

UNIVERSIDAD DE ORIENTE

Facultad de Matemática y Computación



RECUPERACIÓN BASADA EN CONTENIDO EN IMÁGENES DE RAYOS X MEDIANTE HISTOGRAMAS DE PALABRAS VISUALES

Tesis en opción del título de
Máster en Ciencia de la Computación

Autor: Ing. Felipe Rodríguez Arias

Tutores: DrC. Enrique J. Marañón Reyes

Santiago de Cuba, 2015

RESUMEN

Las imágenes de rayos X constituyen una importante tecnología para aplicaciones de seguridad, especialmente en sitios donde la exploración de bultos, mercancía o valijas en busca de objetos peligrosos es de vital importancia. Según los especialistas la inspección de equipajes mediante imágenes de rayos X presenta limitantes en cuanto a la eficiencia en el reconocimiento de objetos peligrosos y la demora que toma este procedimiento. Recientemente se han llevado a cabo importantes investigaciones en esta área del conocimiento, sin embargo las implementaciones realizadas se han centrado sólo en mejorar el porcentaje de reconocimiento de objetos peligrosos dejando a un lado factores determinantes en la implantación de una solución informática tales como el costo computacional y la portabilidad del código generado. En este trabajo se presenta el desarrollo de un componente de software escrito en C/C++ que describe un procedimiento basado en el algoritmo Saco de Palabras Visuales para la detección de armas cortas. En los experimentos realizados se obtiene una razón de verdaderos positivos de un 93.02%. Además se logra reducir hasta un 68.79% el tiempo de ejecución del algoritmo a través de técnicas de programación multiproceso mediante OpenMP. Los resultados obtenidos demuestran que los valores de acierto del método implementado son coherentes con respecto a otros estudios realizados. De la misma forma exponen la factibilidad del uso de la programación concurrente en procesadores con varios núcleos con el propósito de alcanzar tiempos de ejecución más rápidos en un hardware convencional.

Palabras clave: imágenes de rayos X, Saco de Palabras Visuales, OpenMP, rendimiento, C/C++.

ABSTRACT

X-ray images are an important technology for security applications, especially where the exploration of packages or bags for dangerous objects is vital. According to specialists baggage inspection by X-ray images has limitations regarding efficiency in recognizing dangerous objects and delay takes this procedure. Recently have carried out important research in this area of knowledge, however implementations made have focused only on improving the percentage of recognition of dangerous objects aside determining factors in the implementation of a solution such as computational cost and portability of the generated code. This paper describes the development of a software component written in C / C ++ which describes a method based on the Bag of Visual Words algorithm for detection of small arms is presented. In experiments it is obtained a ratio of true positives to 93.02%. In addition it is possible to reduce up to 68.79% time algorithm execution through multithreaded programming techniques using OpenMP. The results show that the values of success of implemented method are consistent with respect to other studies. Similarly demonstrate the feasibility of using concurrent programming on multi-core processors in order to achieve faster execution times in a conventional hardware.

Keywords: *X-ray images, Bag of Visual Words, OpenMP, performance, C/C++*

ÍNDICE

Introducción	1
Capítulo 1: Marco teórico contextual	10
1.1 Introducción	10
1.2 Equipos de inspección por rayos X.....	11
1.2.1 Sistemas de transmisión de energía simple	12
1.2.2 Sistemas de transmisión de doble energía	13
1.3 Representación de la materia en las imágenes de rayos X.....	15
1.4 Algoritmo Bags of Visual Words.....	16
1.4.1 Extracción de características	18
1.4.2 Agrupación de características	22
1.4.3 Cuantificación vectorial.....	24
1.4.4 Etapa de clasificación	25
1.5 Evaluación del desempeño.....	28
1.6 Rendimiento de computadores paralelos.....	29
1.7 Metodología de desarrollo de software.....	30
1.8 Herramientas y tecnologías.....	32
1.8.1 Lenguaje de programación C++.....	32
1.8.2 Biblioteca de funciones VLFeat 0.9.18	32
1.8.3 Visual Paradigm 6.8.....	33
1.8.4 Doxygen 1.8.5.....	33
1.8.5 OpenCV 2.4.11.....	33
1.9 Conclusiones parciales	34
Capítulo 2: Análisis y Diseño del Sistema.....	36
2.1 Análisis de las variables de configuración del algoritmo BoVW	36
2.1.1 Tamaño del vocabulario de palabras visuales.....	36
2.1.2 Información espacial	37
2.1.3 Asociación de pesos en los histogramas de palabras visuales	39
2.2 Historias de usuario.....	40
2.3 Funcionalidades del sistema	42
2.4 Historias técnicas.....	43
2.5 Usuarios del sistema	44
2.6 Patrones de diseño.....	44
2.6.1 Singleton.....	45
2.6.2 Builder	45

ÍNDICE

2.6.3 GRASP	46
2.7 Diagrama de clases.....	47
2.8 Conclusiones parciales	49
Capítulo 3: Implementación y Resultados	50
3.1 Estructura identificada del formato de imagen IMG de equipos de rayos X.....	50
3.1.1 Estructura de la cabecera.....	51
3.1.2 Estructura del cuerpo de datos de la imagen	53
3.1.3 Estructura de la cola de datos.....	55
3.2 Entrenamiento del clasificador	55
3.3 Descripción de la base de datos de pruebas.....	57
3.4 Análisis de la clasificación en la fase online del algoritmo BoVW.....	58
3.5 Análisis de rendimiento de la etapa online del algoritmo BOVW.....	59
Conclusiones	64
Recomendaciones	65
Anexos.....	66
Anexo 1 – Aceptación de los resultados por parte del cliente.....	66
Anexo 2 - Diagrama de clases, namespace bovw.....	67
Anexo 3 – Diagrama de clases, namespace img.....	68
Bibliografía	69

INTRODUCCIÓN

Desde que el físico Wilhelm Conrad Röntgen descubrió en 1895 que mediante los rayos X se podían identificar las estructuras internas de los objetos, estos han sido utilizados en un sin número de aplicaciones que van desde las imágenes médicas hasta los ensayos no destructivos de materiales, donde el objetivo es analizar las partes internas que son indetectables a simple vista. Los sondeos no intrusivos en objetos a través de los rayos X, se aplican en varios contextos tales como: el análisis de productos alimenticios, la inspección de equipajes y de piezas de automóviles, el control de calidad de las soldaduras, entre otros [1]. Las pruebas de rayos X por lo general consisten en la medición de características específicas de los objetos, con el fin de detectar, reconocer o evaluar peculiaridades buscadas o no deseados de sus partes internas.

Los equipos de inspección por rayos X convencionales miden la energía de los rayos X atenuada después de pasar a través del objeto de exploración. El sistema crea una imagen digital de alta resolución en escala de grises con la cual el operador de seguridad, apoyado en sus conocimientos previos de las formas y las características distintivas de los diferentes objetos vistos a través de una imagen de rayos X, puede identificar armas, explosivos, drogas y artículos de contrabando, por citar algunos [2] [3]. En la actualidad nuevos dispositivos de imágenes de rayos X se han desarrollado; entre ellos se encuentran los equipos de análisis de energía dual que permiten estimar los números atómicos de los materiales. Luego de un posterior procesamiento, la imagen resultante es coloreada, lo que facilita y mejora en parte la visualización de los elementos [4] [5]. A través de este método se obtiene también una medida de la densidad y el grosor del material.

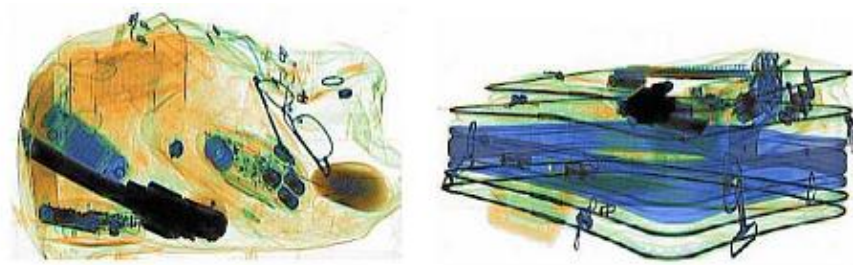


Figura. 1: Dos ejemplos de imágenes de rayos-X que contienen armas de fuego. Son difíciles de interpretar debido al inusual punto de vista.

Sin embargo, las imágenes de rayos X son muy diferentes de las imágenes del espectro visible. Entre las características que las distinguen se puede observar que son imágenes transparentes, los valores de píxel representan la atenuación de múltiples objetos, poseen un alto grado de desorden, pueden resultar ruidosas debido a la baja energía de algunos rayos emitidos por el equipo, y los objetos pueden encontrarse desde inusuales puntos de vista (figura 1), de manera que la interpretación del contenido de las imágenes se torna un problema complejo [6].

Las imágenes de rayos X constituyen una importante tecnología para aplicaciones de seguridad, especialmente en aeropuertos y otros puntos donde la exploración de bultos o equipaje en busca de objetos o sustancias peligrosas se ha convertido en un tema de gran importancia [4]. El auge de incidentes terroristas ocurridos en los últimos años en el mundo; así como el aumento del contrabando de drogas y de medios para llevar a cabo acciones ilegales; han dado lugar a chequeos más exhaustivos y restricciones en el equipaje de mano lo que conlleva a grandes congestiones en las terminales aeroportuarias, retrasos en las salidas de los vuelos y molestias de los pasajeros. Todos estos factores han añadido coste a los viajes aéreos y tienen impactos socioeconómicos negativos.

Una de las prioridades de seguridad en puertos y aeropuertos nacionales e internacionales, radica en controlar la entrada y salida de toda clase de artículos; pero este procedimiento con la utilización de equipos de inspección por rayos X, está siendo afectado por una elevada estadística de falsos positivos (alarmas emitidas cuando no hay elementos peligrosos), según los expertos de la Aduana General de la República (AGR). Por esta razón es normal que se vuelven a inspeccionar alrededor de la tercera parte de las imágenes de los bultos escaneados [7]. El control de equipajes según los funcionarios cubanos de seguridad aeroportuaria, se realiza en dos niveles: el primero se lleva a cabo por los operadores de los equipos en los Puntos de Inspección y Registro (PIR) y el segundo por parte de funcionarios de mayor experiencia en el Punto de Control, verificando el trabajo de los operadores. Debido al enorme flujo de imágenes a procesar y al agotamiento visual, el operador del equipo debe ser relevado de su puesto de trabajo cada cierto tiempo. Teniendo en cuenta la enorme cantidad de equipaje que deben ser inspeccionados, especialmente en los aeropuertos, es deseable automatizar o facilitar en alguna medida el proceso de inspección para reducir la carga de trabajo de los examinadores y por lo tanto aumentar la velocidad de inspección.

Introducción

En Cuba, la marca comercial de la mayoría de los equipos de inspección por rayos X utilizados en el chequeo de equipaje pertenece al fabricante NUCTECH, de procedencia China. Las funcionalidades disponibles en estos sistemas de inspección según [8] se dividen en dos grupos: Funciones básicas, que son filtros elementales del procesamiento de imágenes, tales como llevar a escala de grises, resaltar bordes, seleccionar una región de interés (ROI) y hacerle un acercamiento (zoom) entre otras; y funciones avanzadas, que representan algoritmos de procesamiento de imágenes un poco más allegados a las necesidades de los inspectores, las cuáles son:

- Procesar la proyección de la imagen.
- Ofrecer ayuda para identificar sustancias explosivas, armas o drogas.
- Alarma de zona impenetrable.

Pero estas tres funcionalidades en sí no responden a los intereses del servicio, porque no son eficaces en su desempeño y proporcionan muchas alarmas cuando no hay objetos peligrosos en la detección, por lo que en la mayoría de los casos se encuentran desactivadas.

Con el objetivo de enfrentar estas necesidades, a nivel mundial se han propuesto varios enfoques, que van desde el procesamiento de señales digitales hasta el mejoramiento de los equipos de inspección con nuevas potencialidades. Ejemplo de esto último resulta en la utilización de la tomografía computarizada para separar objetos en imágenes de rayos X [9]. Esta técnica intenta reconstruir los objetos en 3D para permitir el análisis desde varios puntos de vista, lo cual, si bien mejora la visualización, requiere una enorme sobrecarga computacional y consume mucho tiempo lo que limita su aplicación en tiempo real.

Respecto a las principales áreas en las que se aplican técnicas de visión computacional en imágenes de rayos X se encuentran la detección de explosivos, drogas y armas ocultas, mediante diferentes métodos de segmentación de regiones de interés, extracción de características y clasificación [10]. De forma general existen pocas investigaciones sobre el tema, en parte debido a la imposibilidad de los investigadores de acceder a grandes cantidades de imágenes de este tipo que contengan objetos de interés. Debido a esto, en la actualidad se requieren mejoras en las técnicas existentes para satisfacer las necesidades de seguridad. Proveer un mecanismo de detección más

fiable, con pocos falsos positivos, a una velocidad aceptable, con una mayor asistencia a los operadores resulta un reto de investigación real.

Sobre la base de los hechos y las problemáticas descritas anteriormente se realizó un estudio del estado del arte y diversas entrevistas a directivos y especialistas de la aduana en Ciudad de la Habana y Santiago de Cuba que laboran en el Aeropuerto Internacional José Martí, el Aeropuerto Internacional Antonio Maceo y el Puerto de Santiago de Cuba. Del procesamiento de toda la información obtenida se pudieron develar las siguientes **manifestaciones**:

- La inspección de equipaje mediante equipos de rayos X es una actividad que comprende una alta especialización de los inspectores de seguridad.
- En algunas ocasiones el proceso de inspección es realmente lento, según las características de los pasajeros y los equipajes, lo cual provoca mucho desgaste en los operadores de los equipos e inconformidades en los pasajeros.
- El software presente en los sistemas de inspección aduaneros solo contiene algoritmos básicos de procesamiento de imágenes.
- Las imágenes de rayos X contienen características distintivas que las diferencian de otros tipos de imágenes. Exhiben de forma desordenada los objetos de la muestra que representan, los valores de píxel simbolizan la atenuación de múltiples objetos superpuestos y los objetos representados se observan desde inusuales puntos de vista, lo cual dificulta su interpretación.

Todo esto, llevó a un equipo de investigadores y desarrolladores del Centro de Estudios de Neurociencias, Procesamiento de Imágenes y Señales (CENPIS) de la Facultad de Ingeniería Eléctrica de la Universidad de Oriente, en conjunto con otras entidades pertenecientes a la Aduana General de la República (AGR) y al Ministerio del Interior (MININT) a plantearse la necesidad de alguna vía de solución.

Luego de analizar las manifestaciones reveladas y desde la cultura cognitiva y experiencia del postulante, se identifica como **problema de la investigación**, las limitaciones existentes en el proceso de inspección a través de equipos de rayos X, que restringen la identificación de objetos ilícitos y la rapidez del procedimiento en los controles de seguridad.

Entre las **causas** principales de este problema, obtenidas a través del estudio realizado, se revelan:

- Existencia de gran cantidad, en el orden de los miles, de tipos de equipajes y artículos que deben ser chequeados en el proceso de inspección.
- La rapidez en el análisis de las imágenes de rayos X depende de la experiencia de los especialistas. Se necesita entrenamiento de al menos un año para empezar a dominar esta tarea.
- No se tiene conocimiento acerca del formato de imagen IMG que generan los equipos de inspección por rayos X del fabricante NUCTECH.
- No se emplean herramientas de software que posibiliten la identificación de materiales ilícitos en las imágenes de rayos X.

Las causas precisadas indican la necesidad de perfeccionar el proceso de análisis del equipaje en imágenes de rayos X mediante técnicas de visión computacional, el cual se asume como **objeto de la investigación**.

En tal sentido la literatura sobre el tema se basa fundamentalmente en una variedad de propuestas que se centran en la mejora de la imagen y la segmentación. Chen y otros desarrollaron un método combinatorio para fusionar, eliminar el ruido y mejorar las imágenes de rayos X producidas en equipos de inspección de doble energía, lo cual optimiza la detección de objetos [11]. Un enfoque similar presenta He y otros [12] mediante la Transformada Discreta de Wavelet. Por otra parte Singh propone un procedimiento para la eliminación de ruido producido por la incidencia de los rayos de menor energía [13]. A su vez Heitz [14] expone otro método para la separación de objetos en un conjunto de imágenes de rayos X utilizando la propiedad de atenuación que presentan los píxeles cuando los objetos que representan se encuentran superpuestos utilizando para ello un procedimiento de calibración que establece relaciones entre la densidad, el ancho de los materiales y el color que se muestra en la imagen.

Por otra parte, una de las principales limitantes expuestas por los especialistas de la AGR en el proceso de análisis de equipaje radica en que no se conoce la estructura interna del formato de imagen IMG generado por los equipos de inspección del fabricante NUCTECH. Esta problemática imposibilita visualizar los ficheros IMG en otro lugar que no sean los equipos de rayos X. Debido a esto el entrenamiento del personal

debe realizarse en los equipos de inspección, los cuales no siempre se encuentran disponibles para el adiestramiento. Esto repercute negativamente en la vida útil de los equipos ya que cuando no se encuentran en servicio, en vez de estar apagados, se utilizan en el entrenamiento de los nuevos técnicos radiólogos. Una herramienta que facilite la lectura de las imágenes de rayos X en formato IMG posibilitaría el entrenamiento de los técnicos en cualquier computadora personal y alargaría la vida útil de los equipos de inspección.

El estudio hasta aquí realizado ha revelado las limitaciones tecnológicas y metodológicas existentes, las que influyen negativamente en el proceso de análisis del equipaje, desde una perspectiva que permita abordar el problema del reconocimiento y recuperación de objetos peligrosos en imágenes de rayos X, facilitando la clasificación del contenido de las muestras. Por lo que el **campo de acción** se delimita en las herramientas de software para la caracterización e identificación de objetos ilícitos en imágenes de rayos X.

De aquí que resulta necesario profundizar en esta cuestión con el propósito de hallar y adaptar alguna vía de solución a la situación encontrada. Recientemente, el algoritmo denominado Bolsa de Palabras Visuales, *Bags of Visual Words*, BoVW por sus siglas en inglés, se ha utilizado con éxito en la clasificación, detección de objetos, reconocimiento y recuperación de imágenes basada en contenido [15] [16] [17] [18]. BoVW fue propuesto en sus inicios por Csurka [19] para la búsqueda de imágenes por contenido. Más tarde [15] y [16] extienden la utilización de esta técnica a las imágenes de rayos X en sistemas de transmisión de energía simple para reconocer materiales ilícitos. Otra contribución en esta área del conocimiento la realiza Castro [20] con la utilización de imágenes de equipos de energía dual y el estudio de la influencia de diferentes *kernels*, funciones de pérdida y vocabularios en la clasificación de armas cortas.

De forma general existe una bibliografía limitada sobre el tema y las investigaciones que utilizan este algoritmo difieren entre ellas, tanto en los experimentos realizados como en los tipos de imágenes empleadas. Además no se presentan datos relevantes acerca de los tiempos de ejecución de los algoritmos siendo este punto de vital importancia en la implantación de una solución informática, ya que no resulta viable técnicamente un algoritmo con una alta tasa de reconocimiento si este se ejecuta lentamente. De igual

forma, según los directivos de la AGR entrevistados, no se cuenta con una herramienta de software que permita el reconocimiento de armas de fuego en tiempo real en el proceso de inspección. Es por ello que se hace necesario profundizar en el estudio de cada una de las etapas del algoritmo y en su implementación basada en los requerimientos específicos del tipo de equipo de inspección de rayos X que se utiliza en nuestro país, con el objetivo de concretar una herramienta de *software* que posibilite la detección de armas cortas.

Todo lo anterior permite proponer como **objetivo general** la implementación de un componente de *software* que facilite la identificación y reconocimiento de armas cortas en imágenes de rayos X mediante el algoritmo BoVW, que sirva de apoyo a los operadores de los equipos de inspección en el análisis del equipaje.

Por lo que se define como **hipótesis**, si se aplica el algoritmo BoVW mediante la implementación de un componente de *software*, sustentado en las peculiaridades de las imágenes de rayos X y tomando en cuenta las características de los equipos de inspección, se favorece la identificación de armas cortas en el proceso de inspección del equipaje de forma rápida y confiable.

Una vez definido el camino hipotético que dará solución al problema científico de la presente investigación se definen las siguientes **tareas de la investigación**:

- Definir la terminología y métodos empleados en el proceso de inspección por rayos X.
- Caracterizar el formato de imagen IMG de los equipos de rayos X.
- Determinar un procedimiento para la identificación de armas cortas en imágenes de rayos X.
- Implementar el componente de software basado en el método BoVW.
- Realizar un estudio de rendimiento de la implementación realizada con vista a favorecer su ejecución de forma rápida.
- Corroborar la evaluación del algoritmo propuesto en un conjunto de imágenes de rayos X donde estén presentes varios tipos de objetos.

Introducción

Para llevar a cabo la investigación se hace uso de los **métodos científicos** que se resumen a continuación:

- *Histórico – lógico*, para el estudio de la evolución del procesamiento y análisis de imágenes y de los diferentes métodos de extracción de descriptores.
- *Análisis documental*, permitió profundizar en las particularidades, alcance, y actualidad del proceso de visión computacional, así como de las tecnologías y la problemática abordada.
- *Análisis y síntesis*, como método de descomposición del fenómeno objeto de estudio, permitió conocer las particularidades de las partes y la integración de los momentos esenciales de cada una de ellas en un todo orgánico, para la concepción del marco teórico.
- *Modelación*, facilitó la concepción del componente informático para la extracción de características, identificación y reconocimiento.
- *Observación*, para la selección de las combinaciones de parámetros y descriptores en el algoritmo propuesto.
- *Medición y experimentación*, con el objetivo de evaluar el método propuesto.
- *Entrevista y encuesta*, como herramientas para dominar el proceso de inspección a través de equipos de rayos X, y facilitar la extracción de los requerimientos.

El **aporte práctico** de la presente investigación se sustenta en la implementación de un componente de *software* que concreta un procedimiento basado en el algoritmo BoVW y facilita la identificación y el reconocimiento de armas cortas en imágenes de rayos X de equipos de inspección de energía dual utilizados en nuestro país.

El **impacto social** de la investigación radica en la posibilidad de favorecer el logro de resultados significativos en el proceso de inspección de equipajes mediante equipos de rayos X, a partir de:

- Favorecer el entrenamiento de los especialistas.
- Disminuir el uso de los equipos en el proceso de entrenamiento.
- Potenciar la rapidez y eficacia de la identificación y el reconocimiento de armas cortas.

- Reducir el esfuerzo y la carga del operador en el proceso de inspección.

El presente documento se encuentra estructurado en una introducción, tres capítulos, conclusiones, recomendaciones, bibliografías y anexos. En el capítulo 1 se expone el marco teórico contextual de la investigación, referente al análisis de las imágenes de rayos X, la identificación automática de objetos peligrosos y los distintos equipos de inspección; así como las herramientas y metodología de software a emplear en el desarrollo del sistema. El capítulo 2 expone los detalles de las fases de análisis y diseño de la metodología seleccionada, se muestran las funcionalidades del sistema, los requisitos no funcionales y el diseño de clases de la herramienta. El capítulo 3 recoge los estudios experimentales realizados para mostrar la efectividad del sistema implementado.

CAPÍTULO 1: MARCO TEÓRICO CONTEXTUAL

1.1 Introducción

En la actualidad, en contraste con la alta especialización de los técnicos radiólogos y la utilización de equipos de rayos X de última generación en la inspección de equipajes en puertos y aeropuertos, subsisten dificultades, que en ocasiones impiden el normal desarrollo de esta actividad. Cada vez más es deseable la implementación de herramientas que reduzcan la carga de trabajo de los especialistas, mejoren la eficiencia en la clasificación de objetos y aumente la velocidad del procedimiento, siempre teniendo como premisa que no se desea eliminar del todo la labor consiente del hombre sino humanizar, fortalecer y complementar su tarea [6].

En el proceso de requisa aduanal en los puertos y aeropuertos de nuestro país, intervienen varios grupos de trabajo, entre ellos se encuentra el Departamento de Análisis y Enfrentamiento (DAE) que se encarga del estudio de la información adelantada sobre pasajeros, cargas y aeronaves, realizando las caracterizaciones de los vuelos y estableciendo determinados perfiles de riesgos. De acuerdo a ello se organiza y dirige la estrategia de enfrentamiento de la unidad. También existen las áreas de operaciones especiales, integradas por un personal con una preparación especializada para el enfrentamiento directo de algunas situaciones; tales como la brigada canina y el área de inspección con empleo de sistemas de rayos X. Esta última la conforman los radiólogos en conjunto con los sistemas de inspección de equipajes por rayos X, mediante los cuales se pueden identificar sustancias y objetos peligrosos en las valijas de los pasajeros o los contenedores de carga.

A pesar del enfrentamiento y el esfuerzo sistemático que se desarrolla por estos grupos de trabajo; según los expertos de la AGR, el proceso de detección de objetos ilícitos por los sistemas de inspección, está siendo afectado por una elevada estadística de falsos positivos (alarmas emitidas cuando no hay elementos peligrosos), en particular durante el empleo de los equipos del fabricante NUCTECH¹, los cuales son destinados a la

¹ La compañía NUCTECH es una empresa China creada en 1997, especializada en la creación y comercialización de sistemas de seguridad para la aviación civil, las aduanas, los metros urbanos, ferrocarriles, entre otros; la cual se ha convertido en uno de los mayores proveedores de equipos de inspección por rayos X a nivel mundial [53]. Los equipos de este fabricante son los más utilizados en Cuba.

inspección de bultos y equipajes de mano, ocasionando molestias entre los pasajeros y aumentando la demora en el proceso de inspección, de ahí la necesidad de identificar y evaluar algoritmos de visión computacional que faciliten la inspección automática de las imágenes de rayos X.

1.2 Equipos de inspección por rayos X

Los equipos de inspección por rayos X se basan en el principio de diferenciar los objetos de una muestra o valija a partir de la absorción de rayos X que presenten los materiales que lo componen [21]. Esencialmente estos equipos se componen de un emisor de los rayos X, un conjunto de sensores para cuantificar la energía del rayo al atravesar cada una de las regiones del objeto y una banda transportadora para desplazar el objeto a inspeccionar. El resto del equipo se compone de dispositivos de control, de protección y de interacción con el operador.

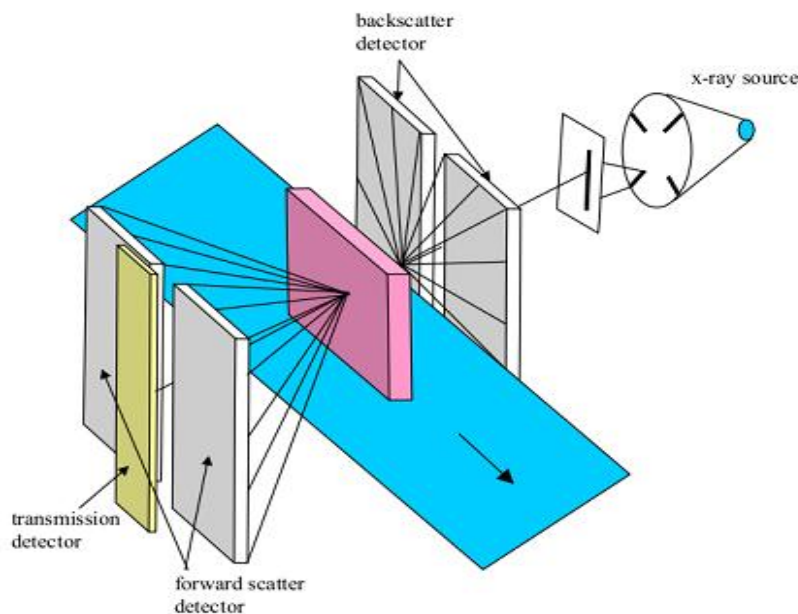


Figura 1.1: Esquema de equipo de inspección por rayos X

En la figura 1.1 se muestra el esquema de un equipo de inspección por rayos X. El modo de funcionamiento, de estos equipos para formar la imagen del objeto bajo análisis, de manera simplificada es el siguiente. En un instante de tiempo dado, los sensores, que usualmente se encuentra en distribución en fila o en forma de L, alineados con el emisor en un plano perpendicular a la dirección de movimiento de la banda transportadora,

registran el valor de la intensidad de rayo que llega a dicho sensor. Este conjunto de rayos registra una columna de la imagen (cada sensor corresponde a un píxel de la imagen). En la medida que la banda transportadora se desplaza, se van generando las distintas columnas de una imagen.

Para identificar los diferentes objetos presentes en la muestra es necesario relacionar la absorción registrada en cada píxel con las propiedades del material de que este se compone. Entre las propiedades de un material que inciden en el nivel de absorción de los rayos X, en un material puro, están el número atómico y la densidad [14].

La ecuación 1 expresa la relación entre la intensidad de rayo emitido, el número atómico, la densidad de la sustancia y la intensidad del rayo luego de atravesar el material.

$$I(E) = I_0 e^{-\alpha(Z,E)\rho t} \quad (1.1)$$

Donde ρ es la densidad del material, I_0 es la intensidad del rayo emitido, t es el ancho de material por donde pasa el rayo y E la energía de los fotones de rayo X.

Dado que la mayoría de los materiales en la naturaleza no son puros, se define el número atómico efectivo Z_{eff} como el valor promedio de los números atómicos de las diferentes sustancias que lo componen, ponderado por las proporciones de cada sustancia en el material [3]. Para representar esta propiedad en una imagen digital es asociado un color diferente a rangos determinados de Z_{eff} de manera que pueda visualizarse como una imagen en colores. Este tipo de imágenes son conocidas como imagen en pseudo-color o falso color.

1.2.1 Sistemas de transmisión de energía simple

Los sistemas de rayos X convencionales de energía simple son efectivos para detectar objetos de composición metálica (cuchillos, armas, etc.), mientras que las armas de baja densidad podrían pasar desapercibidas.

Dos ejemplos de imágenes de un solo nivel de energía generadas por estos sistemas de detección se muestran en la figura 1.2. La imagen de la izquierda muestra objetos peligrosos de alta densidad; y la derecha muestra objetos peligrosos de baja densidad.

Los píxeles de más alta intensidad en la imagen corresponden a objetos de alta densidad, mientras que los píxeles de más baja intensidad corresponden a objetos de baja densidad en los equipajes [22].

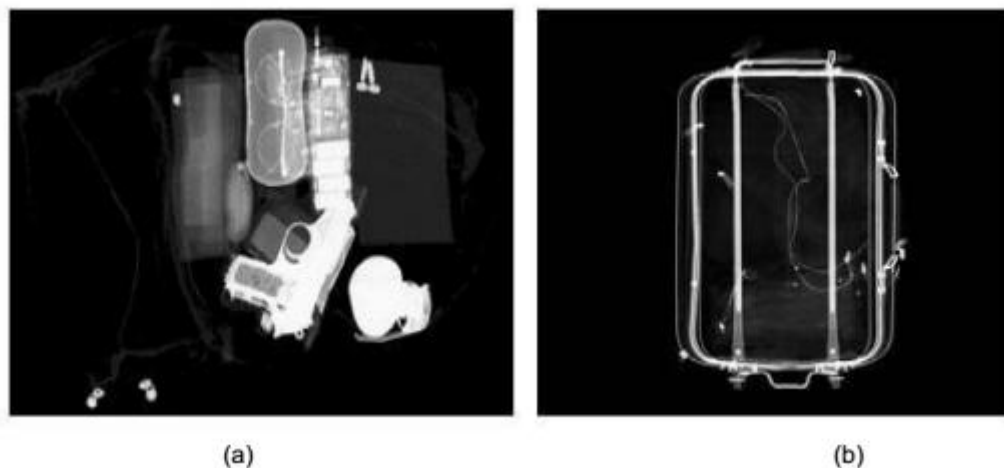


Figura 1.2: Imágenes de rayos X de energía simple. (a) Objetos de alta densidad. (b) Objetos de baja densidad

Como se muestra anteriormente en las imágenes de rayos X generadas por equipos de un solo nivel de energía, especialmente en la imagen de la derecha, diferentes objetos de la muestra son difíciles de distinguir por observación directa desde el sistema de detección.

1.2.2 Sistemas de transmisión de doble energía

Un sistema de imágenes de rayos X basado en el empleo de energía dual, usa dos niveles de energía, uno mayor y otro de menor energía [14]. A niveles de energía mayores ($> 100kV$), la energía absorbida depende de la densidad del material. A mayor densidad, más energía absorbida por el objeto y en consecuencia, más oscura aparece la imagen. A menores niveles de energía ($< 80kV$), la absorción depende del número atómico efectivo y del espesor del material. Mientras las áreas de metal pesado son oscuras en ambas imágenes de nivel bajo y nivel alto las áreas de los elementos más ligeros son más oscuras en la imagen de bajo nivel. El equipo combina las dos imágenes generadas y asocia tonalidades de color naranja, verde y azul en función de la absorción de los materiales. Gracias a esto elementos ligeros como el carbón, nitrógeno y oxígeno pueden ser detectados.

Los equipos de inspección por rayos X de energía dual tienen la capacidad de entregar varios tipos de imágenes en pseudo-color, en función del tipo de objeto que se desee enfatizar. El modelo de color denominado *colorize*, en los equipos del fabricante NUCTECH, es uno de los más utilizados por los radiólogos en el proceso de inspección de equipajes en los puertos y aeropuertos cubanos. En el modelo *colorize*, los materiales orgánicos se muestran con colores de tono naranja, los materiales metálicos con colores de tono azul y aquellos que no clasifican en ninguno de los dos grupos anteriores se representan en colores de tono verde. En la figura 1.3 se muestra una imagen tomada con un equipo NUCTECH usando el modelo de color *colorize*.

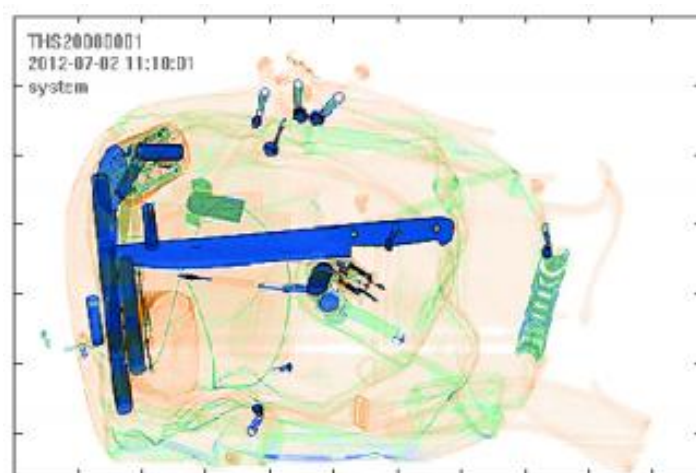


Figura 1.3: Imagen en pseudo-color obtenida por un equipo de inspección de rayos X de energía dual empleando el modelo de color “colorize”

En la actualidad existen otros equipos de inspección por rayos X que son capaces de obtener múltiples vistas de la muestra desde diferentes ángulos, posibilitando al operador reconocer con mayor facilidad los objetos. Mediante esta técnica se reconstruyen los objetos en tercera dimensión a través de una tomografía computarizada [9] [23] lo cual requiere de una enorme sobre carga computacional, y dificulta su utilización práctica. Aunque la aplicación de imágenes en 3D ha sido ampliamente utilizada en la industria médica, sólo recientemente se ha aplicado en los equipos de rayos X.

1.3 Representación de la materia en las imágenes de rayos X

Las imágenes de rayos X obtenidas en equipos de energía dual constituyen una combinación de dos radiográficos adquiridos a diferentes niveles de energía con los que es posible obtener la densidad y el número atómico de los materiales. Las imágenes de baja y alta energía son fusionadas con la ayuda de una tabla de colores para lograr una imagen de color que facilita la interpretación del contenido de las muestras [6].

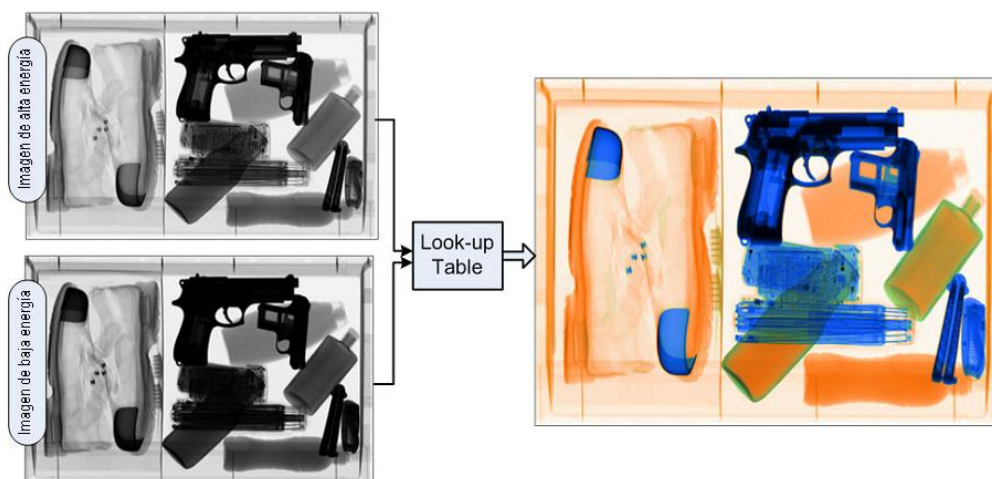


Figura 1.4: Obtención de una imagen de rayos X en un equipo de energía dual

Este tipo de imágenes poseen características visuales que pueden ser aprovechadas, por ejemplo: el color que se observa se corresponde con el número atómico del material. La materia que compone a los objetos se representa en función del peso atómico efectivo (Z_{eff}) de los elementos químicos, este es transformado en diferentes tonalidades de color naranja, verde y azul en una imagen digital, esta asignación se produce en tiempo real según la absorción de rayos X de los materiales de la muestra.

La combinación de diferentes átomos con su peso atómico forma moléculas dando una mezcla de materiales. Además, la superposición en los equipajes origina los distintos pesos atómicos efectivos los cuales son proporcionales a la densidad física de los materiales. El color indica el tipo de material, la luminosidad, la intensidad y el espesor del material. Cuando existe superposición de materiales predomina el color del material de mayor absorción de rayos X [24].

Con el modelo de color *colorize* de los equipos de energía dual se identifican las propiedades de la materia de los artículos en inspección, representándolos según su masa atómica en diferentes tonalidades de color:

- Los elementos con $Z_{eff} \leq 11$ se muestran de color naranja, corresponden a materiales orgánicos; ej. alimentos, ropas, explosivos y drogas.
- Los elementos con $Z_{eff} \geq 18$ se muestran de color azul, corresponden a los materiales inorgánicos pesados; ej. metales: hierro, cobre, oro, plata, etc.
- Los elementos $Z_{eff} < 11 > 18$, se representan de color verde, son los elementos semipesados intermedios, inorgánicos como el cloro, sodio, sales, aluminio, entre otros.

Tabla 1.1: Representación de los diferentes grupos de elementos según su masa atómica efectiva

	Color	Z_{eff}	Tipo de elementos
Grupo 1	Naranja	Menor de 11	Elementos más ligeros: hidrógeno, carbono, nitrógeno, oxígeno y sus compuestos moleculares. Materiales orgánicos: Como muchos explosivos, plásticos, papel, textiles, producto a alimenticios, madera y agua.
Grupo 2	Verde	Entre 11 y 18	Elementos semipesados: aluminio sodio, cloro, y cloruro de sodio.
Grupo 3	Azul	Más de 18	Elementos pesados, metales: titanio, cromo, hierro, níquel, cobre, zinc, plomo oro y plata.

1.4 Algoritmo Bags of Visual Words

El método Saco de Palabras Visuales está constituido por tres fases, en su desarrollo se utilizan tanto algoritmos de clasificación supervisada como no supervisada (figura 1.5). La primera fase se basa en la construcción de un vocabulario de palabras visuales empleando como fuente una base de datos de imágenes. Para ello, se extraen rasgos visuales de todas las imágenes con algún método de extracción de características. Luego se utiliza un algoritmo de agrupamiento (*clustering*), usualmente k-means, que crea grupos de rasgos visuales similares entre sí. Los centros de cada grupo son llamados palabras visuales.

La segunda etapa tiene como objetivo el entrenamiento de un clasificador binario, el más utilizado en este caso es la Máquina de Soporte Vectorial (SVM). En esta fase de entrenamiento se lleva a cabo la representación de las imágenes como histogramas de palabras visuales. Los histogramas se construyen mediante una cuantificación vectorial de los rasgos extraídos de la imagen con el vocabulario de palabras visuales [25]. La meta es asignar cada rasgo de la imagen a la palabra visual más cercana. Con los histogramas de todas las imágenes se entrena al clasificador binario SVM, el cual define el hiperplano óptimo que maximiza el margen entre las muestras de cada clase respecto a la frontera de separación de los datos. Para poder tratar con datos que no son linealmente separables se introduce la función *kernel* y la función de pérdida. El *kernel* permite expandir los datos a un espacio de mayor dimensión buscando una mejor separación lineal entre las clases, con la función de pérdida se puede observar el máximo error de la clasificación.

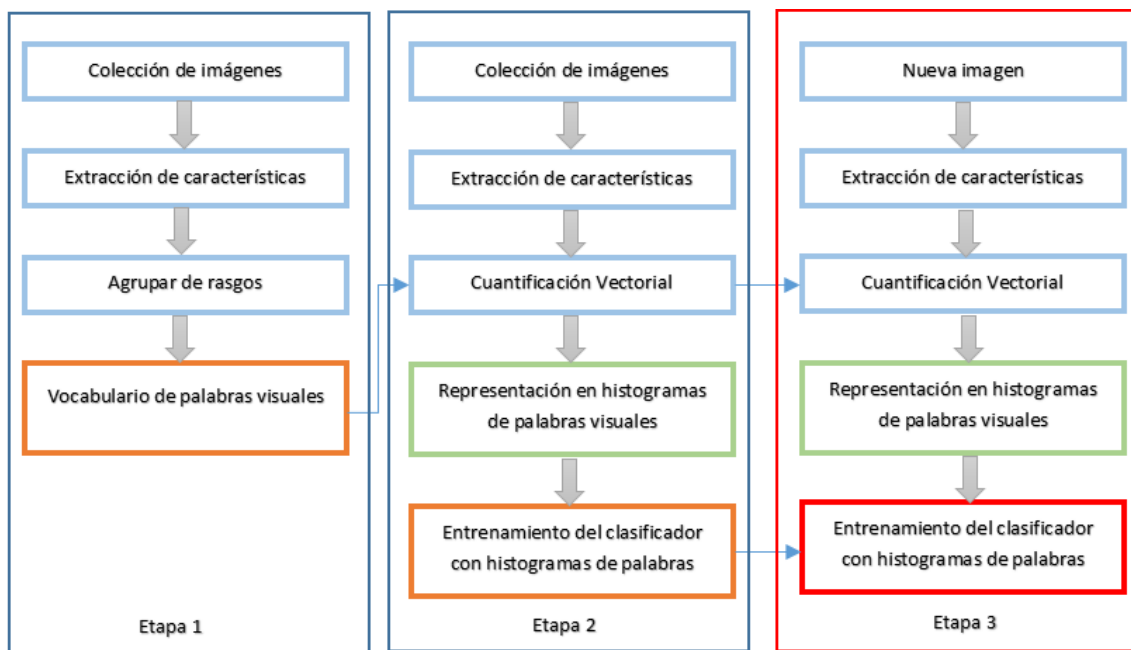


Figura 1.5: Esquema general del método BoVW

La tercera fase utiliza los datos de las dos etapas anteriores, el vocabulario de palabras visuales y el entrenamiento del clasificador basado en los histogramas del universo de imágenes. Es el proceso mediante el cual se logra la clasificación de una nueva imagen, no contenida en el proceso de entrenamiento. Para ello la nueva imagen es transformada en un histograma de palabras visuales, y comparada en el clasificador SVM quien determina si el histograma pertenece o no a la clase objetivo.

Las dos primeras etapas se caracterizan por una enorme sobrecarga computacional y se realizan de forma *offline*, o sea en el proceso de entrenamiento del algoritmo. La meta aquí consiste en ajustar los parámetros de configuración del método con el propósito de lograr a un porcentaje elevado de reconocimiento de los objetos que se desean identificar en las imágenes. La tercera etapa se realiza de forma *online* y se identifica por ser mucho más rápida que las fases anteriores. Esta última fase es la que se debe potenciar mediante una implementación robusta, extensible y con tiempos de respuesta cortos que permita su utilización por parte de los especialistas como ayuda en el proceso de inspección, ya que no resulta viable técnicamente un algoritmo con una alta tasa de reconocimiento si este se ejecuta lentamente.

1.4.1 Extracción de características

El término extracción de características hace referencia a la tarea de localizar en una imagen puntos o regiones tanto relevantes en cuanto a la cantidad de información de su entorno -por ejemplo, por la existencia de bordes o texturas- como estables frente a las perturbaciones y transformaciones locales y globales que puede sufrir la imagen. Esto último garantiza que dichos puntos se puedan detectar de forma confiable y con cierta exactitud al realizar esta tarea de forma repetitiva en diferentes entornos.

El proceso de extracción de características es la primera tarea a efectuar para el desarrollo del método BoVW. Este consiste en hallar los descriptores de los puntos clave que representen detalles distintivos en las imágenes. Se utiliza para ello un algoritmo de extracción de características en imágenes, entre los más conocidos se encuentran SIFT [26], HOG [27], SURF [28], detector de Harris [29], MSER [30], entre otros. De manera general los descriptores extraídos por estos métodos deben ser invariantes en alguna medida a las variaciones irrelevantes en la tarea de clasificación (transformaciones de la imagen, variaciones de iluminación y oclusión) pero deben, de la misma forma, contener suficiente información discriminativa que posibilite su clasificación al nivel de clase [5].

1.4.1.1 Scale Invariant Feature Transform

El algoritmo Transformada Característica Invariante a Escala, SIFT por sus siglas en inglés, fue presentado por David Lowe en 2004 [26]. Este método permite la extracción

de puntos de interés o puntos clave distintivos de una imagen que pueden ser utilizados en el reconocimiento de objetos en imágenes. Su éxito frente a otros métodos empleados radica en que estos puntos de interés son invariantes a escala, rotación, cambios de puntos de vista, adición de ruido, y es parcialmente invariante a cambios de iluminación [26]. Estas peculiaridades hacen a este método lo suficientemente robusto para la detección y el reconocimiento de objetos en imágenes y video.

Para detectar los puntos SIFT lo primero que se hace es representar la imagen en diferentes escalas y tamaños mediante filtrados sucesivos con la función gaussiana de diferentes escalas (sigmas). Posteriormente se ejecuta la diferencia de gaussianos (*Difference of Gaussian*, DoG); que se obtiene con la resta de dos imágenes filtradas consecutivas para comenzar a identificar los puntos de interés.

La figura 1.6 muestra los puntos clave SIFT detectados en una imagen. Estos son representados a través de una región circular con una orientación en el punto de localización.

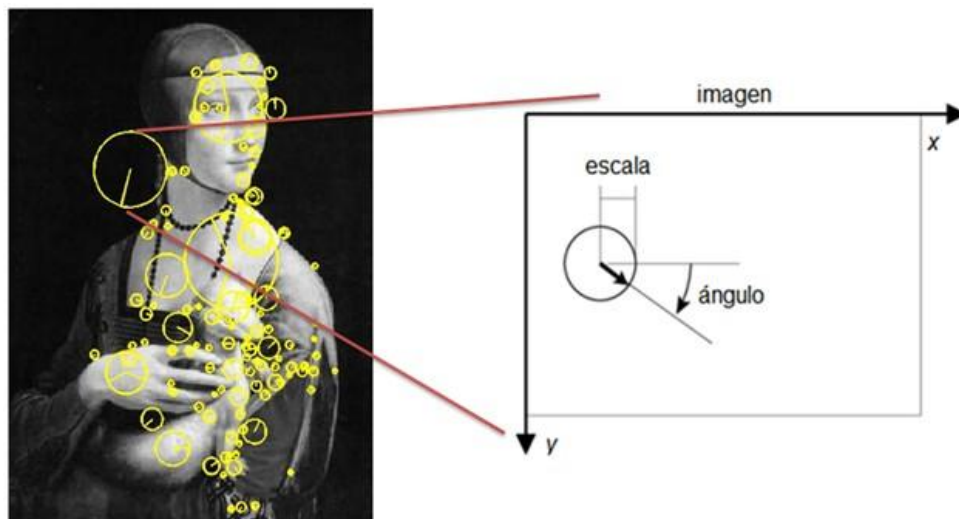


Figura 1.6: Ejemplo de punto clave SIFT.

Los puntos clave tienen las siguientes características:

- La posición del punto clave es determinada por las coordenadas (x, y) del centro.

- La escala es el parámetro de la función gaussiana con la cual se realiza el filtrado de la imagen, permitiendo que el punto clave sea invariante a cambios de tamaños en la imagen.
- La orientación expresa la dirección del punto clave en radianes, posibilitando que el punto identificado sea invariante a rotación.

Los puntos con una región circular mayor corresponden a puntos detectados en escalas mayores, donde la imagen es filtrada con una mayor función gaussiana y viceversa para puntos con una menor región circular. En la figura 1.6 se pueden observar estas características.

El algoritmo SIFT construye sobre cada punto clave un descriptor que no es más que un histograma espacial en 3D de los gradientes de la imagen. Sobre el punto clave se proyecta una ventana gaussiana circular (centro de la figura 1.7), dentro se puede observar que hay 16 regiones (rectángulos en magenta grueso). Se obtiene entonces un histograma de orientación para cada una de estas regiones. Los 360 grados de orientación se cuantifican en 8 niveles, donde cada nivel corresponde a un cambio en la orientación de 45 grados.

La longitud de cada flecha corresponde a la suma de las magnitudes de los gradientes que se encuentran próximos a esa dirección dentro de la región. Las coordenadas espaciales son cuantificadas en cuatro niveles, donde cada nivel corresponde a una región [26]. Finalmente el descriptor está constituido por un arreglo de 4 x 4 de histogramas con 8 orientaciones en cada cuadrado. El descriptor SIFT está conformado por $4 \times 4 \times 8 = 128$ valores característicos para cada punto clave.

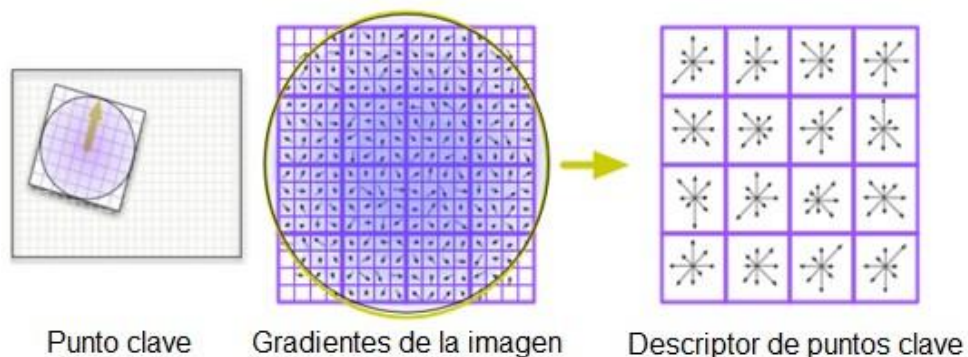


Figura 1.7: Descriptor SIFT

Entre los parámetros del algoritmo se encuentra el *factor de magnificación* que determina el tamaño del descriptor multiplicándolo por la escala en que se encuentra el punto clave. Es decir el tamaño del descriptor depende directamente del factor de magnificación y de la escala en que se encuentra el punto clave.

El descriptor SIFT es uno de los descriptores más utilizados en la extracción de características y ha tenido muy buenos resultados en aplicaciones de visión por computadora. Se ha demostrado que tiene mejor desempeño frente a otros algoritmos en diferentes contextos [29].

1.4.1.2 Dense SIFT

DSIFT es una versión rápida de SIFT para un conjunto denso de puntos de interés, cuyos parámetros de configuración son definidos a priori. Este algoritmo es aproximadamente equivalente a ejecutar SIFT en una cuadrícula de puntos densa a una escala y orientación fija. La principal ventaja de la versión densa de SIFT es que es mucho más rápida que el SIFT original [31]. DSIFT tiene algunas nuevas características que explican el porqué es más rápido que el SIFT estándar:

1. La localización de cada punto clave es predefinida y no depende de las características del gradiente del pixel.
2. La escala de cada punto clave es la misma y se define mediante un parámetro en la configuración del algoritmo.
3. La orientación de cada punto clave es siempre cero.

Con estas nuevas características DSIFT puede extraer mayor cantidad de puntos clave en menos tiempo de lo que lo hace SIFT. DSIFT especifica el tamaño del descriptor mediante el parámetro tamaño de ranura o *binSize*, el cual controla la dimensión de la ranura espacial de SIFT en pixeles. En el descriptor del SIFT estándar, el tamaño de la ranura está relacionado con la escala del punto clave y por el factor de magnificación, el cual por defecto está en 3.

1.4.1.3 Pyramid Histogram Of Visual Words

Para la extracción de características el algoritmo a utilizar debe priorizar una cantidad de elementos que contengan información relevante frente al costo computacional,

debido a las condiciones de seguridad de la aplicación. Una propuesta que maximiza la cantidad de descriptores y ofrece buenos tiempos de ejecución la constituye el algoritmo PHOW (*Pyramid Histogram Of visual Words*) propuesto por Bosch en 2007 [32], el cual resulta conveniente para su aplicación en las imágenes de rayos X debido a que estas poseen menor textura y resultan más opacas que las imágenes del espectro visible. PHOW es una variante de DSIFT que extrae los descriptores a múltiples escalas fijas en los diferentes canales de una imagen en colores, favorece extraer mayor cantidad de puntos característicos lo que repercute en una mayor precisión en la clasificación [17] [33].

El algoritmo PHOW consiste en calcular los descriptores SIFT en una cuadrícula con espacio de M píxeles. En cada uno de estos puntos los descriptores son calculados sobre cuatro regiones circulares fijas. De manera que cada punto es representado por cuatro descriptores SIFT dado que se calculan sobre cuatro escalas fijas, de igual forma la orientación de los descriptores es fijada a un valor constante, siempre en 0. La figura 1.8 (c) muestra los descriptores PHOW cuando los valores de escala son fijados en 4, 6, 8, 10 píxeles. También se observan las diferencias entre los puntos SIFT y PHOW.

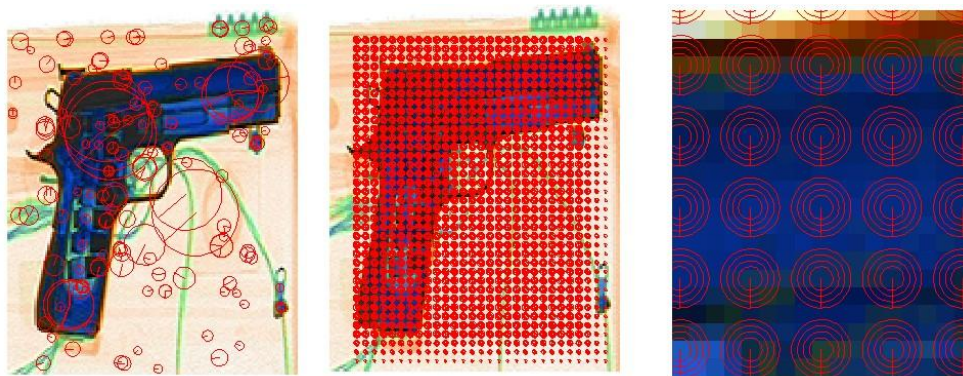


Figura 1.8: De izquierda a derecha: a) Descriptores SIFT, b) Descriptores PHOW, c) zoom sobre los descriptores PHOW.

1.4.2 Agrupación de características

Luego de la extracción de características es necesario utilizar un algoritmo de agrupamiento para organizar y asociar los datos según sus similitudes. *K-means* es

uno de los algoritmos descrito en la literatura más utilizados para lograr esto. Su propósito es dividir un conjunto de vectores en k grupos distintos alrededor de un vector media común. Es decir se trata de encontrar k centros distintos, también como centroides los cuales deben representar un patrón compartido por los puntos claves en ese grupo. En el método BoVW estos grupos representan las llamadas *palabras visuales* y una colección de estas palabras se conoce como *vocabulario visual*. El número de grupos o palabras visuales determinan el tamaño del vocabulario visual. Al conjunto de los centroides se le conoce como *codebook* [34]. La figura 1.9 representa un ejemplo de una agrupación mediante el algoritmo *k-means*, donde los puntos negros son los centroides y los grupos de color son los vectores de características.

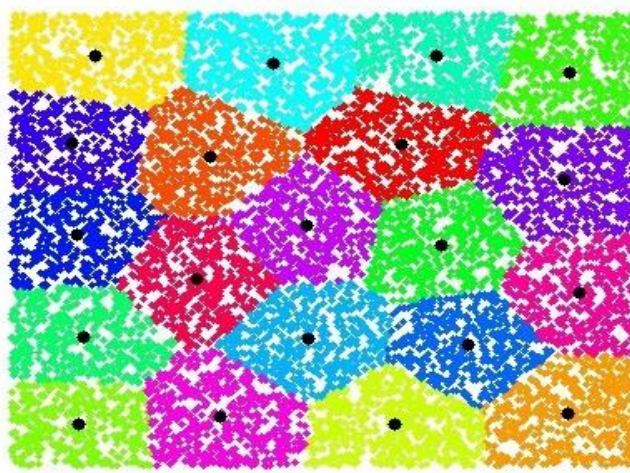


Figura 1.9: Ejemplo de agrupación mediante el algoritmo *k-means*

Dado un conjunto de observaciones (x_1, x_2, \dots, x_n) , donde cada observación es un vector real de d dimensiones, *k-means* construye una partición de las observaciones en k conjuntos $S = S_1, S_2, \dots, S_k$ con el fin de minimizar la suma del cuadrado de las distancias de los puntos al centro de su grupo.

El algoritmo *k-means* busca k vectores μ_1, \dots, μ_k de manera que sea minimizado el error acumulativo. Donde μ_i son los centroides y representan la media de los puntos en S_i . El algoritmo es implementado mediante un proceso iterativo donde se pretende alcanzar el mínimo, pero no siempre se obtiene el mismo resultado.

Según se muestra en la figura 1.10 cada centroide \mathbf{w} representa a todos los descriptores en cada palabra visual. Este se obtiene a través del promedio de la ubicación de todos los elementos de su grupo.

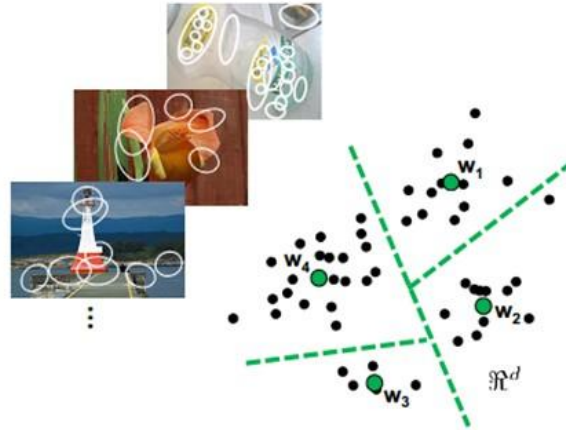


Figura 1.10: Representación de las palabras visuales (\mathbf{w}_1 , \mathbf{w}_2 , \mathbf{w}_3 y \mathbf{w}_4), mediante el agrupamiento de los descriptores de características con k -means.

1.4.3 Cuantificación vectorial

Luego de la extracción de características de las imágenes en la base de datos, utilizada en la construcción del vocabulario y el agrupamiento de los descriptores según su similitud; se desea representar una nueva imagen de entrada mediante la frecuencia de repetición de sus características visuales. Usando la cuantificación vectorial, cada nuevo descriptor identificado en la nueva imagen, es asignado al centroide más cercano, es decir cada descriptor se asocia a la palabra visual más cercana del vocabulario como muestra la figura 1.11. Posteriormente se procede a construir un histograma de palabras visuales; el cual es una representación gráfica de la cantidad de descriptores que fueron asociados a cada palabra visual [35]. Este histograma es el vector de rasgos o vector característico que se utiliza en el entrenamiento del algoritmo clasificador. En la figura 1.11 se puede observar que la palabra visual \mathbf{w}_2 es a la que mayor cantidad de descriptores se le asoció, un total de 6 descriptores, seguidamente de \mathbf{w}_4 con 3 asignaciones. Mientras que a \mathbf{w}_1 y \mathbf{w}_3 cuentan con solo un descriptor.

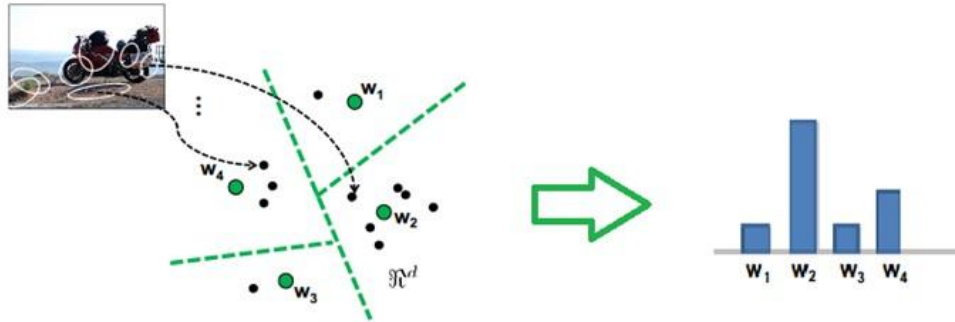


Figura 1.11: Cuantificación vectorial para la obtención del histograma de palabras visuales.

1.4.4 Etapa de clasificación

Los problemas de clasificación en visión por computadora centran su interés en saber si una determinada muestra, en función de las características que presenta, pertenece a una u otra clase. En el ámbito del aprendizaje de máquina consiste en encontrar una regla de decisión tal que, dada una muestra externa, ésta sea asignada a su clase correspondiente.

La etapa de clasificación para la predicción de las clases a la que pertenecen las muestras utilizando algoritmos de clasificación supervisada, se basa fundamentalmente en dos etapas, denominadas entrenamiento y prueba. Durante el entrenamiento son enviadas al algoritmo clasificador las muestras etiquetadas, es decir que se conoce a que clase pertenece, con el propósito de que el algoritmo clasificador “aprenda” a reconocer los diferentes tipos. Posteriormente son utilizadas las muestras de prueba, para evaluar cuan preciso es el clasificador en el proceso de tipificación de los nuevos datos de entrada [36].

En el estado del arte se han desarrollado diversos métodos de aprendizaje basándose en la representación de saco de palabras visuales. Estos métodos pueden ser divididos en dos tipos de modelos: los generativos y los discriminativos; los cuales, en su mayoría se utilizan en el ámbito de reconocimiento de textos, imágenes y video. Algunos de los modelos generativos más empleados son el clasificador Naïve Bayes y los modelos bayesianos jerárquicos. Entre los modelos discriminativos se encuentran las Máquinas de Vectores Soporte (SVM).

1.4.4.1 Fundamentos de las Máquinas de Vectores Soporte

La Máquina de Vectores Soporte o SVM (*Support Vector Machine*) fue desarrollada en 1963 por Vapnik y Lerner en [37] en los laboratorios AT&T, pero no fue hasta los años noventa, con la publicación del primer artículo científico por Boser en 1992 y por Cortes y Vapnik en 1995, cuando fue desarrollada y generalizada. Fue ideada originalmente para la resolución de problemas de clasificación binarios en los que las clases eran linealmente separable. Es por esto por lo que también se le conocía como Hiperplano² Óptimo de Separación, *OSH (Optimal Separating Hyperplane)*, ya que la solución obtenida era aquella en la que se clasificaban de manera correcta todas las muestras, colocando el hiperplano de separación lo más lejos posible de todas ellas [38]. Para lograr esto, se realiza un entrenamiento con los datos dispuestos para ese fin, a partir del cual se define el hiperplano óptimo, es decir, el que maximiza el margen entre las muestras de cada clase respecto a la frontera de separación.

El objetivo es que, tras este entrenamiento, la máquina generalice bien para datos nuevos que no han participado en el entrenamiento, es decir, clasifique correctamente a que clase pertenecen cada una de las muestras. Para ello, intenta buscar una función $f(x)$ que minimice el coste empírico sobre el conjunto de entrenamiento, manteniendo a su vez la correcta generalización de la máquina.

Para obtener una buena generalización, hay que tratar de evitar dos de los efectos típicos del entrenamiento:

- **Sobreajuste:** los parámetros obtenidos en el entrenamiento se ajustan demasiado a las muestras por lo que se pierde generalización y al introducir una nueva muestra no se clasifica bien. La figura 1.12 se puede ver un ejemplo de sobreajuste producido por línea continua.
- **Subajuste:** el hiperplano trazado es demasiado sencillo y no logra una buena generalización debido a que se han utilizado muy pocas muestras de entrenamiento, por lo que la máquina no generara buenos resultados con

² Un hiperplano es una generalización del concepto de plano. En un espacio de una única dimensión (como una recta), un hiperplano es un punto; divide una línea en dos líneas. En un espacio bidimensional (como el plano xy), un hiperplano es una recta; divide el plano en dos mitades. En un espacio tridimensional, un hiperplano es un plano corriente; divide el espacio en dos mitades. Este concepto también puede ser aplicado a espacios de cuatro dimensiones y más, donde estos objetos divisores se llaman simplemente hiperplanos, ya que la finalidad de esta nomenclatura es la de relacionar la geometría con el plano.

muestras nuevas. La línea discontinua de la figura 1.12 ilustra un ejemplo de esto.

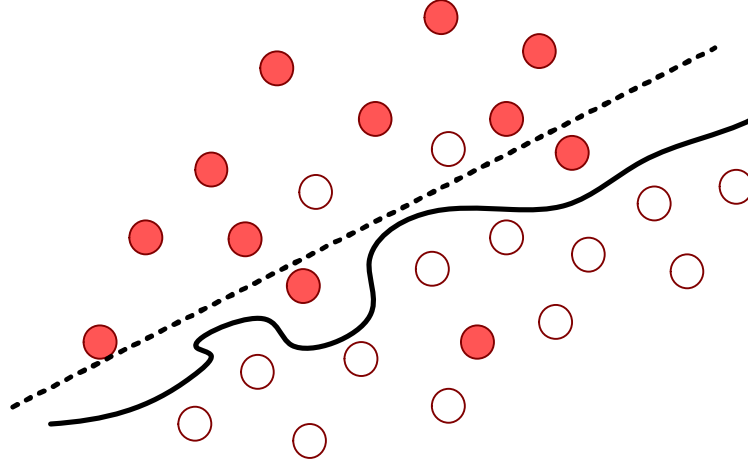


Figura 1.12: Sobreajuste y subajuste.

Al comparar una SVM con otros paradigmas de aprendizaje y procesamiento automático como las Redes Neuronales Artificiales (RNA), se puede comprobar que la SVM obtiene una mejor generalización, que hace que funcione bien cuando hay pocos datos y el espacio de entrada es de una dimensión alta [39].

Para poder tratar con datos que no son linealmente separables se introduce la función *kernel* y la función de pérdida. El *kernel* permite expandir los datos a un espacio de mayor dimensión buscando una mejor separación lineal entre las clases. Existen diferentes tipos de *kernels*, los lineales son más eficientes en el entrenamiento, pero los no lineales logran una mejor separación de clases [33], no obstante los *kernels* homogéneos aditivos agrupan ambas ventajas, [40]. La función de pérdida permite manejar muestras mal clasificadas en los datos de entrenamiento y da una medida del error cometido. Algunas de las funciones de pérdidas más utilizadas son *Linear* (*L1*), *Square* (*L2*), *Hinge*, *Square Hinge*, *Logistic*.

Existen distintas funciones *kernel* que permiten adaptar la SVM a cada conjunto de muestras, con el fin de obtener mejores resultados. Las más usadas son:

- Lineal: $k(x, x_i) = x * x_i$ (1.3)

- Polinomio: $k(x, x_i) = (\gamma x * x_i + C)^\alpha$ (1.4)

- Gaussiana: $k(x, x_i) = \exp(-\gamma |x - x_i|^2)$ (1.5)

- Base radial: $k(x, x_i) = e^{-\gamma \|x - x_i\|^2}$ (1.6)

- X^2 o *Chi-squared*: $k(x, x_i) = e^{-\gamma X^2(x, x_i)}$ donde $X^2(x, x_i) = \sum_j \frac{(x_j - x_{ij})^2}{x_j + x_{ij}}$ (1.7)

- Hellinger: $k(x, x_i) = \sqrt{x \cdot x_i}$ (1.8)

- Intersección de histogramas: $k(x, x_i) = \min\{x, x_i\}$ (1.9)

Siendo γ una constante de proporcionalidad, C un coeficiente y el rango del polinomio.

El incremento en el número de dimensiones debido a la transformación podría repercutir enormemente en el coste computacional, incrementándolo considerablemente.

1.5 Evaluación del desempeño

Con el objetivo de evaluar el desempeño de los sistemas de recuperación de imágenes dos medidas estadísticas denominadas, *Recall* y *Precision* [41], son prestadas de la tradicional recuperación de información. En un sistema de recuperación de imágenes, una vez que se ha completado el proceso de búsqueda, el conjunto de imágenes se divide en dos grupos, estos son, el conjunto de imágenes que se recuperaron y el conjunto de imágenes omitidas. Las imágenes que conforman cada grupo se subdividen a su vez, en las que son relevantes según los criterios de búsqueda y las imágenes que no cumplen estos criterios. Estos grupos se pueden observar en la siguiente tabla I:

Tabla 1.2: *Precision - Recall en la recuperación de imágenes basada en contenido*

	Relevantes	Irrelevantes	
Recuperados	a	b	(a + b) Todas las imágenes recuperadas
No recuperados	c	d	(c + d) Todas las imágenes omitidas
	(a + c) Todas las imágenes relevantes de la colección	(b + d) Todas las imágenes irrelevantes de la colección	(a + b + c + d) Todas las imágenes de la colección

Para cada imagen a consultar, las imágenes relevantes en la colección, son las imágenes que son perceptualmente similares a la consulta especificada. Otra forma de expresar estos términos es de la siguiente manera:

$$precision = \frac{A}{B} = \frac{a}{a+b} \quad (1.10)$$

$$recall = \frac{A}{c} = \frac{a}{a+c} \quad (1.11)$$

Donde:

A es el conjunto de imágenes relevantes que el sistema recuperó en una búsqueda específica.

B es el conjunto de imágenes que el sistema recuperó para una búsqueda específica.

C es el conjunto de imágenes relevantes para una búsqueda específica.

Por lo tanto, *Precision* es la división del conjunto formado por la intersección de las imágenes que el sistema recuperó en una búsqueda específica (B), con el conjunto de imágenes relevantes para la búsqueda específica (C), entre el conjunto de imágenes que el sistema recuperó para una búsqueda específica (B). De la misma forma *Recall* sería la división del conjunto formado por la intersección de las imágenes que el sistema recuperó para una búsqueda específica (B), con el conjunto de imágenes relevantes para una búsqueda específica (C), entre el conjunto de imágenes relevantes para una búsqueda específica (C).

1.6 Rendimiento de computadores paralelos

Una forma de medir la calidad de un sistema paralelizado consiste en comparar la velocidad conseguida en el sistema paralelo (con N procesadores) con la velocidad conseguida con un solo procesador [42]. Para ello se define la ganancia de velocidad o aceleración (*speed up*) de un sistema de N procesadores como:

$$S(N) = t(1)/t(N) \quad (1.12)$$

Donde $t(1)$ es el tiempo empleado para ejecutar el proceso si se utiliza un solo procesador y $t(N)$ es el tiempo empleado para ejecutarlo en el sistema paralelo con N procesadores. En condiciones ideales, $t(N) = 1/N$ con lo que, en esas mismas condiciones, la ganancia de velocidad dada por la ecuación 1.1 será

$$S(N)_{ideal} = \frac{t(1)}{t(N)} = \frac{1}{(1/N)} = N \quad (1.13)$$

Una forma de medir el rendimiento de un sistema resulta de comparar la ganancia de velocidad del mismo con la ganancia de velocidad ideal dada por 1.2, a esta medida se le denomina eficiencia:

$$E(N) = \frac{S(N)}{S(N)_{ideal}} = \frac{S(N)}{N} = \frac{\frac{t(1)}{t(N)}}{N} = \frac{t(1)}{Nt(N)} \quad (1.14)$$

La ley de Amdahl [43] pone un límite superior a la ganancia en velocidad, y por tanto también a la eficiencia, de un sistema paralelo atendiendo al hecho, de que los procesos suelen tener partes que no pueden ser ejecutadas en paralelo, sino solo de forma secuencial pura. De esta forma puede verse que:

$$S(N)_{max} = \frac{N}{1+(N-1)*f} \quad (1.15)$$

$$E(N)_{max} = \frac{1}{1+(N-1)*f} \quad (1.16)$$

Llamando f a la fracción de tiempo que no se puede paralelizar sobre el total del tiempo de ejecución.

1.7 Metodología de desarrollo de software

El esquema tradicional para abordar el desarrollo del software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante y en donde se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad [44].

Las metodologías tradicionales como *Rational Unified Process* (RUP), requieren la generación de una gran cantidad de documentación que pocas veces se mantiene y utiliza, las decisiones sobre la planificación del proyecto son tomadas al inicio y no es flexible ante posibles cambios sin afectar el costo en las fases siguientes de la metodología. Es por ello que muchos equipos de desarrollo obvian pasos en aras de simplificar un poco el proceso. Por tal razón las metodologías ágiles surgen para ser adaptadas a nuevos entornos de desarrollo.

Las metodologías ágiles promueven generalmente un proceso de gestión de proyectos que fomenta el trabajo en equipo, la organización y responsabilidad propia, un conjunto de mejores prácticas de ingeniería que permiten la entrega rápida de software de alta calidad, y un enfoque de negocio que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía. La Tabla 1.2 describe las diferencias entre las metodologías tradicionales y ágiles [45].

Tabla 1.3: *Diferencias entre las metodologías tradicionales y las ágiles*

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

La Programación Extrema (XP) es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. La metodología XP permite que el programador se centre más en el proceso de desarrollo del software, eliminando la documentación innecesaria, obteniendo buenos resultados en breves períodos de tiempo [46].

1.8 Herramientas y tecnologías

En este apartado se describen las principales características de las herramientas y tecnologías empleadas en el desarrollo del componente de software.

1.8.1 Lenguaje de programación C++

C++ es un lenguaje de programación de propósito general basado en el lenguaje de programación C. Es una extensión del lenguaje C que incluye clases, sobrecarga de operadores, funciones, conversión de tipos, referencias, tipos constantes, entre otras características novedosas.

C++ puede considerarse como un lenguaje multi-paradigma, en el cual, el programador puede escoger el estilo de programación y establecer las características del lenguaje para satisfacer el problema. El lenguaje de programación abarca tres paradigmas de programación: la programación estructurada, la genérica y la programación orientada a objetos. Soporta los conceptos básicos de la orientación a objetos como: clases, métodos, mensajes, objetos, subclases y herencia, incluyendo la herencia múltiple. El eje central de la programación en C++ es la clase y se permite la definición de nuevos tipos de datos y la combinación con los ya existentes.

C++11 es la última especificación aprobada del estándar de C++ la cual incluye varias adiciones al núcleo del lenguaje y extiende la biblioteca estándar de C++. Entre las características a destacar del nuevo estándar se encuentran, las funciones lambda, la inicialización uniforme, las plantillas con número variable de argumento, la adición de la palabra reservada `auto` para la inicialización de tipos, entre otras facilidades. C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a un alto como a un bajo nivel.

1.8.2 Biblioteca de funciones VLFeat 0.9.18

Para el desarrollo de la etapa de implementación se hizo uso de la biblioteca de funciones VLFeat [31], la cual integra implementaciones de alta calidad de algoritmos comunes de visión por computadora mediante dos interfaces, una para MATLAB y otra para C/C++, que facilitan el desarrollo en la etapa de implementación y en la realización de pruebas. Dicha biblioteca de funciones se caracteriza por su flexibilidad y portabilidad, se distribuye bajo licencia de código abierto GNU GPL e incluye mejoras

en algoritmos de extracción de características para imágenes. Como característica distintiva de VLFeat todos los nombres de funciones, variables globales y constantes comienzan con *vl_*.

1.8.3 Visual Paradigm 6.8

Visual Paradigm para UML es una de las herramientas UML comercial, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, a la vez que permite la captura de requisitos, análisis, diseño e implementación. Visual Paradigm proporciona características tales como generación del código, informes e ingeniería inversa. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y viceversa.

1.8.4 Doxygen 1.8.5

Doxygen es un generador de documentación automático, capaz de extraer la información de la documentación de los propios archivos fuentes de un programa. Se utiliza especialmente para los lenguajes de programación Java, C, C++, aunque también pudiera ser empleado en Python, C# y PHP. Dado que es fácilmente adaptable, funciona en la mayoría de sistemas Unix, así como en Windows y Mac OS X.

Doxygen es un acrónimo de *dox* (*document*) *gen* (*generator*), generador de documentación para código fuente. Varios proyectos internacionales del software libre como KDE usan Doxygen para generar la documentación de su API. KDevelop incluye soporte para Doxygen.

1.8.5 OpenCV 2.4.11

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que

sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores de múltiples núcleos. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

1.9 Conclusiones parciales

- El estudio hasta aquí realizado ha posibilitado el análisis de la situación actual del proceso de inspección aduanal de equipajes, mediante la utilización de equipos de rayos X para la identificación de sustancias y objetos peligrosos.
- Se presentaron las principales características de los equipos de inspección de la firma NUCTECH utilizados en nuestro país, clasificándose estos dos tipos: los sistemas de transmisión de energía simple y los de energía dual.
- Se realizó una representación de la materia en imágenes de rayos X, lo que permite un acercamiento a la forma en que son creadas las imágenes digitales de los equipos de inspección.
- Se llevó a cabo una descripción general del algoritmo BoVW en la que se identificaron tres etapas. Las dos primeras hacen referencia a las fases de entrenamiento del algoritmo. La tercera etapa lleva a cabo el proceso de clasificación de una nueva imagen a partir de sus descriptores utilizando el entrenamiento de un clasificador.
- Se presentó la metodología XP como una alternativa que guiará el proceso de desarrollo de software, así como las herramientas y tecnologías empleadas para darle soporte al mismo. XP permitió acortar el tiempo de desarrollo del sistema, dándole más valor a la implementación del sistema y generando solo la documentación esencial que sirva de guía para la implementación.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se dará una visión de la estructura del sistema, en cuanto a su arquitectura, el diseño de clases y otros aspectos utilizando para ello la metodología de software XP.

2.1 Análisis de las variables de configuración del algoritmo BoVW

Con el objetivo de profundizar en las características y particularidades del método BoVW, en esta sección se presenta un análisis de algunos los principales parámetros y procedimientos a tener en cuenta con vista a favorecer los resultados en la etapa de diseño e implementación del componente de software.

2.1.1 Tamaño del vocabulario de palabras visuales

El tamaño del vocabulario está determinado por la cantidad de grupos que se desean en el proceso de agrupamiento del algoritmo *k-means*. Elegir el tamaño adecuado del vocabulario representa un compromiso entre discriminación y generalización. Un vocabulario pequeño que contenga pocas palabras visuales no es muy discriminatorio porque diferentes puntos clave pueden pertenecer a una misma palabra visual. A medida que se incrementa el tamaño del vocabulario las características son más discriminativas pero también son menos generalizables y aumenta la posibilidad de que haya palabras visuales que agrupen únicamente ruido, es decir, puntos clave similares pueden pertenecer a diferentes palabras visuales [19].

En la literatura no hay un consenso exacto sobre el tamaño que debe tener el vocabulario. También depende de las imágenes que se utilicen y de la aplicación. Los trabajos que se han hecho comprenden tamaños entre varios cientos hasta decenas de miles [25], aunque se ha demostrado que con vocabularios de mayor tamaño es más probable que sea más precisa la clasificación sin quitar de vista el costo computacional [17] [33]. A mayor tamaño del vocabulario mayor separabilidad lineal de las clases, incluso puede llegar a un punto en que no sea necesaria la utilización de un *kernel* en el entrenamiento del algoritmo clasificador.

Existen otras técnicas para la construcción de un vocabulario como son las investigaciones de Perronnin [47] donde define el concepto de vocabulario universal y vocabulario adaptado. Un vocabulario universal es aquel que se construye con todas

las imágenes posibles de todas las clases. Un histograma construido con este vocabulario no es lo suficientemente poderoso como para ayudar a distinguir las clases. Un histograma adaptado es aquel que se construye con imágenes que pertenecen a una misma clase. Si una imagen pertenece a una clase c es recomendable describirla con las palabras visuales de esa clase c que con las palabras del vocabulario universal, pero si la imagen pertenece a otra clase entonces las palabras del vocabulario universal la describirán mejor. Se pueden construir histogramas combinados en base a estos vocabularios de manera que una imagen puede ser descrita con un conjunto de estos histogramas. Cada uno contiene una contribución del vocabulario universal y del adaptado.

2.1.2 Información espacial

La principal desventaja del algoritmo BoVW descrita en la literatura es la carencia de información espacial entre las palabras visuales porque el histograma solo cuenta la ocurrencia de ellas y no su relación espacial. La falta de relación espacial entre las palabras visuales puede ser tanto una ventaja como una desventaja en dependencia de la aplicación. Por un lado, mostrando la ocurrencia de las palabras visuales, no la relación espacial entre ellas, se obtiene una significativa flexibilidad a los puntos de vista y cambios de posición. Por el otro lado la relación espacial entre los puntos clave puede ser un importante factor discriminatorio que BoVW pierde [35]. Si se tiene una imagen y se divide en varias piezas como si de un rompecabezas se tratase donde cada pieza corresponde a una palabra visual, el histograma de la imagen armada y desarmada serán iguales. Esto ocurre si no se tiene en cuenta la información espacial.

Existen algunas técnicas para incluir la información espacial ya que la misma puede contener información útil para la tarea de clasificación. Una forma es dividir la imagen en rectángulos de iguales dimensiones, se calculan las palabras visuales de cada una de estas regiones y se concatenan las características de estas regiones dentro de un vector característico de mayor dimensión [25]. Dividiendo la imagen en $m \times n$ regiones aumentan las dimensiones del vector de rasgos a $m \times n$ veces, elevando el costo computacional.

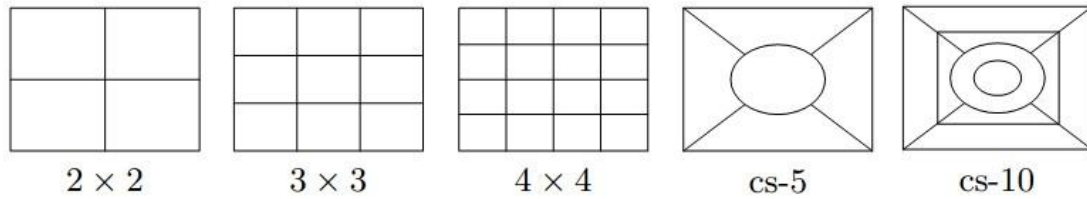


Figura 2.1: Formas de dividir la imagen para incluir la información espacial.

Existen otras formas de dividir la imagen a parte de la división en rectángulos como se puede observar en la figura 2.1. Esta propuesta fue publicada por Viitaniemi en 2009 [48]. Además de introducir la técnica de *soft tiling* frente al tradicional *hard tiling* demostrando su superioridad en las pruebas que realizó. El *soft tiling* consiste en asignar los puntos de interés no solamente a una región espacial de la imagen (tradicional *hard tiling*) sino en diferentes regiones en dependencia de la cercanía del punto de interés a la región. Esto se determina mediante un conjunto de máscaras suavizadas. El *soft tiling* es muy parecido al pesado suave que se utiliza en la cuantificación vectorial para la construcción del histograma de palabras visuales que se explica en la próxima sección.

Otra técnica utilizada para incluir la información espacial es mediante una pirámide espacial que propone Lazebnik en 2006 [49] y que se menciona también en [48]. Esta consiste en concatenar los vectores característicos obtenidos de diferentes niveles de divisiones de la imagen, es decir de divisiones rectangulares de 1×1 , 2×2 y 4×4 . Donde se obtiene un vector característico de mayor dimensionalidad que cuando se utiliza un solo nivel.

Cuando se utilizan técnicas para incluir información espacial hay que tener en cuenta el tamaño del vocabulario. Es mejor utilizar vocabularios pequeños cuando se divide la imagen en muchas regiones (como por ejemplo 6×6) que vocabularios de gran tamaño y viceversa [48].

En la práctica, por la dimensión del descriptor asociado a cada punto, muchas de las características extraídas se solapan. Implícitamente esto provee cierta dependencia geométrica, o sea, cierta dependencia espacial entre los descriptores. Esto quiere decir que, aunque no se tuviese en cuenta la mayor parte de la información espacial, siempre hay una pequeña porción presente.

2.1.3 Asociación de pesos en los histogramas de palabras visuales

En la representación original de BoVW existen varias técnicas para asignar pesos a los histogramas. Por ejemplo, dos descriptores son considerados idénticos si son fijados a la misma palabra visual (centroide) o completamente desiguales si son asignados a diferentes palabras visuales. Puede suceder que dos rasgos visuales muy cercanos, sean dados a palabras visuales diferentes (figura 2.2). En la práctica este tipo de asignación (pesado duro) conduce a errores de cuantificación debido a la variabilidad en el descriptor. Variabilidad que puede venir de diversas fuentes como el ruido, cambios de iluminación o inestabilidad en el proceso de detección.

El pesado suave o asignación suave (*soft assignment*) fue propuesto por Philbin [50] para contrarrestar el error de cuantificación. Consiste en asignar un peso a las r palabras visuales más cercanas (típicamente $r = 3$). Este peso está determinado por la expresión:

$$w(d) = e^{-\frac{d^2}{2\sigma^2}} \quad (2.1)$$

Donde d es la distancia euclidiana del descriptor al centroide y σ es el parámetro del método.

En la figura 2.1 se muestran un conjunto de descriptores (números) y palabras visuales con su centroide (letras). Si se utiliza asignación dura los descriptores 1, 2, 3 serían asignados a la palabra visual A y 4 a C , nunca 3 y 4 serían asignados a la misma palabra visual estando incluso muy cerca en el espacio de características. Sin embargo en el pesado suave 3 y 4 serían asignados a A , B y C con sus correspondientes pesos.

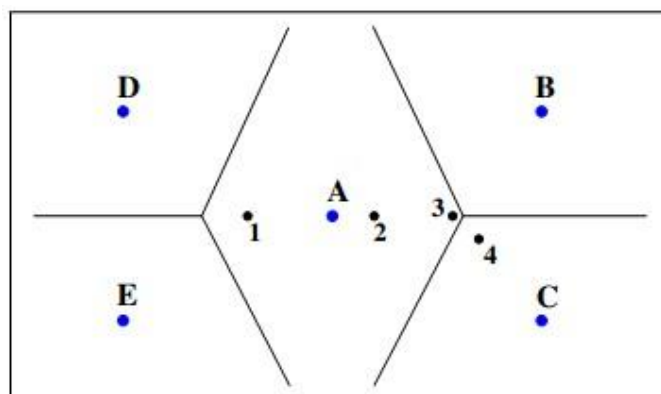


Figura 2.2: Pesado suave vs asignación dura.

2.2 Historias de usuario

Las historias de usuario conforman la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales [51]. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración. A continuación, se describen las historias de usuarios más importantes para el diseño e implementación del sistema.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar imagen digital en formato IMG.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Felipe Rodríguez Arias	Iteración Asignada: Primera iteración
Prioridad en Negocio: Alta	Puntos Estimados: 8 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:
Objetivos: Conocer el tipo de formato IMG entregado por los equipos de rayos X NUCTECH.	
Descripción: El análisis de las imágenes de equipos de rayos X del fabricante NUCTECH debe permitir conocer la estructura interna del formato digital IMG. Es necesario detectar por separado la información de la cabecera y los datos de los pixeles de la imagen, además de permitir la visualización de la imagen resultante.	
Observaciones: Los equipos de rayos X del fabricante NUCTECH guardan la información de las imágenes de rayos X obtenidas de un escaneo en un formato digital con extensión .IMG. Estos ficheros son de estructura desconocida y solo son leídos por el software del equipo de inspección.	

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar detección de armas cortas.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Felipe Rodríguez Arias	Iteración Asignada: Segunda iteración
Prioridad en Negocio: Alta	Puntos Estimados: 12 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:
Objetivos: Detectar de forma automática la presencia o no de armas cortas en regiones de interés de una imagen de rayos X.	
Descripción: La gestión de la detección de armas cortas permitirá la clasificación de una región de interés de una imagen de rayos X según su contenido. Si la región seleccionada contiene armas cortas la clasificación devolverá un valor de 1, en caso contrario -1. Se empleará como algoritmo de clasificación el método BoVW, centrando su funcionamiento en la tercera etapa del procedimiento general. Los procedimientos a desarrollar serán la extracción de características, la cuantificación vectorial, la representación de los datos mediante un histograma de palabras visuales y finalmente la clasificación del histograma obtenido.	
Observaciones: Ninguna observación	

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Localizar detección.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Felipe Rodríguez Arias	Iteración Asignada: Tercera iteración
Prioridad en Negocio: Alta	Puntos Estimados: 8 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:

Objetivos:

Extender el uso del algoritmo clasificador para su utilización en una imagen de rayos X que represente una escena real de un proceso de inspección a través de un equipo NUCTECH.

Descripción:

La ejecución de una ventana deslizante sobre una imagen de rayos X debe permitir la localización en la imagen de un arma corta. El sistema debe permitir la configuración de los parámetros del tamaño de la ventana y el paso de la ventana; además de mezclar en una sola detección general detecciones que se solapen y pertenezcan a un mismo objeto peligroso.

Observaciones: Ninguna observación

2.3 Funcionalidades del sistema

Las funcionalidades del sistema no son más que las condiciones o capacidades que tienen que ser alcanzadas por el sistema para satisfacer las necesidades del cliente. A continuación, se exponen los requisitos funcionales identificados:

FS 1 - Gestionar imagen digital en formato IMG.

- FS 1.1 – Determinar variables de la cabecera de datos: Obtiene información necesaria para conocer el tipo de software del equipo de inspección que se utilizó para generar el fichero IMG, además de propiciar otras variables como son, la intensidad máxima de los rayos X, el identificador del sistema de inspección, el tipo de formato empleado, el número de secuencia.
- FS 1.2 – Obtener los datos de la imagen de rayos X: Es necesario conocer el ancho y largo de la imagen, la cantidad de canales, el valor de la matriz de píxeles por canal y la tabla de colores que emplea.
- FS 1.3 – Transformar el espacio de colores de HSI a RGB.
- FS 1.4 – Realzar los bordes: Filtra la imagen resultante para obtener bordes más prominentes que faciliten la visualización.
- FS 1.5 – Visualizar la imagen de rayos X.

FS 2 - Gestionar detección de armas cortas.

- FS 2.1 – Transformar el espacio de colores de BGR a HSV.
- FS 2.2 – Obtener canales de la imagen: Obtiene por separado la información de cada uno de los canales de la imagen en correspondencia con el espacio de colores seleccionado.
- FS 2.3 – Extraer descriptores de la imagen objetivo: Permite la obtención de los puntos característicos de la imagen.
- FS 2.4 – Calcular histograma de palabras visuales: Representa los descriptores como histogramas de palabras.
- FS 2.5 – Clasificar un histograma: Dado un histograma de palabras visuales, lo clasifica como arma o no arma.

FS 3 – Localizar detección.

- FS 3.1 – Definir variables de la ventana deslizante: Las variables a configurar son el ancho, el largo, el paso y la escala de ventana. (La ventana deslizante tiene forma de rectángulo).
- FS 3.2 – Mostrar múltiples detecciones: Luego del recorrido de la ventana deslizante por la imagen de la escena de rayos X, se mostrarán de forma pronunciada las ventanas.
- FS 3.3 – Mezclar las detecciones superpuestas.

2.4 Historias técnicas

Las historias técnicas definen las cualidades que el producto debe tener, es necesario pensar en estas propiedades como características que hacen al producto atractivo, usable o confiable, especifican propiedades del sistema, como las restricciones de entorno o de implementación, así como la facilidad de mantenimiento, de interfaz, entre otros.

- ✓ **HT 1 - Portabilidad:** El componente de software a desarrollar debe contar con características que permitan su despliegue en varios sistemas operativos.

- ✓ **HT 2 – Entorno e Implementación:** Se utilizará el lenguaje de programación C/C++ para la implementación. El resultado de esto será una librería que pueda ser utilizada desde otras aplicaciones.
- ✓ **HT 3 - Rendimiento:** El sistema debe contar con cortos tiempos de respuesta y velocidad en el procesamiento de la información con el propósito de favorecer la agilidad del procedimiento. El análisis de una imagen no debe exceder los 10 segundos.
- ✓ **HT 4 – Hardware:** Los requerimientos mínimos de hardware que satisfacen los demás requerimientos son una PC Core2 Duo a 3.0 GHz, 4 GB de RAM, 160 GB de disco duro.

2.5 Usuarios del sistema

El componente de software propuesto, por ser concebido como una librería, presenta un solo actor que puede tener acceso sin distinción a todas las funcionalidades del sistema ya sea mediante una llamada directa a sus métodos o a través de la construcción de un envoltorio (*wrapper*) que permita su utilización desde otras tecnologías.

Tabla 2.1: Actores del sistema

Actores del sistema	Justificación
Aplicación cliente	Es el sistema externo que solicita los servicios de este componente de software o librería.

2.6 Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Son soluciones basadas en la experiencia y que han demostrado su efectividad en el transcurso del tiempo, seguidamente se muestran algunos de los principales patrones de diseño empleados.

2.6.1 Singleton

Corresponde a patrones de diseño que solucionan problemas de creación de instancias. El patrón de diseño *Singleton* (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón *Singleton* se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).

Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

El patrón *Singleton* provee una única instancia global gracias a que:

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea invocado directamente.

2.6.2 Builder

El patrón de diseño *Builder* (Constructor) es usado para permitir la creación de una variedad de objetos complejos desde un objeto fuente, el objeto fuente se compone de una variedad de partes que contribuyen individualmente a la creación de cada objeto complejo a través de un conjunto de llamadas a interfaces comunes. Su intención es la de abstraer el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto, de tal forma que el mismo proceso de construcción pueda crear representaciones diferentes. Entre las ventajas de su implementación podemos encontrar las siguientes:

- Reduce el acoplamiento.

- Permite variar la representación interna de estructuras compleja, respetando la interfaz común de la clase *Builder*.
- Se independiza el código de construcción de la representación. Las clases concretas que tratan las representaciones internas no forman parte de la interfaz del *Builder*.
- Cada *Builder* concreto tiene el código específico para crear y modificar una estructura interna concreta.
- Permite un mayor control en el proceso de creación del objeto.

2.6.3 GRASP

En diseño orientado a objetos, GRASP son *patrones generales de software para asignación de responsabilidades*, es el acrónimo de "GRASP (*object-oriented design General Responsibility Assignment Software Patterns*)". Aunque se considera que más que patrones propiamente dichos, son una serie de buenas prácticas de aplicación recomendable en el diseño de software.

Dentro de sus recomendaciones se encuentran los conceptos de cohesión y acoplamiento, los cuales, si bien no están íntimamente relacionados, se sugiere tener un mayor grado de cohesión con un menor grado de acoplamiento. De esta forma se tiene menor dependencia y se especifican los propósitos de cada objeto en el sistema.

Si existe una alta cohesión en un sistema, la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Por otra parte, el acoplamiento informático indica el nivel de dependencia entre las unidades de software de un sistema informático, es decir, el grado en que una unidad puede funcionar sin recurrir a otras; dos funciones son absolutamente independientes entre sí (el nivel más bajo de acoplamiento) cuando una puede hacer su trabajo completamente sin recurrir a la otra. En este caso se dice que ambas están desacopladas. El bajo acoplamiento entre las unidades de software es el estado ideal que siempre se intenta obtener para lograr una buena programación o un buen diseño. Cuanto menos dependiente sean las partes que constituyen un sistema informático, mejor será el resultado. Sin embargo, es imposible un desacoplamiento total de las unidades.

2.7 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas. El diseño del diagrama de clases del sistema ha sido dividido en dos secciones principales que definen las responsabilidades de las clases según la tarea que resuelven, es por esto que se han definido dos subsistemas de implementación *img* y *bovw*.

La figura 2.3 muestra el diseño del diagrama de clases del subsistema de implementación que ejecuta el procedimiento basado en el algoritmo BoVW. Las clases *BoVW* y *PHOW* implementan los algoritmos descritos anteriormente. Se puede observar que ambas clases delegan la inicialización de sus atributos en instancias de las clases abstractas *BoVWBuilder* y *PHOWBuilder* respectivamente, que a la vez heredan de la clase abstracta *Builder*. Como las clases *Builder* son abstractas y se definen como atributos en las clases que representan los algoritmos (*BoVW* y *PHOW*), es necesario crear una clase hija que implemente e inicialice los valores de los atributos de las clases padres (*BoVWBuilder* y *PHOWBuilder*) para una configuración del algoritmo que se desee probar; es por ello que se deben implementar las clases *BoVWBuilderImpl* y *PHOWBuilderImpl*. Este diseño permite extender la utilización de los algoritmos a nuevas implementaciones con solo heredar de alguna de las clases *Builder* y sobrescribir la inicialización de los atributos que se deseen. Esta representa una de las principales ventajas de la implementación del patrón de diseño *Builder*.

Se puede observar también la clase plantilla *Singleton* que implementa el patrón de diseño con el mismo nombre. Su utilización viene dada principalmente para las instancias de la clase *BoVW*, la cual interesa que sea única debido a que, en el proceso de creación de los objetos de esta clase se cargan varios ficheros de texto que guardan información acerca del entrenamiento del clasificador. Con el patrón de diseño *Singleton* se garantiza que esto ocurra solo una vez para todas las instancias de la clase *BoVW*.

Por otra parte las clases *ImageProcess* y *Rectangle* son las encargadas de la ejecución del algoritmo clasificador sobre una ventana deslizante que se proyecte a lo largo y

ancho de la imagen de rayos X. La clase *Util* contiene métodos estáticos de utilidad para el resto de las clases del espacio de nombres definido.

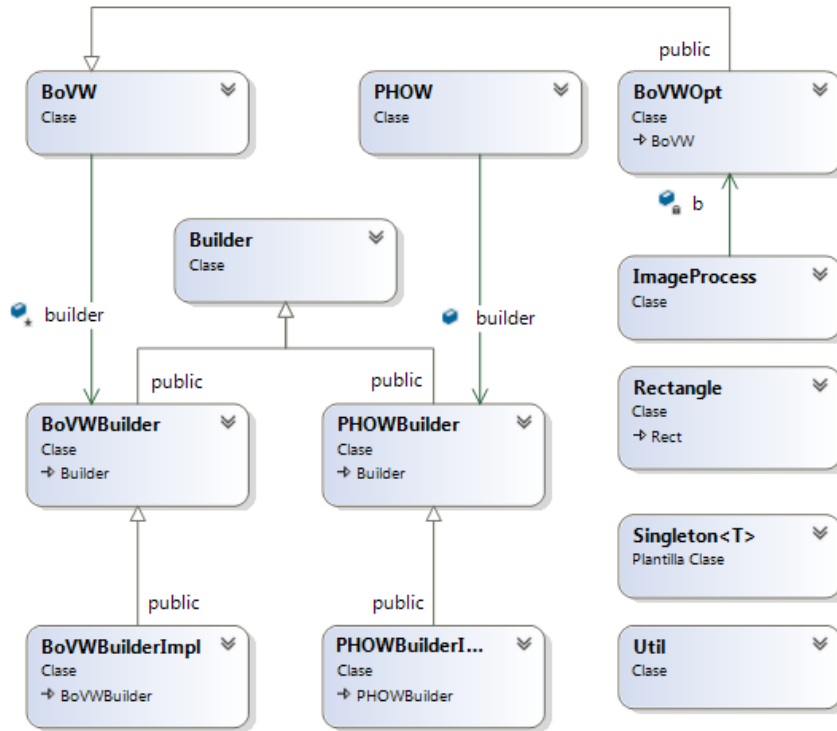


Figura 2.3 Diagrama de clases del subsistema *boww*

La figura 2.4 describe los atributos y las funciones que permiten la lectura de las imágenes en formato IMG creadas en el equipo de inspección por rayos X. De estos ficheros se debe conocer la estructura de la cabecera de datos que identifican el tipo y la versión de software del equipo en que fue creada, además de los datos de los píxeles de los distintos canales de la imagen de rayos X. La clase *HSITable* almacena la información referente a la paleta de colores empleada para poder visualizar el modelo de color “colorize”. La clase *ImageDataControl* define la estructura interna de la información de los píxeles como es almacenada en los ficheros de imágenes en formato IMG.

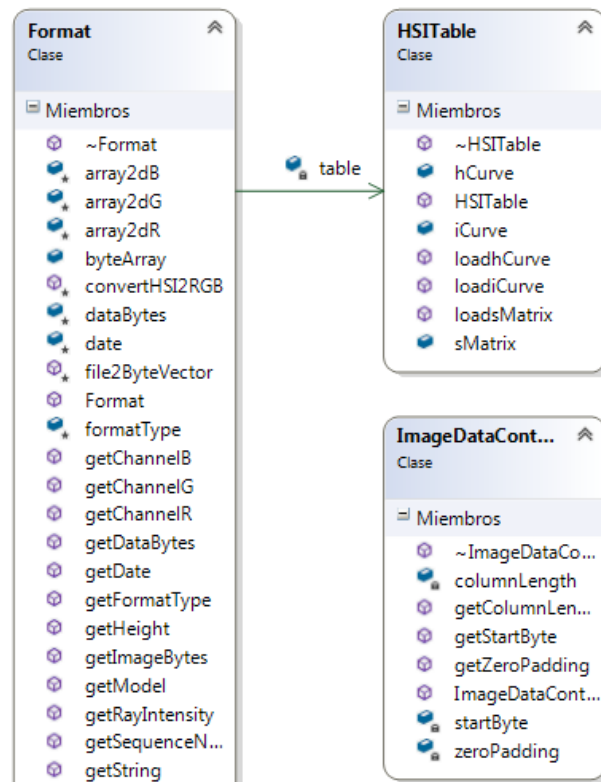


Figura 2.4 Diagrama de clases subsistema *img*

2.8 Conclusiones parciales

En el desarrollo de este capítulo se han mostrado elementos de análisis del algoritmo BoVW con vistas a facilitar el diseño y la implementación del componente de software. Además la utilización de varios patrones de diseño en la construcción del diagrama de clases del sistema, ha permitido obtener una aplicación robusta, extensible y de fácil mantenimiento, sustentado bajo el paradigma de la programación orientada a objetos. A través de la metodología de software XP se definieron las funcionalidades del sistema requeridas por el cliente, así como los requerimientos que deben cumplirse en la fase de desarrollo.

CAPÍTULO 3: IMPLEMENTACIÓN Y RESULTADOS

En este capítulo se describen las fases de implementación y pruebas ejecutadas en el proceso de desarrollo del software. Además serán explicados los aspectos técnicos y de programación más importantes, teniendo en cuenta el diseño de clases propuesto en el capítulo anterior. Se presenta la estructura identificada del formato IMG, así como los resultados obtenidos en diversos experimentos realizados y su valoración respecto otros estudios referenciados en la literatura.

3.1 Estructura identificada del formato de imagen IMG de equipos de rayos X

En esta etapa de la investigación se registraron un conjunto de 717 imágenes de rayos X en formato IMG de escenas reales obtenidas en el proceso de inspección utilizando diferentes equipos del fabricante NUCTECH. La tabla 3.1 muestra la distribución de las imágenes evaluadas por equipos.

Tabla 3.1 Distribución de imágenes utilizadas por modelo de equipo

Equipo de rayos X	Cantidad de imágenes
Imágenes de prueba	30
CX100100T	3
CX6550B	1
CX150180S	25
CX100100TI	26
CX100100TI-H	1
CX100100TIE	13
CX150180SIE	23
CX5030T	595
Total	717

Luego de múltiples pruebas se determinó que la estructura general del formato IMG está formada por tres bloques: la cabecera, el cuerpo de datos de la imagen y la cola de datos. La cabecera posee una longitud fija, de 824 bytes y el cuerpo de datos de la imagen tiene una longitud variable en función del tamaño de los datos obtenidos en el proceso de inspección. Se ha detectado además que la cola de datos puede o no estar presente en los ficheros.

3.1.1 Estructura de la cabecera

En la estructura del formato IMG se han identificado los siguientes campos:

- etiqueta del inicio: (byte 1-16) se encuentra al inicio de todos los ficheros válidos*, estos valores no cambian de fichero a fichero, sirve para identificar el formato de fichero IMG.
 - *Del conjunto total de ficheros IMG analizados, solo cuatro no cuentan con esta etiqueta. Dichos ficheros tiene en cero todos sus bytes. Por lo cual se considera se produjo un error a la hora de generar los mismos.
- dato desconocido: (byte 17) toma valores 1 para el formato 1.
- byte tamaño de cola: (byte 21) El valor de este byte es un indicador del tamaño de los datos de cola del fichero. El valor de este byte menos tres por 60 es igual al número de bytes de los datos de cola.

$$LengthCola[bytes] = (B21 - 3) * 60 \quad (3.1)$$

- Igualmente se cumple que el número de bytes del fichero IMG menos el tamaño del campo de datos de la imagen (imageByte) y el tamaño de la cola es constante, igual a 824 (número de bytes de la cabecera)

$$fileLength[bytes] = headerLength[bytes] + imageByte[bytes] + lengthCola[bytes] \quad (3.2)$$

- byte de formato: (byte 23) toma valores superior a 100. Define el formato de fichero IMG en dependencia de la versión del software con que fue creado el archivo.
- dato desconocido: (byte 25) ha tomado valores de 103 y 105.
- etiqueta desconocida: (byte 57), estos valores no cambian de fichero a fichero, toma siempre valor 3.
- etiqueta desconocida: (byte 61-62), estos valores no cambian de fichero a fichero, toman valor 56 y 2 respectivamente.
- etiqueta temporal: (byte 77-198, 333-445 y 589-602), aparece con imágenes almacenadas temporalmente en el equipo CX5030. No cambian entre imágenes.
- etiqueta desconocida: (byte 637), toma valor 1 para todas las imágenes.
- DATABYTES: (byte 641 - 644). Este dato indica la cantidad de bytes que tiene el fichero desde el byte 648 hasta el final del bloque de datos. El byte 641 es el

byte menos significativo (LSB) y el byte 644 es el byte más significativo (MSB).

Ejemplo:

$$dataBytes = B(644) * 2^{24} + B(643) * 2^{16} + B(642) * 2^8 + B(641) \quad (3.3)$$

- ancho de la imagen: (byte 649-650) número de columnas de la imagen. El byte 649 (LSB) y 650 (MSB).
- alto de la imagen: (byte 651-652) número máximo de filas de la imagen o número de sensores del modelo. Las imágenes se recortan usualmente a un número menor de filas, restringiéndose a la zona donde aparece un objeto. Byte 651 (LSB) y 652 (MSB)
- dato desconocido : (byte 653) han aparecido valores de (1)
- dato desconocido : (byte 654) han aparecido valores (0)
- dato desconocido : (byte 655) han aparecido valores (1 y 255)
- dato desconocido : (byte 656) han aparecido valores (1 y 255)
- IMAGEBYTES: (byte 657 - 660). Este dato indica la cantidad de bytes que tiene el bloque de datos, desde el byte 825 hasta el final del bloque de datos. Byte 657 (LSB) y el byte 660(MSB). Ejemplo:

$$imageBytes = B(660) * 2^{24} + B(659) * 2^{16} + B(658) * 2^8 + B(657) \quad (3.4)$$

- número de secuencia: (byte 661 - 664). Este dato compone número de la secuencia que aparece en el nombre de la imagen. Se representa desde el byte 661 (LSB) hasta el byte 664 (MSB) y se codifica de la siguiente manera:

$$NoSecuencia = B(664) * 2^{24} + B(663) * 2^{16} + B(662) * 2^8 + b(661) \quad (3.5)$$

- dato desconocido: (byte 651 - 664).
- modelo: (byte 665 - 688) nombre del equipo, cada letra está representada por dos bytes codificado en ASCII.
- fecha y hora de la adquisición: (byte 697 - 734) cada letra está representada por dos bytes codificado en ASCII.
- ID del sistema: (byte 765 - 786) cada letra está representada por dos bytes codificado en ASCII.
- Intensidad del rayo: (byte 799 - 800) aparece en el equipos con versiones del software (V3.1.xxx.xxx), codifica el valor de intensidad del equipo hasta las décimas, en kV.

- dato desconocido: (byte 801 - 802) aparece en el equipos con versiones del software (V3.1.xxx.xxx)
- dato desconocido: (byte 803 - 804) aparece en el equipos con versiones del software (V3.1.xxx.xxx)
- dato desconocido: (byte 805 - 806) aparece en el equipos con versiones del software (V3.1.xxx.xxx)

En la figura 3.1 muestra un esquema de la estructura identificada en los archivos con formato IMG de equipos de rayos X del fabricante NUCTECH.

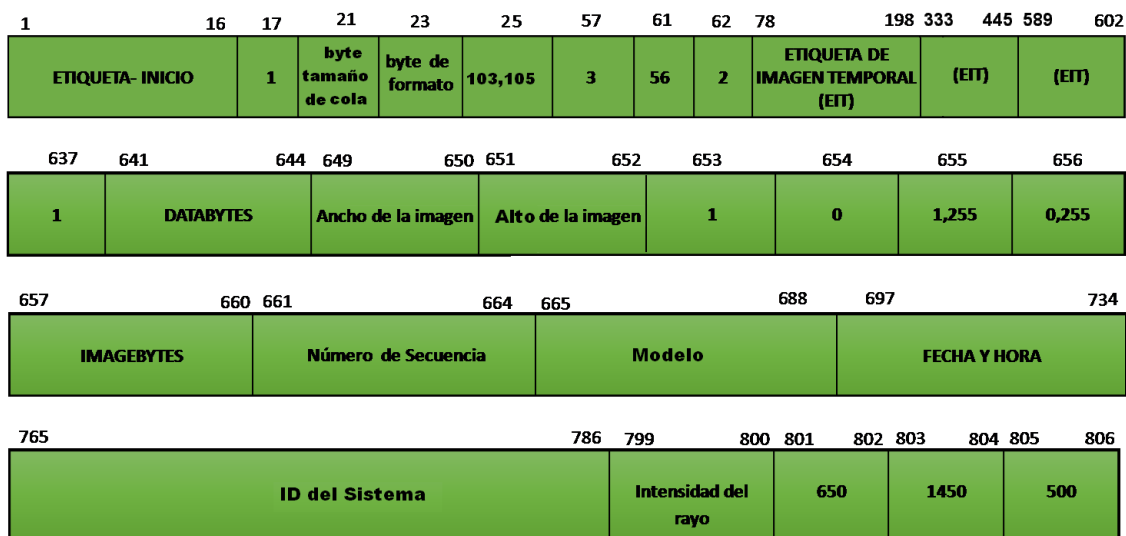


Figura 3.1: Secuencia de bytes y su significado en los archivos IMG.

3.1.2 Estructura del cuerpo de datos de la imagen

El cuerpo de datos de la imagen está dividido en segmentos de columna (CS). Cada uno de estos segmentos tiene la información correspondiente a una columna de la imagen. Los segmentos CS tienen longitud variable, pues solo codifican la información del objeto detectado (figura 3.3). El número de segmentos debe coincidir con el campo “ancho de la imagen” que se encuentra en la cabecera.

Segmento de Columna (CS): Cada segmento CS está dividido a su vez en 3 o 4 bloques de información denominados (CL, ZP, ZM y CD):

- Longitud de columna (CL): 2 bytes que codifican la cantidad de píxeles codificados en la columna. Primer byte (LSB), segundo byte (MSB).

- Longitud de fondo (ZP): 2 bytes que codifican la cantidad de píxeles en blanco añadidos al inicio de la columna.
- Marca de ceros (ZM): 4 bytes iguales a cero que marcan el inicio de los datos de la columna a continuación.
- Datos de columna (CD): $4 * CL$ bytes que contienen la información de los píxeles del objeto representado.

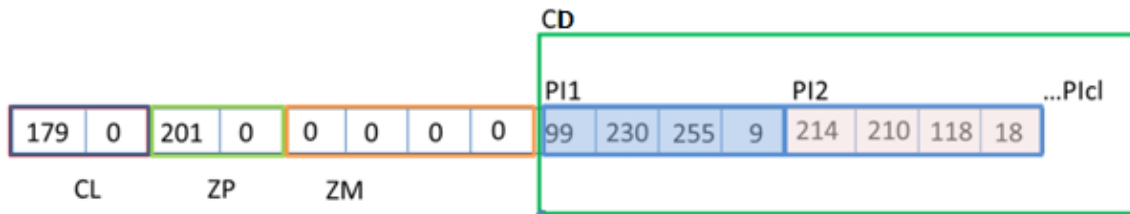


Figura 3.2: Estructura de un segmento y sus bloques CL, ZP, ZM y CD.

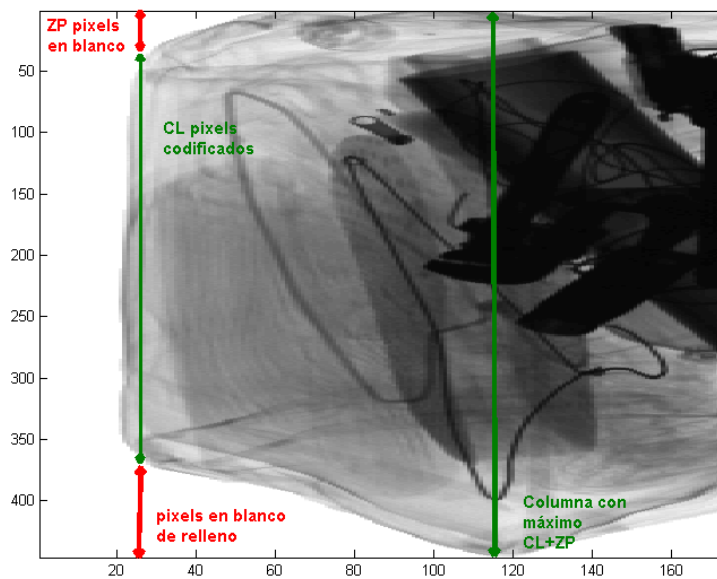


Figura 3.3: Esquema de formación de la imagen a partir de los campos CL y ZP.

En las columnas de la imagen que solo contienen ceros no aparece el bloque CD. Se determinó además que el bloque de datos de columna (CD) se divide en grupos de píxeles de 4 bytes. A la imagen formada con el i -ésimo byte de cada pixel se le denomina canal. El número de filas o altura de la imagen se obtiene de la columna cuya suma del

número de ceros al inicio y la longitud de la columna sea máxima entre todas las columnas. Para alcanzar el mismo número de filas, el resto de las columnas se rellenan de tantos píxeles.

3.1.3 Estructura de la cola de datos

De los 717 ficheros válidos analizados, 55 han presentado el bloque de cola de datos. En estos ficheros la cola de datos estuvo dividida en bloques de 60 bytes. Cada bloque comienza con una etiqueta de 12 bytes, igual en todos los ficheros ([2 0 0 0 48 0 0 0 0 0 0]) seguidos por un campo de datos de 8 bytes y finalmente el resto ha permanecido en cero. Finalmente luego de ejecutar la lectura del formato IMG en cada una de sus partes se obtiene una imagen en pseudo color semejante a la que se puede observar en el equipo de rayos X (figura 3.4).

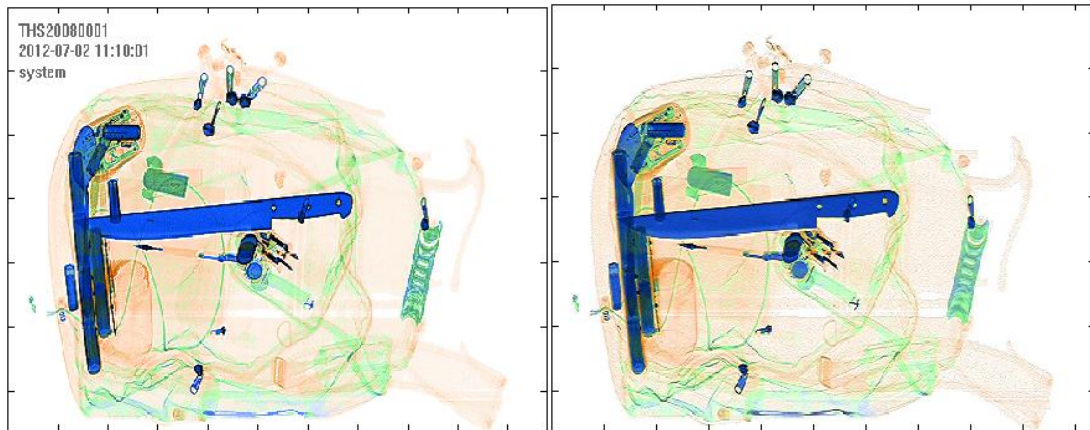


Figura 3.4: Imagen en pseudo color entregada por el equipo (izquierda). Imagen en pseudo color obtenida a partir del IMG (derecha).

3.2 Entrenamiento del clasificador

En la etapa de entrenamiento del algoritmo referente a las fases 1 y 2 del método BoVW (figura 1.5) se empleó para el desarrollo la plataforma MATLAB 2014^a y la biblioteca de funciones VLFeat. Se utilizó una base de datos para el entrenamiento compuesta por 312 imágenes positivas y 567 negativas, las cuales representan instancias de ventanas deslizantes con resoluciones de 100x100 píxeles (figura 3.5).



Figura 3.5: Instancias de ventanas deslizantes utilizadas como imágenes en el entrenamiento. Imágenes positivas (izquierda). Imágenes negativas (derecha).

En el entrenamiento se siguieron las pautas descritas por Castro [20] para el método BoVW, debido a los buenos resultados del experimento aplicado y principalmente por el tipo de imágenes de rayos X contenidas en la base de datos. Estas imágenes pertenecen a equipos de inspección de energía dual, de amplia difusión en la actualidad; las cuales permiten una estimación de la densidad y el número atómico de los materiales mediante tonos de color que van desde el naranja para los materiales orgánicos hasta el azul para los materiales inorgánicos, pasando por el verde para los materiales intermedios.

Se empleó un muestreo denso en la extracción de características mediante el algoritmo *PHOW* (*Pyramid Histogram Of Visual Words*) propuesto por [32] teniendo en cuenta que los objetos en las imágenes de rayos X poseen poca textura y es necesario obtener más información de ellos para su clasificación. PHOW realiza un muestreo denso de puntos con espacio de M píxeles (se seleccionó $M = 4$) a distintas escalas y está basado en los descriptores de SIFT, [26]. Las escalas se definen modificando el ancho de la ranura espacial del descriptor SIFT, en este caso se seleccionaron los valores 4, 6, 8 y 10 píxeles respectivamente. Las características fueron extraídas en cada uno de los canales de la imagen en el espacio de colores HSV.

En el agrupamiento de las características se empleó el algoritmo de clasificación no supervisada *kmeans*. Se construyó un vocabulario universal de 1000 palabras utilizando las características extraídas de todas las regiones de las imágenes en la base de datos. Los histogramas se construyeron normalizados y usando asignación dura (*hard assignment*). Se empleó el kernel homogéneo aditivo X^2 (chi-cuadrado) y la función de

pérdida *Square* (L2). El parámetro de regularización λ del algoritmo de entrenamiento se determinó en valor de $\lambda = 0,0001$.

El resultado final del proceso de entrenamiento está determinado por dos matrices de datos numéricos que representan el vocabulario de palabras visuales y el entrenamiento del clasificador SVM. Los datos del vocabulario se constituyen en una matriz de 1000 filas por 384 columnas que caracterizan los 1000 centroides calculados por el algoritmo *kmeans* (cada uno representa una palabra visual) y los 384 coeficientes obtenidos mediante el algoritmo PHOW. El clasificador SVM es representado por un vector de 5000 elementos ya que la utilización del kernel homogéneo aditivo X^2 eleva en un factor de 5 la dimensión de los datos de entrada.

3.3 Descripción de la base de datos de pruebas

La base de datos de prueba está compuesta por 172 imágenes positivas y 120 imágenes negativas, las cuales pertenecen a equipos de inspección de energía dual, estas no han sido utilizadas en el proceso de entrenamiento del algoritmo, representan escenas capturadas en el proceso de inspección por rayos X y tienen una resolución media de 573x515 píxeles (figura 3.6). En cada imagen de rayos X de la base de datos de prueba se ejecutan como promedio 116 ventanas deslizantes que evalúan el clasificador a lo largo y ancho de toda la imagen.

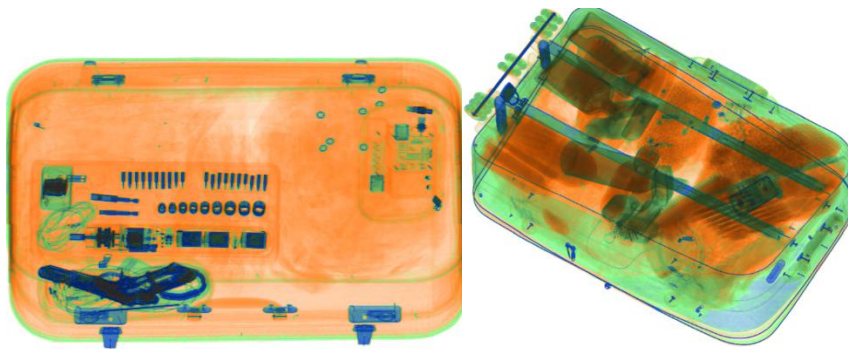


Figura 3.6: Imágenes de pruebas. Imagen positiva, que contienen al menos un arma (izquierda). Imagen negativa, sin armas (derecha).

Las armas en las imágenes de prueba aparecen en diversas condiciones con diferentes niveles de complejidad para la detección de los objetos que fueron catalogadas en los siguientes grupos, 1: objetos con oclusión propia, 2: solapadas con objetos metálicos,

3: partes no metálicas, 4: distorsión geométrica de la adquisición, 5: parcialmente desarmada y 6: visible frontal simple (figura 3.7). Existen también armas que contienen una combinación de las situaciones que se describen.



Figura 3.7: Distintas situaciones de las armas en las imágenes de entrada

3.4 Análisis de la clasificación en la fase online del algoritmo BoVW

La etapa *online* o etapa 3 del algoritmo BoVW (figura 1.5) fue implementada empleando el lenguaje de programación C/C++, con la ayuda de las bibliotecas de funciones VLFeat y OpenCV. Con el propósito de mejorar los tiempos de ejecución en esta etapa también se emplearon técnicas de programación paralela mediante OpenMP. La etapa 3 define una serie de pasos que deben ser ejecutados en cada instancia de una ventana deslizante que se traslade por toda la imagen de prueba con el propósito de localizar el objeto peligroso. El total de 292 imágenes de la base de datos de prueba fue sometido a este procedimiento y se obtuvieron los resultados que se muestran en la tabla 3.2.

Tabla 3.2: Resultados del proceso de evaluación en la etapa 3 del algoritmo BoVW

	Imágenes con armas	Imágenes sin armas	Total
Recuperados	160 (a)	6 (b)	166
No recuperados	12 (c)	114 (d)	126
Total	172	120	292

Siguiendo las métricas de *Precision* y *Recall* se obtienen los siguientes valores: $Precision = \frac{a}{a+b} = \frac{160}{166} = 0.9638$, $Recall = \frac{a}{a+c} = \frac{160}{172} = 0.9302$; con lo que se constata que se ha alcanzado un 93.02% de reconocimiento de armas cortas con una precisión de un 96.38% para la base de datos de entrada. Comparando los resultados con otras implementaciones existe una diferencia de sólo un -4% respecto a lo logrado por Castro [20], aunque se utilizan diferentes bases de datos para la evaluación se reafirma la viabilidad del método al considerarse buenos los resultados obtenidos.

Otro análisis realizado al respecto radica en conocer la cantidad de ventanas deslizantes falsas positivas donde hubo una detección acertada de la imagen de prueba (figura 3.8). Del total de ventanas evaluadas, 19952, calculadas multiplicando las 172 imágenes positivas de la base de datos por las 116 ventanas deslizantes que como promedio se ejecutan en cada imagen, han sido marcadas como ventanas falsas positivas sólo 149 lo cual representa el 0.74%.

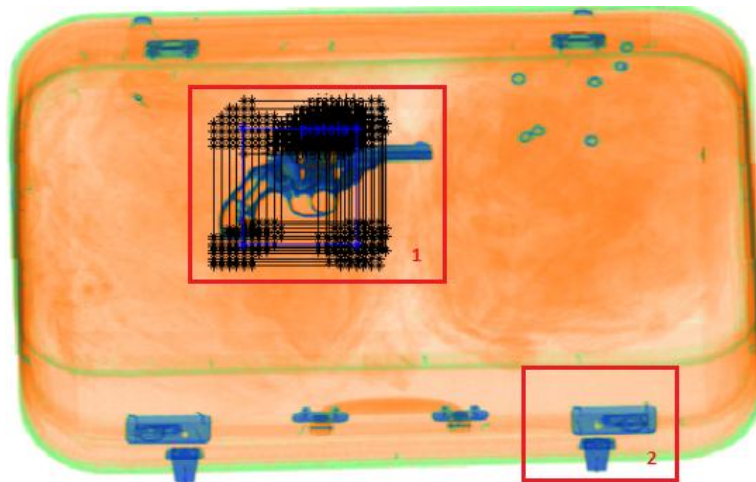


Figura 3.8: Escena obtenida en una inspección. 1: Múltiple detecciones sobre un objeto,
2: Detección falsa positiva

3.5 Análisis de rendimiento de la etapa online del algoritmo BOVW

Una de las principales características de los equipos de inspección de energía dual del fabricante NUCTECH, utilizados por la aduana, es su modularidad. Todos los componentes internos del equipo son administrados a través de una placa base estándar con factor de forma ATX (*Advanced Technology Extended*) similar a las computadoras personales de escritorio; utilizando un puerto PCI Express para el envío y recepción de datos. La placa base utiliza un procesador Intel y una versión modificada de Windows XP como sistema operativo.

La evaluación de rendimiento no se pudo realizar directamente en los equipos de inspección por problemas de logística y disponibilidad. En este caso las pruebas se realizaron en una computadora personal Core i3 de 4 núcleos a 3.1GHz, 4GB de RAM con sistema operativo Windows7. En la evaluación del desempeño se emplearon

técnicas de programación paralela para determinar la ganancia de velocidad en el tiempo de ejecución en la etapa 3 del algoritmo. Se usó una base de datos de 8700 imágenes con resolución 100x100 píxeles y se determinó el tiempo de ejecución promedio para una sola imagen variando los hilos de ejecución de la implementación mediante el método `omp_set_num_threads()` de OpenMP.

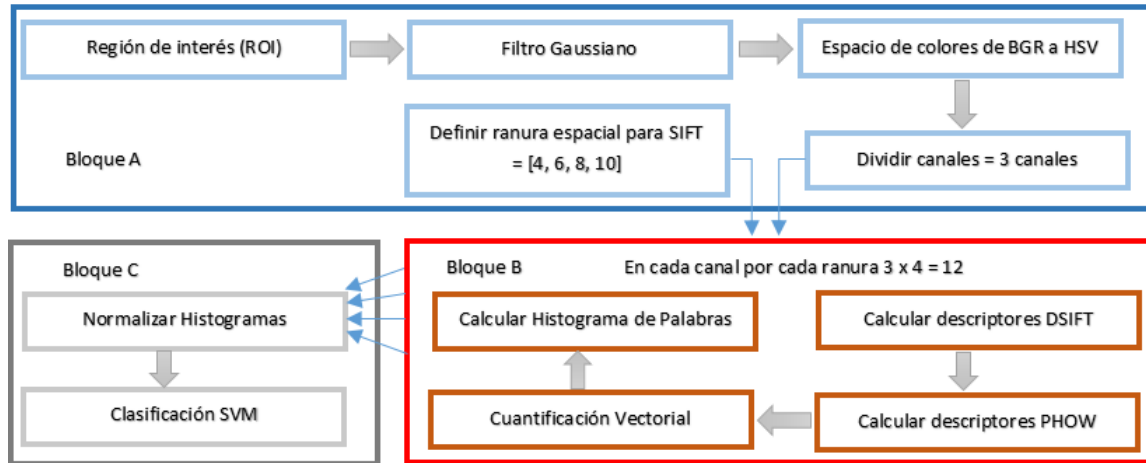


Figura 3.9: Diagrama de bloques ampliado del algoritmo BoVW en la etapa 3

La figura 3.9 muestra el diagrama de bloques de la implementación realizada con algunos ajustes para permitir la división del problema en partes independientes que permitan su análisis. El bloque A corresponde a las tareas de inicialización de variables, donde se preparan las estructuras de datos necesarios para ejecutar la extracción de características que se realiza en el bloque B. Luego se construye el histograma de palabras visuales a partir de la normalización de los histogramas obtenidos en cada canal de la imagen por cada ranura espacial del SIFT y se entrega al clasificador para su evaluación (bloque C).

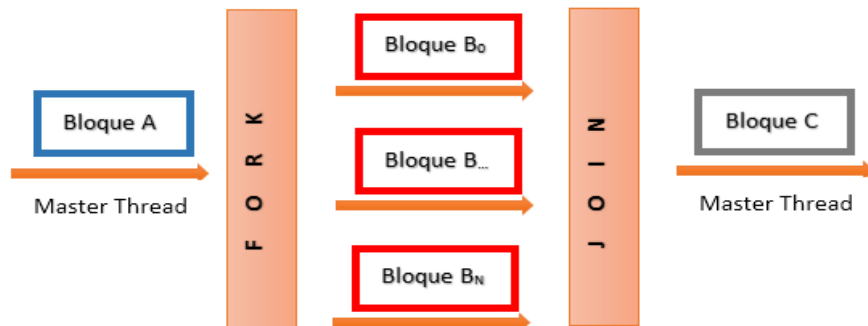


Figura 3.10: Modelo de programación OpenMP adecuado a la etapa 3 de BoVW

El bloque B efectúa un proceso repetitivo en el que no existen dependencias de datos entre una y otra iteración, lo que facilita su adecuación a un entorno de ejecución paralelo (figura 3.10). En la tabla 3.3 se muestra el comportamiento de los tiempos de ejecución en milisegundos de los diferentes bloques y se observa la ganancia de velocidad cuando se aumenta el número de procesadores empleados comparando los datos con la ejecución secuencial.

Tabla 3.3: Tiempos de ejecución (ms) de los bloques en la etapa 3

Algoritmo	Secuencial	2 Thread	3 Thread	4 Thread
Bloque A	3,2	3,2	3,2	3,2
Bloque B	99,2	52,34	40,28	31
Bloque C	1,8	1,8	1,8	1,8
Total	104,2	57,34	45,28	36

Cuando se efectúa la versión secuencial se observa que la fracción de tiempo de los bloques A y C (no paralelos) representan juntos el 5.0% del tiempo total y se mantienen inalterables en las otras versiones. La ley de Amdahl establece una cota superior en la ganancia máxima de velocidad cuando $f = 0.05$ y $N = 4$, que es $S(4)_{max} = \frac{4}{1+(4-1)*0.05} = 3.47$ y una eficiencia máxima a alcanzar de $E(4)_{max} = \frac{1}{1+(4-1)*0.05} = 0.86$; luego se comparan estos valores con la ganancia y la eficiencia obtenida en la mejor de las pruebas. Empleando los 4 núcleos del procesador se logra una ganancia de velocidad $S(4) = \frac{t(1)}{t(4)} = \frac{99.2}{31} = 3.2$, con una eficiencia $E(4) = \frac{t(1)}{4t(4)} = \frac{99.2}{4*31} = 0.8$. La diferencia entre los valores máximos ideales y los obtenidos es pequeña, se encuentran en el orden de los 0.27 y 0.06 respectivamente, lo que indica que la implementación realizada aprovecha con efectividad los recursos de multiprocesamiento del procesador.

En términos de tiempo de ejecución se puede apreciar una notable mejora cuando se efectúa la versión paralela en comparación con la versión secuencial. Según la tabla 3.3, el procedimiento secuencial completo tarda unos 104.2 ms y ha sido mejorado hasta completar la tarea en 36 ms con 4 núcleos. Si se tiene en cuenta que se evalúan 116 ventanas deslizantes como promedio en cada imagen, se ha logrado reducir el tiempo de ejecución de 12.08 a 3.77 segundos en el análisis de una muestra, lo cual representa una mejora de un 68.79% respecto al tiempo de ejecución detectado en la versión secuencial.

Capítulo 3: Implementación y resultados

Luego de observar el comportamiento en un procesador *Core i3*, se realizaron pruebas en distintas configuraciones de *hardware* con el propósito de observar la ganancia de velocidad y la eficiencia de los procesadores en la ejecución de la sección paralela del método implementado (figuras 3.11 y 3.12).

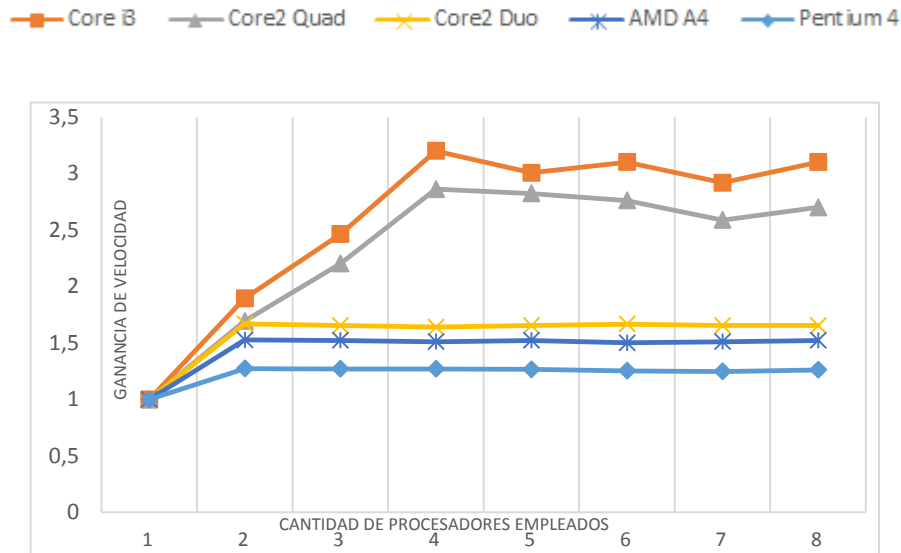


Figura 3.11: Ganancia de velocidad (Aceleración)

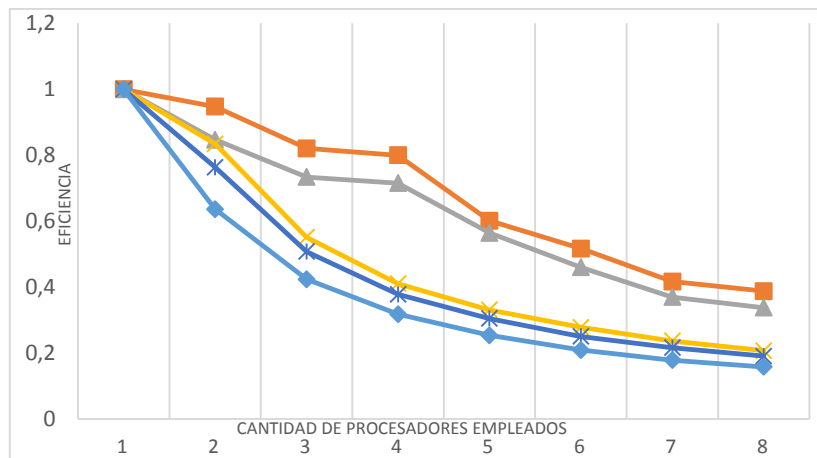


Figura 3.12: Eficiencia de los procesadores

El procedimiento `omp_set_num_threads()` de OpenMP permite establecer un número mayor de hilos de ejecución que la cantidad de núcleos de procesamiento incluidos en un procesador, no obstante el algoritmo gana en velocidad hasta el número real de núcleos del procesador. De esta forma se observa una mejora en la aceleración hasta el procesador 2 en las configuraciones *Pentium 4*, *AMD A4 3305*, *Core2 Duo*, los cuales

tienen 2 núcleos de procesamiento. En el caso del *Core2 Quad* y el *Core i3* se observan ganancias hasta los 4 hilos de ejecución.

La figura 3.13 muestra en perspectiva los tiempos de ejecución de cada uno de los bloques en los diferentes procesadores utilizando las capacidades máximas de procesamiento paralelo de cada arquitectura para esta implementación.

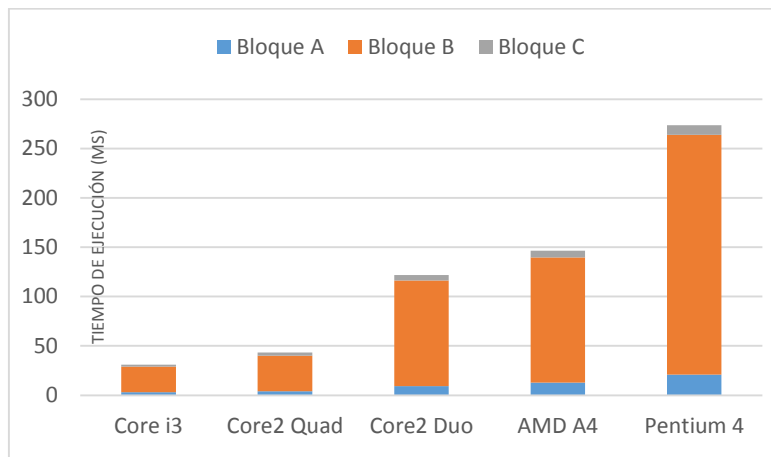


Figura 3.13: Tiempo de ejecución de los bloques del algoritmo en diferentes arquitecturas

Si bien las ganancias en velocidad y la eficiencia en las diferentes arquitecturas de hardware dependen de otros factores como la velocidad del disco duro, la memoria RAM, la velocidad y caché del CPU, el coprocesador matemático o el sistema operativo empleado, los resultados alcanzados ofrecen una idea del comportamiento del algoritmo de forma general en cada arquitectura de hardware.

CONCLUSIONES

En este trabajo se mostró el desarrollo de un procedimiento para el reconocimiento de armas cortas en imágenes de rayos X utilizando el algoritmo Saco de Palabras Visuales. Se logró además, identificar la estructura interna del formato IMG de imágenes de rayos X que utilizan los equipos de inspección del fabricante NUCTECH. En la implementación de la herramienta de software se estableció especial énfasis en la etapa *online* del método que se ejecuta en tiempo real.

Con el propósito de mejorar el tiempo de ejecución de la solución se identificaron los procedimientos que más tiempo consumían, se analizó la dependencia de datos existente y se llevaron a cabo mejoras con la adición de concurrencia en el procesamiento de datos, lográndose una reducción en el tiempo ejecución en paralelo de un 69.78% con respecto al tiempo de ejecución secuencial.


Se alcanzó una clasificación satisfactoria de un 93.02% con una precisión de un 96.38% en imágenes que contenían armas de fuego. Este trabajo, además de validar los resultados de investigaciones anteriores, contribuye a facilitar la implantación de un algoritmo de visión computacional en un entorno de producción; donde se proporcione una ayuda más a los operadores de los equipos de rayos X, incidiendo en un proceso de inspección más rápido y preciso.

RECOMENDACIONES

- ❖ Indagar sobre la información que aún se desconoce del formato IMG.
- ❖ Continuar el estudio del formato IMG con el propósito de incluir en la implementación, posibles variantes del formato en los nuevos modelos de equipos de reciente adquisición del fabricante NUCTECH. Por ejemplo el FastScanner de la ZEDM.
- ❖ Investigar una alternativa para disminuir en lo posible las detecciones falsas positivas.
- ❖ Incluir la implementación del componente de software desarrollado en el sistema XymaVision de DATYS.

ANEXOS

Anexo 1 – Aceptación de los resultados por parte del cliente.



Aduana General de la República
Oficina del Vice Jefe

ORDINARIO
Ejemplar No. _
Hoja 1

La Habana, 9 de Febrero de 2015
"Año 57 de la Revolución".

VJ/ 31

DRC. MARTHA DEL CARMEN MESA VALENCIANO
RECTORA DE LA UNIVERSIDAD DE ORIENTE

Estimada compañera:

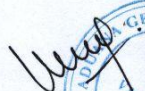
El pasado 29 de Enero asistimos a CITI para la presentación del Proyecto **"Detección de Objetos Peligrosos en Imágenes de Rayos X"**, de los autores DrC. Enrique J. Marañón Reyes, Ing. Lic. Frank Sanabria Macías, Ing. Felipe Rodríguez Arias, Ing. Pedro J. Soto Vega y el estudiante David Castro Pinol, todos pertenecientes a la Universidad de Santiago de Cuba.


Destacamos que en el proyecto incursionan en todas las variantes actuales que se citan en la literatura para la detección de objetos peligrosos en imágenes radiológicas y han llegado a conclusiones que a nuestra institución interesa, ya que los resultados obtenidos cumplen con nuestros requisitos y expectativas.

Es importante reconocer el desempeño, responsabilidad y profesionalidad de estos compañeros en el resultado de este proyecto, por lo que proponemos valore la posibilidad de continuar con estas investigaciones, ya que abordan temas de gran interés para nuestra institución en el enfrentamiento al tráfico ilícito de objetos prohibidos, siendo una herramienta de gran utilidad para nuestros operadores de equipos de detección con Rayos X o similar.

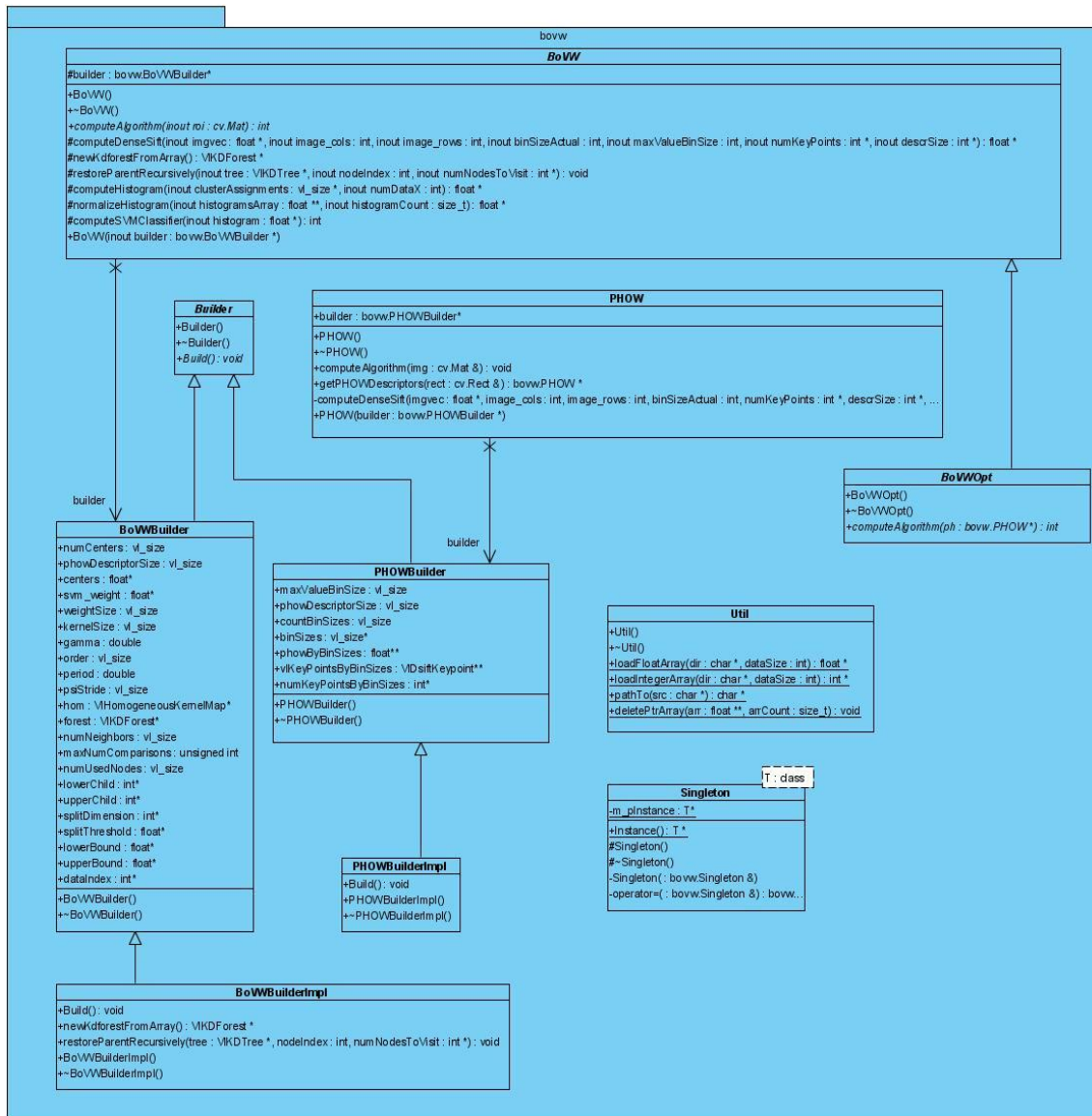
Al tanto de cualquier consideración.

Respetuosamente,

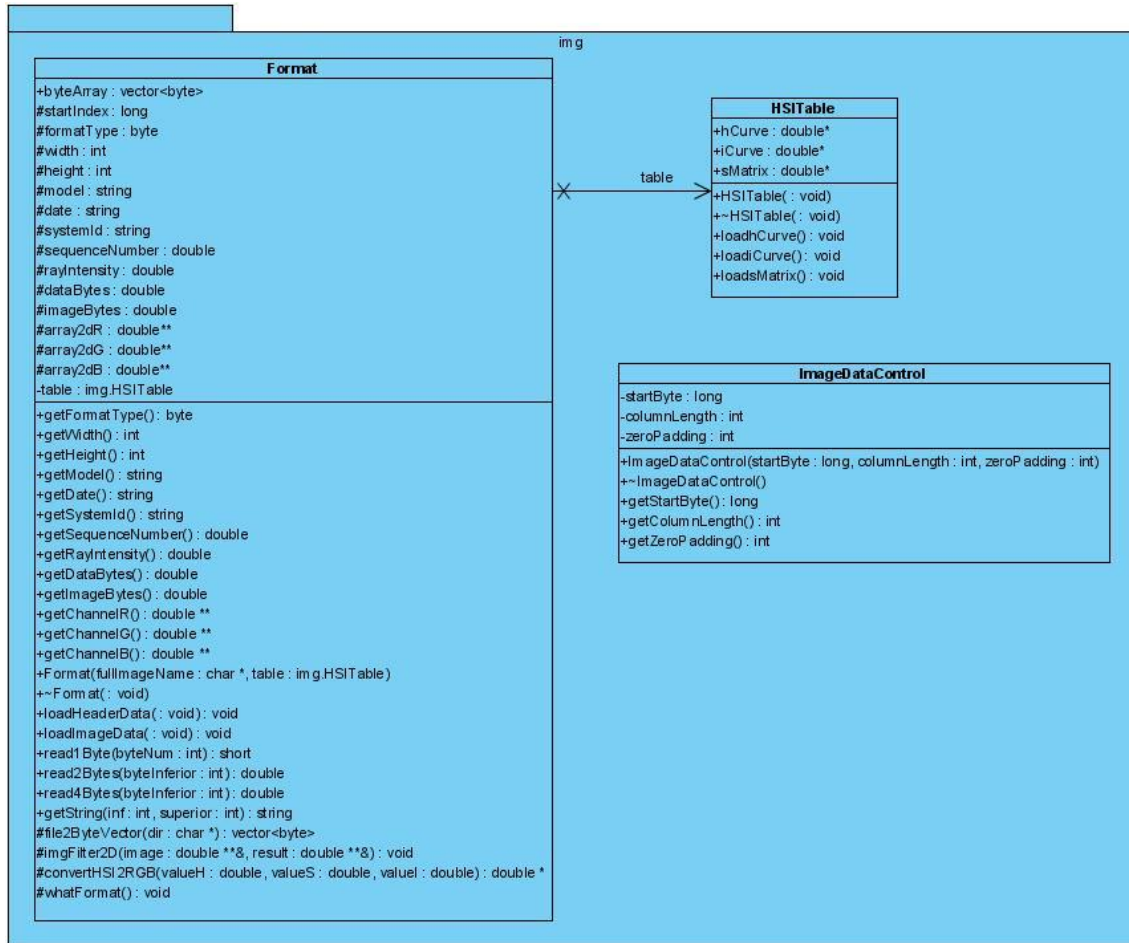

CORONEL WILLIAM PÉREZ GONZALEZ



Anexo 2 - Diagrama de clases, namespace bov



Anexo 3 – Diagrama de clases, namespace img



BIBLIOGRAFÍA

- [1] D. Mery, «X-ray Testing: The State of the Art,» *The e-Journal of Non-Destructive Testing and Ultrasonics*, vol. 18, nº 9, pp. 1-10, 2013.
- [2] G. Giakos, N. Shah y S. Chowdhury, «A novel sensor for X-ray imaging applications,» *IEEE Transactions on Instrumentation and Measurement*, vol. 49, pp. 300-306, 2000.
- [3] Q. Lu, «The utility of x-ray dual-energy transmission and scatter technologies for illicit material detection,» 1999.
- [4] S. Singh y M. Singh, «Explosives detection systems (EDS) for aviation security,» *Signal Processing, Elsevier*, vol. 83, pp. 31-55, 2003.
- [5] P. Evans, M. Robinson y H. Hon, «Multiple view dual-energy X-ray imaging,» Oct. 2001.
- [6] M. Bastan, M. R. Yousefi y T. M. Breuel, «Visual words on baggage X-ray images,» de *Springer*, 2011.
- [7] L. P. León, «Descriptores para la caracterización cuantitativa de la calidad del sistema de inspección aduanal de equipajes por rayos X: Modelo CX5030T,» 2012.
- [8] N. Corporation, «Operation Manual: X-ray Inspection System,» Beijing, China, 2010.
- [9] T. Wang, Evans y J.P.O., «Stereoscopic dual energy X-ray imaging for target materials identification,» 2003.

- [10] S. Green, M. Blumenstein, V. Muthukkumarasamy y J. Jo, «Investigation of a Classification-based Technique to Detect Illicit Objects for Aviation Security.,» 2005.
- [11] Z. Chen, Y. Zheng, B. Abidi, D. Page y M. Abidi, «A Combinational Approach to the Fusion, De-noising and Enhancement of Dual-Energy X-Ray Luggage Images,» 2005.
- [12] X. He, P. Han, X. Lu y R. Wu, «A New Enhancement Technique of X-Ray Carry-on Luggage Images Based on DWT and Fuzzy Theory,» 2008.
- [13] M. Singh y S. Singh, «Image Enhancement Optimization for Hand-Luggage Screening at Airports,» 2005.
- [14] G. Heitz y G. Chechik, «Object separation in x-ray image sets,» de *IEEE*, 2010.
- [15] M. Bastan, W. Byeon y B. T. M, «Object Recognition in Multi-View Dual Energy X-ray Images,» 2013.
- [16] D. Turcsany, A. Mouton y T. P. Breckon, «Improving feature-based object recognition for X-ray baggage security screening using primed visualwords,» de *IEEE*, 2013.
- [17] F. E. Nowak y B. Triggs, «Sampling Strategies for Bag-of-Features Image Classification,» 2006.
- [18] A. Vedaldi, V. Gulshan, M. Varma y A. Zisserman, «Multiple kernels for object detection,» 2009.
- [19] G. Csurka, C. Dance, L. Fan, J. Willamowski y C. Bray, «Visual categorization with bags of keypoints,» 2004.

- [20] D. Castro, F. Sanabria, E. Marañón y F. Rodríguez, «Reconocimiento de armas en imágenes de rayos X mediante Saco de Palabras Visuales,» *Revista Cubana de Ciencias Informáticas*, vol. 1, p. 161, 2016.
- [21] X. Shi, «Improving object classification in X-ray luggage inspection,» 2000.
- [22] C. Collough, *The AAPM/RSNA Physics Tutorial for Residents*, vol. 17, EBooks, 1997, p. 18.
- [23] J. Evans, M. Robinson y S. Godber, «Pseudo-tomographic X-ray imaging for use in aviation security,» 1998.
- [24] ESPAC, «Curso de Operadores de Equipos de Rayos X,» Empresa de Seguridad y Protección de la Aviación Civil, Ciudad de la Habana, 2014.
- [25] J. Yang, Y.-G. Jiang, A. G. Hauptmann y C.-W. Ngo, «Evaluating bag-of-visual-words representations in scene classification,» de *ACM*, 2007.
- [26] D. G. Lowe, «Distinctive image features from scale-invariant keypoints,» *International journal of computer vision*, vol. 60, nº 2, pp. 91-110, 2004.
- [27] N. Dalal y B. Triggs, «Histograms of oriented gradients for human detection,» de *IEEE*, 2005.
- [28] H. Bay, A. Ess, T. Tuytelaars y L. V. Gool, «Speeded-up robust features (SURF),» *Computer vision and image understanding*, vol. 110, nº 3, pp. 346-359, 2008.
- [29] K. Mikolajczyk y C. Schmid, «An affine invariant interest point detector,» de *European Conference of Computer Vision*, Springer, 2002, pp. 128-142.

- [30] J. Matas, O. Chum, M. Urban y T. Pajdla, «Robust wide-baseline stereo from maximally stable extremal regions,» *Image and vision computing*, vol. 22, nº 10, pp. 761-767, 2004.
- [31] A. Vedaldi y B. Fulkerson, «VLFeat: An open and portable library of computer vision algorithms,» de *ACM*, 2010.
- [32] A. Bosch, A. Zisserman y X. Munoz, «Image classification using random forests and ferns,» 2007.
- [33] K. Chatfield, V. Lempitsky, A. Vedaldi y A. Zisserman, «The devil is in the details: an evaluation of recent feature encoding methods,» 2011.
- [34] B. Vedaldi y A. Fulkerson, «Running k-means,» 2012. [En línea]. Available: <http://www.vlfeat.org/tutorials/Kmeans-clustering>. [Último acceso: 25 Mayo 2015].
- [35] K. Grauman y B. Leibe, *Visual object recognition*, Morgan and Claypool, 2011.
- [36] N. Cristianini y J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, VM Metrics, 2002, p. 93.
- [37] V. Vapnik y A. Lerner, «Pattern recognition using generalized portraid method,» 1963.
- [38] C. Burges, *A tutorial on support vector machines for pattern recognition*, vol. 2, Bell Laboratories, Lucent Technologies, 1998, pp. 121-167.
- [39] R. S. Santofimia, «Reconocimiento en Imágenes de Baja Resolución mediante Clasificadores SVM,» 2008.

- [40] A. Vedaldi y A. Zisserman, «Efficient additive kernels via explicit feature maps,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, nº 3, pp. 480-492, 2012.
- [41] A. Smeulders, M. Worring, S. Santini, G. A. y J. R., «Content based image retrieval at the end of the early years,» 2000.
- [42] A. Berstein, «Analysis of programs for parallel procesing.,» *IEEE Transactions on Computers*, vol. Oct., p. 15, 1966.
- [43] G. Amdahl, «Validity of the Single Processor Aproach to Achieving Large Scale Computing Capabilities,» 1967.
- [44] INTECO, «Curso de Desarrollo Ágil,» 2009. [En línea]. Available: <http://books.openlibra.com/pdf/Curso-de-Desarrollo-agil.pdf>. [Último acceso: Diciembre 2015].
- [45] R. Figueroa, C. J. Solís y A. A. Cabrera, «Metodologías de Software,» Junio 2008. [En línea]. Available: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>. [Último acceso: Marzo 2015].
- [46] G. Fernández Escribano, «Introducción a Extreme Programming,» 2002. [En línea]. Available: <http://www.infoab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>. [Último acceso: Noviembre 2015].
- [47] F. Perronnin, C. Dance, G. Csurka y M. Bressan, «Adapted vocabularies for generic visual categorization,» de *Computer Vision--ECCV 2006*, Springer, 2006, pp. 464-475.
- [48] V. Viitaniemi y J. Laaksonen, «Spatial extensions to bag of visual words,» de *ACM*, 2009.

- [49] S. Lazebnik, C. Schmid y J. Ponce, «Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,» de *IEEE*, 2006.
- [50] J. Philbin, O. Chum, M. Isard, J. Sivic y A. Zisserman, «Lost in quantization: Improving particular object retrieval in large scale image databases,» 2008.
- [51] C. P. Letelier Penadés, «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 2005. [En línea]. Available: <http://www.willydev.net/descargas/masyxp.pdf>. [Último acceso: 6 Marzo 2016].
- [52] M. Jones y P. Viola, «Robust real-time object detection,» 2001.
- [53] C. NUCTECH, «NUCTECH Corporation,» 2012. [En línea]. Available: <http://www.cccme.org.cn/shop/cccme5080/index.aspx>. [Último acceso: 14 Enero 2014].
- [54] V. N. Vapnik y V. Vapnik, Statistical learning theory, vol. 1, Wiley New York, 1998.