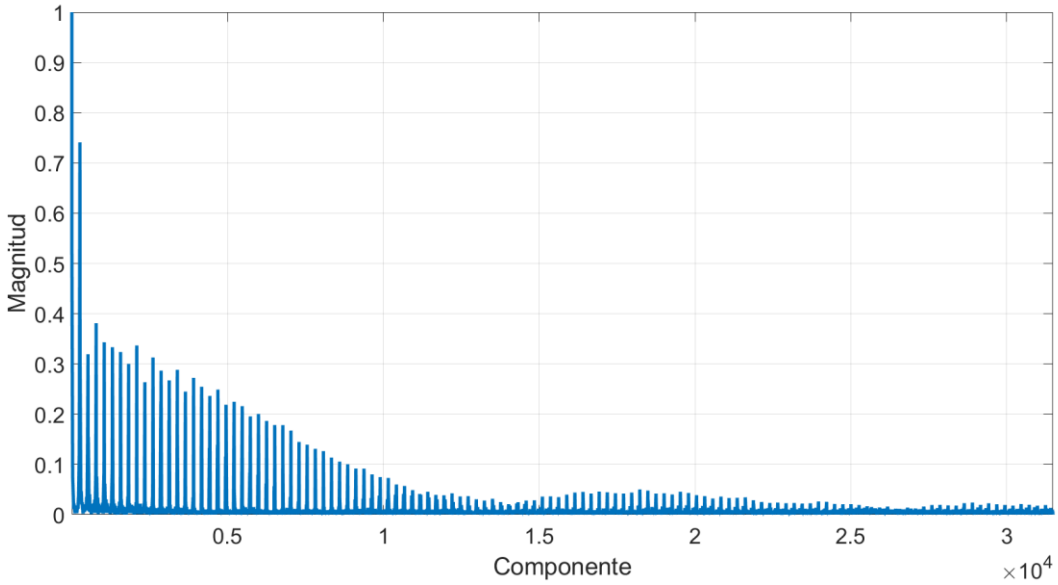


Figura 5.32 Resultado de aplicar DFT a imagen con polaridad

Correlación de fase

En esta sección se mostrarán resultados de aplicar la correlación de fase utilizando el descriptor de la DFT. Se realizarán dos pruebas, una primera con dos imágenes de *"Triangle"* idénticas (pertenecientes al primer set de imágenes mostradas en la introducción), forzando en una de ellas un desplazamiento horizontal de 90 píxeles y vertical de 40 píxeles y ruido, y una segunda con dos imágenes del escenario *"Testbed"* separadas temporalmente.

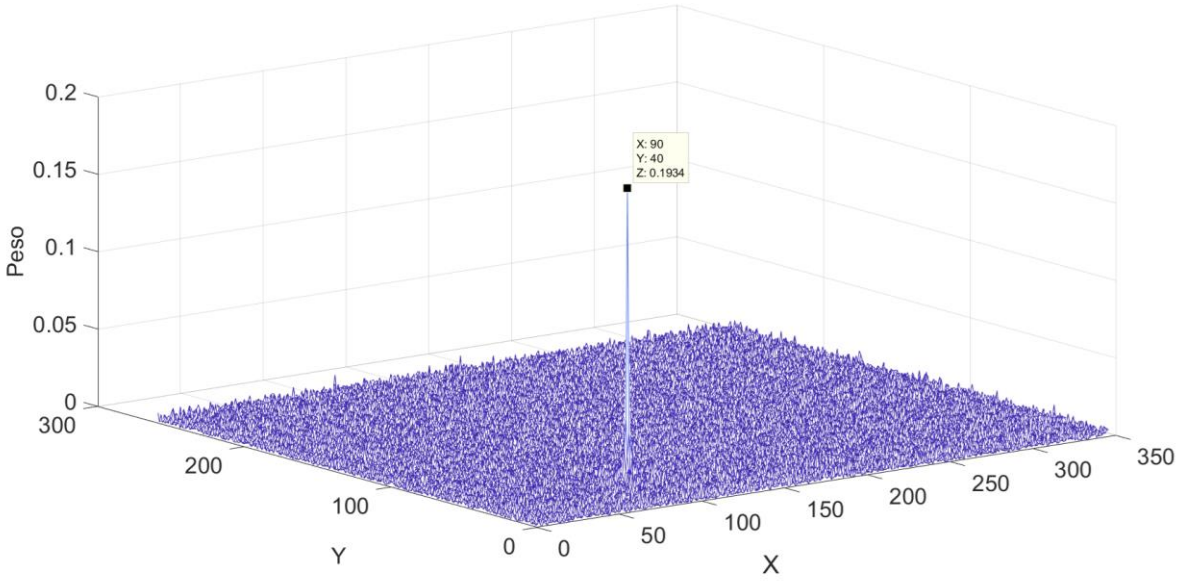
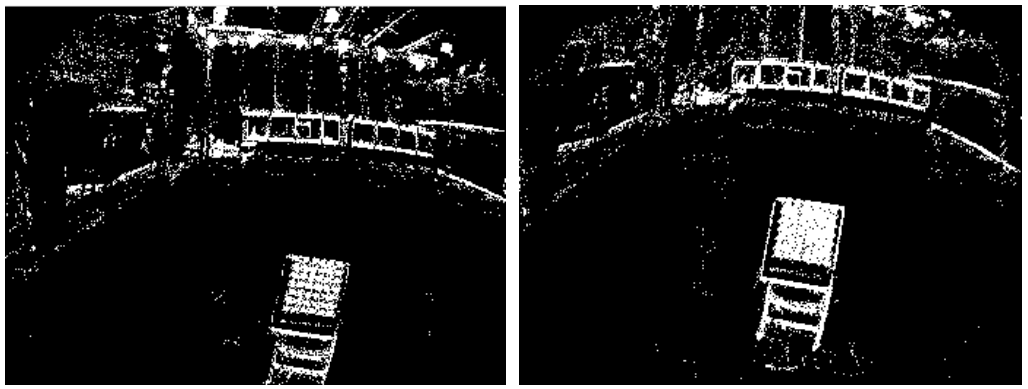


Figura 5.33 Resultado de aplicar correlación de fase en presencia de ruido



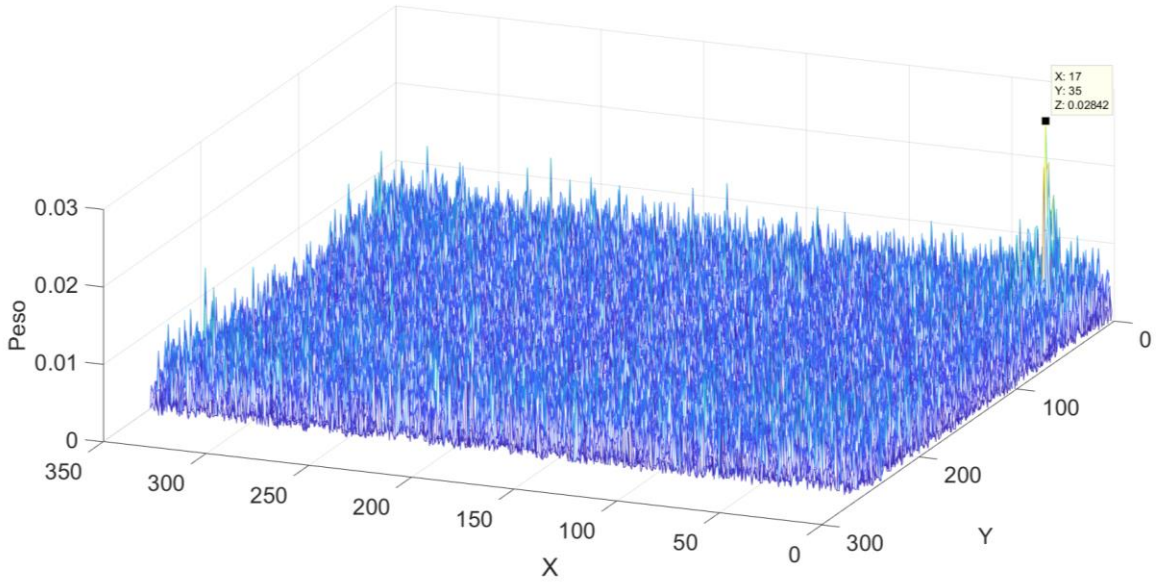


Figura 5.34 Resultado de aplicar correlación de fase en el caso de un escenario real

5.4.1 Análisis de DFT

Tras realizar los diferentes experimentos, se puede apreciar que la distribución en frecuencias para el conjunto de imágenes es diferente, mostrando la capacidad de caracterización que otorga, ver figuras 5.23 – 5.30.

Con respecto al ruído, se conserva la distribución de magnitudes de la imagen original, aunque en general con un valor más bajo, debido a la adición de un ruído blanco, como se aprecia en la figura 5.27. La conservación de dicha distribución es interesante, ya que muestra que aún existiendo ruído el descritpor reconoce la figura.

Usar conjuntamente la DFT con la polaridad muestra resultados diferentes a la salida, ver figura.32. Esto otorga al descriptor no únicamente información del objeto o escenario que se encuentra en la imagen, si no de la dirección de movimiento.

Analizando los módulos y el ángulo entre vectores, tablas 5.5 y 5.6, para este descriptor no es posible conocer de manera clara si será posible la separación espacial del conjunto de datos, ya que poseen valores parecidos. Aún así, se muestra que para triángulos similares se obtienen valores más bajos que para el caso

Tabla 5.5 Comparación de descriptores DFT en figuras geométricas

Comparación	**Error**	**Ángulo (º)**
Instante diferente	1.73	26.89
Incompleta	2.38	33.99
Traslación	1.40e-15	3.5202e-06
Rotación	2.95	48.34
Ruidosa	2.62	38.01
Figura diferentes	3.68	45.08
Polaridad	1.05	15.31

Tabla 5.6 Comparación de descriptores DFT en escenarios reales

Comparativa	**Error**	**Ángulo (º)**
Testbed 1 – Testbed 2	1.65	36.04
Hills – Soccer	3125210.12	38.26
Hills – Testbed	2797146.17	35.71
Soccer – Testbed	2844542.18	36.67

El resultado obtenido tras aplicar correlación para imágenes de contornos nos indica que es posible reconocer la traslación, aunque se posea menor información frecuencial debido a ser imágenes de contorno. Como se

aprecia en la figura 5.33, el método es robusto frente a ruido, consiguiendo un resultado preciso. Para el caso en el que existan pequeñas diferencias entre las imágenes, el resultado es más difuso, ya que no existe un único candidato, como se puede apreciar en la figura 3.34. Una posible solución a calcular el centroide de dicha región, consiguiendo precisión subpixel.

5.5 Conclusiones

Los resultados obtenidos con los tres descriptores son satisfactorios, ya que permiten describir la imagen de una manera global y reconocen el suficiente número de características dentro de ellas para permitir diferenciarlas. En general, ninguno de los tres descriptores muestran a priori robustez frente a rotación de figuras, ya que ninguno posee invarianza a rotación, por lo que será necesario un análisis más complejo para comprobar su eficacia.

Analizando el módulo de la diferencia y el ángulo entre vectores, únicamente GIST permite reconocer qué comparaciones son agrupables espacialmente. Aún así, el criterio utilizado es suficiente pero no necesario, lo que no implica que HOG y la DFT no puedan diferenciar. Esto resultados son producidos principalmente, por el alto número de componentes que poseen dichos descriptores.

Los tiempos de ejecución, mostrados en la tabla 5.7, son muy diferentes y en general, y variarán dependiendo de los parámetros para la generación, que incrementan o disminuyen el cómputo.

Tabla 5.7 Tiempos de ejecución de los descriptores

	HOG	**GIST**	**DFT**
Tiempos de ejecución	3.223 s	2.934 s	0.005 s

Los métodos HOG y GIST han sido desarrollados completamente, mientras que para la transformada de Fourier se ha hecho uso de una función propia de Matlab, que calcula la FFT bidimensional de la imagen. Dicha función posee una mejor optimización y tiempos de procesamiento mucho menores. Por ello, los tiempos de ejecución mostrados en la tabla son meramente informativos para conocer el computo de los métodos desarrollados, de forma que sea posible una mayor optimización en desarrollos futuros. A esto se le añade la poca eficiencia de MATLAB en el cálculo con matrices, lo que empeora aún más los resultados finales.

En general, el factor temporal es importante, ya que, debido a la alta latencia que poseen las cámaras de eventos, es necesario tener métodos que sean capaces de obtener información del conjunto con rapidez. Todo para evitar la pérdida de datos o la degradación de las propiedades temporales de los eventos.

6 CONCLUSIONES Y DESARROLLOS FUTUROS

6.1 Introducción

En este capítulo se muestran las conclusiones finales obtenidas a lo largo de todo el proyecto y las siguientes pruebas para continuar con el estudio de descriptores. Se realizará un resumen de los métodos obtenidos, sus características para posibles aplicaciones y su influencia en cámara de eventos.

Finalmente, se propondrán modificaciones de los descriptores para futuros estudios en conjunto con etapas de postprocesamiento, resaltando los resultados más interesante obtenidos en las diferentes etapas.

6.2 Conclusiones finales

El objetivo del proyecto es conseguir caracterizar imágenes de eventos de manera global, es decir, conseguir agrupar la información de cada imagen en un único descriptor. Para ello, se realizó la búsqueda de información sobre qué descriptores globales tenían más importancia a día de hoy, se estudiaron y se han planteado una serie de modificaciones para su uso sobre eventos.

Dichas modificaciones cumplen su objetivo y aportan información única de cada escena, permitiendo distinguirlas. Además, las cámaras de eventos aportan muchas ventajas frente a las cámaras tradicionales. Por ejemplo, las imágenes ya no son sensibles a las condiciones de iluminación, lo que ahorra el cómputo que conllevaría solucionar este problema de asociación, consiguiendo tiempos de procesamiento menores.

Los resultados obtenidos son un paso importante en visión por computador en eventos, ya se que muestran las diferencias en la descripción global de imágenes tradicionales e imágenes de eventos. Se ha comprobado que es posible aportar información temporal al conjunto utilizando el SAE, con la que es posible conocer la distribución de elementos a lo largo del tiempo y la velocidad de movimiento. Además, el uso de la polaridad, otorga al descriptor información de la dirección en la que se creó la imagen y puede ayudar a conocer la dirección de movimiento del ornitóptero.

Gracias a esto, es posible continuar con la investigación para mejorar los resultados obtenidos y comprobar que, verdaderamente, es posible caracterizar una imagen correctamente, aumentando la sencillez del procesamiento. Con el tiempo, si se mejoran los resultados en descripción de imágenes de eventos, podrían existir aplicaciones en la que prescindir de un sensor tradicional, empleando únicamente un sensor de eventos.

6.3 Desarrollos futuros

6.3.1 Agrupación de eventos y filtrado

Como se ha comprobado, la agrupación de eventos no es trivial y cada una de las propuestas presentaban en general, una serie de ventajas y desventajas frente la anterior.

Todo ello se debe a la dependencia con la escena, que será el factor clave para la elección del método de agrupación, por lo que, se plantea el estudio de métodos dinámicos, que permitan analizar la imagen resultante y modifiquen los parámetros en tiempo real de vuelo para conseguir mejores resultados. Es de gran importancia generar imágenes suficientemente fieles a la escena, ya que la mala agrupación de los eventos conllevará a la imposibilidad de obtener información de esta.

Además, el uso de una matriz de votaciones posee aspectos interesante en general tanto como filtrado como para agrupación de eventos. Con esta idea, es posible obtener un nuevo descriptor local, basado en el funcionamiento por votos y que puede permita obtener puntos equivalentes para imágenes obtenidas en localizaciones diferentes, aprovechando la alta latencia de las cámaras de eventos y reduciendo el computo.

6.3.2 Modificación de descriptores

Los descriptores desarrollados poseen buenos resultados en su aplicación con eventos y permiten la distinción entre imágenes, tras un primer análisis sencillo basados en el módulo y la observación directa de los resultados. Aún así, los tiempos de ejecución no son los deseados, por lo que es necesario desarrollar técnicas de

Por ello, es necesario un estudio más profundo utilizando clasificadores como SVM, para comprobar que realmente se pueden separar en el espacio. En conjunto a esto, se le pueden añadir procesos de reducción de dimensiones utilizando el método de PCA, permitiendo visualizar en dos o tres dimensiones la disposición de los puntos en el espacio y el coste computacional sobre la etapa de clasificación en intervenciones reales.

HOG

La obtención del gradiente para la aplicación sobre el SAE es interesante, ya que permite conocer las direcciones de movimiento de un elemento o una escena completa. De hecho, se verifica con los resultados obtenidos. Por ello, es necesario continuar con su estudio para conseguir extraer dicha información que se ha añadido al descriptor.

GIST

De los tres descriptores desarrollados, la modificación de GIST planteada en los análisis muestra un buen comportamiento, y permite desligar la necesidad de utilizar descriptores locales para general descriptores globales. Este avance es importante y provoca el nacimiento de nuevas investigación en búsqueda de conseguir el mismo resultado con otros métodos.

DFT

El tratamiento frecuencial en imágenes en eventos posee características importantes y permite el tratamiento de los eventos desde otro punto de vista, lo que puede facilitar el entendimiento y el tratamiento de estos con técnicas innovadoras.

Una posible aplicación para el estudio es la capacidad de eliminar objetos dentro de una escena compleja, conociendo la distribución en frecuencias del objeto en particular, independientemente de dónde se sitúe esta en la imagen, ver figura 6.1. Es interesante poder eliminar las frecuencias no deseadas que pertenecen a elementos que no aportan información en la intervención de nuestro robot, por lo que es necesario un estudio en profundidad de esta técnica.

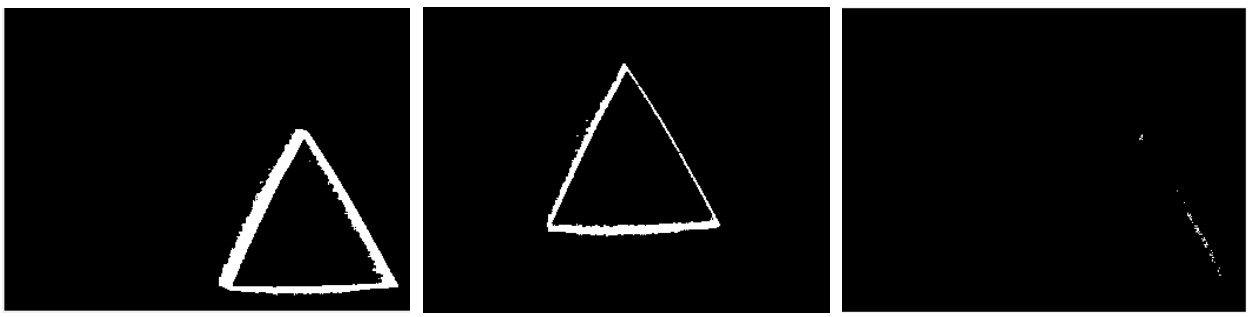


Figura 6.1 Resultado de eliminar frecuencias en una imagen de eventos

Como se aprecia en la figura 6.1, utilizando la imagen del triángulo sin trasladar es posible eliminar las frecuencias pertenecientes al triangulo trasladado conservando la posición original.

Por último, tras los resultados obtenidos utilizando la DFT para relacionar imágenes trasladadas, se ha comprobado que el uso de esta técnica posee buenos resultados en imágenes con eventos, lo que puede ayudar a realizar un seguimiento espacial del vuelo del ornitóptero, relacionando diferentes imágenes y obteniendo una matriz de transformación que indique el desplazamiento.

Para conseguir un resultado más preciso, es posible conseguir también correlación angular y a escala, con un procesamiento más complejo que el desarrollado en este proyecto.

Al igual que los métodos explicados en este proyecto, existen multitud de otras opciones de descripción global que podrían llegar a aportar mejores resultados, por lo que es materia de investigación aplicar procedimientos similares al utilizado en este proyecto para su comprobación.

REFERENCIAS

[1] GALLEGO, Guillermo, et al. Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*, 2019.

[2] GRIFFIN, «https://griffin-erc-advanced-grant.eu/,» GRVC. [En línea].

[3] MATLAB. [En línea].

[4] HARRIS, Chris, et al. A combined corner and edge detector. En *Alvey vision conference*. 1988. p. 10-5244.

[5] NOBLE, J. Alison. Finding corners. *Image and vision computing*, 1988, vol. 6, no 2, p. 121-128.

[6] TRAJKOVIĆ, Miroslav; HEDLEY, Mark. Fast corner detection. *Image and vision computing*, 1998, vol. 16, no 2, p. 75-87.

[7] BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. Surf: Speeded up robust features. En *European conference on computer vision*. Springer, Berlin, Heidelberg, 2006. p. 404-417.

[8] LOWE, David G. Object recognition from local scale-invariant features. En *Proceedings of the seventh IEEE international conference on computer vision*. Ieee, 1999. p. 1150-1157.

[9] PAYÁ, Luis, et al. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors*, 2014, vol. 14, no 2, p. 3033-3064.

[10] PAYÁ, Luis, et al. Using omnidirectional vision to create a model of the environment: A comparative evaluation of global-appearance descriptors. *Journal of Sensors*, 2016, vol. 2016.

[11] ALZUGARAY, Ignacio; CHLI, Margarita. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 2018, vol. 3, no 4, p. 3177-3184.

[12] VASCO, Valentina; GLOVER, Arren; BARTOLOZZI, Chiara. Fast event-based Harris corner detection exploiting the advantages of event-driven cameras. En *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016. p. 4144-4149.

[13] MUEGGLER, Elias; BARTOLOZZI, Chiara; SCARAMUZZA, Davide. Fast event-based corner detection. 2017.

[14] LI, Ruoxiang, et al. Fa-harris: A fast and asynchronous corner detector for event cameras. En *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019. p. 6223-6229.

[15] M. Týč, «https://sthoduka.github.io/imreg_fmt/docs/overall-pipeline/,» 18 May 2017. [En línea].

[16] THOMAS, G. A. Television motion measurement for DATV and other applications. *NASA STI/Recon*

Technical Report N, 1987, vol. 88, p. 13496.

[17] ABDOU, Ikram E. Practical approach to the registration of multiple frames of video images. En *Visual Communications and Image Processing'99*. International Society for Optics and Photonics, 1998. p. 371-382.

[18] CHEN, Guang, et al. Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 2020, vol. 37, no 4, p. 34-49.

[19] DALAL, Navneet; TRIGGS, Bill. Histograms of oriented gradients for human detection. En *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Ieee, 2005. p. 886-893.

[20] OLIVA, Aude; TORRALBA, Antonio. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 2001, vol. 42, no 3, p. 145-175.

[21] MANJUNATH, Bangalore S.; MA, Wei-Ying. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 1996, vol. 18, no 8, p. 837-842.

[22] MENEGATTI, Emanuele; MAEDA, Takeshi; ISHIGURO, Hiroshi. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 2004, vol. 47, no 4, p. 251-267.

GLOSARIO

CPS
Cross Power Spectrum 12, 43
DFT
Discrete Fourier Transform 9, 12, 27, 37 - 42, 59 - 69
DVS
Dynamic Vision Sensor 1
eFAST
event Features from Accelerated Segment Test 10
FAST
Features from Accelerated Segment Test 11
FFT
Fast Fourier Transform 38
FS
Fourier Signature 9
FT
Fourier Transform 37
GRVC
Grupo de Robótica, Visión y Control 1
HOG
Histogram of Oriented Gradients 8, 27, 28, 31, 32, 34, 36, 48 - 53, 60, 66, 68
LP
Low Pass 8
PCA
Principal Components Analysis 5, 7, 11, 68
RGB
Red, Green y Blue 5
ROS
Robot Operating System 3
SAE
Surface of Active Events 10, 11, 22, 23, 31, 32, 51, 52, 53, 67, 68
SIFT
Scale Invariant Feature Transform 7
SLAM
Simultaneous Localization And Mapping 7
SSD
Sum of Square Differences 6
SURF
Speeded-Up Robust Features 7
UAV
Unmanned Aerial Vehicle 1