

Detección de Objetos Abandonados

Trabajo de Grado

AUTOR:

Henry Robert Fuentes Tovar

DIRECTOR:

Ing. Francisco Carlos Calderón Bocanegra. M.Sc

Facultad de Ingeniería

Dpto. de Ingeniería Electrónica

Pontificia Universidad Javeriana

Abril de 2011

Detección de Objetos Abandonados

AUTOR:

Henry Robert Fuentes Tovar

DIRECTOR:

Francisco Calderón Bocanegra



Systems, Intelligence, Robotics and Perception

Facultad de Ingeniería

Dpto. de Ingeniería Electrónica

Pontificia Universidad Javeriana

Abril de 2011

ARTICULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Solo velará porque no se publique nada contrario al dogma y a la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia.”

1	Tabla de contenido	
1.1	Índice de Figuras	II
1.2	Índice de Tablas	III
2	INTRODUCCIÓN:	1
3	MARCO TEÓRICO	3
3.1	Sistemas de Video Vigilancia Inteligente	3
3.2	OpenCV.....	3
3.3	Captura del video	3
3.4	Segmentación del Primer Plano	3
4	ESPECIFICACIONES:	6
5	DESARROLLOS:	8
5.1	Entradas y Salidas	8
5.2	Hardware del sistema	8
5.2.1	Cámara PlayStation Eye.....	8
5.2.2	Plataformas de Procesamiento.....	8
5.3	Software del Sistema.....	9
5.4	Algoritmo del Sistema.....	9
5.4.1	Conversiones de Color	9
5.4.2	Diagrama en Bloques del Algoritmo.....	10
5.4.3	Actualización de Fondo.....	10
5.4.4	Bloque de Detección de Movimiento.....	13
5.4.5	Extracción de Primer Plano.....	14
5.4.6	Detección de Regiones Estáticas.....	15
5.4.7	Detección de Objetos Abandonados.....	16
5.4.8	Filtrado de Ruido.....	17
5.5	Bases de Datos	18
5.5.1	VDOA	18
5.5.2	Otras Bases de Datos.....	18
5.6	Configuración del sistema.....	19
5.6.1	Análisis de la Escena.....	19
5.6.2	Escogencia de Parámetros	20
5.7	Carga Computacional.....	21
6	ANÁLISIS DE RESULTADOS:	22
6.1	Pruebas con videos de archivo	22

6.1.1	VDOA.....	22
6.1.2	AVSS2007.....	25
6.1.3	CAVIAR.....	27
6.2	Pruebas sobre la BeagleBoard.....	28
6.3	Condiciones de Iluminación.....	30
6.3.1	Exteriores.....	30
6.3.2	Interiores.....	31
6.4	Cambio de Escena.....	33
6.5	Objetos Pequeños.....	34
6.6	Condiciones Tiempo.....	35
6.7	Abandoned Object Detection de Matlab.....	36
7	CONCLUSIONES:.....	40
7.1	Conclusiones.....	40
7.2	Recomendaciones.....	41
8	BIBLIOGRAFÍA:.....	43

1.1 Índice de Figuras

FIGURA 1	DIAGRAMA EN Z DE LA MEDIA MÓVIL.....	5
FIGURA 2	DIAGRAMA DE BLOQUES GENERAL.....	7
FIGURA 3	DIAGRAMA DE BLOQUES EL ALGORITMO PROPUESTO.....	10
FIGURA 4	(A) FONDO ACTUAL CON SOMBRAS, (B) FONDO ORIGINAL CON SOMBRAS.....	11
FIGURA 5	(A) FONDO ACTUAL MAL CONSTRUIDO, (B) FONDO ORIGINAL MAL CONSTRUIDO.....	12
FIGURA 6	(A) FONDO ACTUAL BIEN CONSTRUIDO, (B) FONDO ORIGINAL BIEN CONSTRUIDO.....	13
FIGURA 7	(A) PÍXELES MÓVILES, (B) ESCENA ORIGINAL}.....	14
FIGURA 8	(A) PRIMER PLANO ORIGINAL, (B) PRIMER PLANO ACTUAL.....	15
FIGURA 9	MAPA DE REGIONES ESTÁTICAS.....	16
FIGURA 10	(A) DETECCIÓN DE OBJETOS ABANDONADOS 1,(B) DETECCIÓN DE OBJETOS ABANDONADOS 2.....	17
FIGURA 11	ELEMENTO ESTRUCTURANTE.....	17
FIGURA 12	(A) SIN FILTRADO, (B) CON FILTRADO.....	18
FIGURA 13	(A) OBJETO ABANDONADO VP1,(B) RUIDO VP1.....	22
FIGURA 14	OBJETO REMOVIDO.....	23
FIGURA 15	(A) OBJETO ABANDONADO A VP4, (B) OBJETO ABANDONADO B VP4.....	23
FIGURA 16	(A) OBJETOS REMOVIDOS A Y B VP4, (B) OBJETOS ABANDONADO C Y D Y REMOVIDO A VP4.....	23
FIGURA 17	(A) OBJETO ABANDONADO A, (B) OBJETOS ABANDONADOS A Y B VP6.....	24
FIGURA 18	(A) REMOVIDO A VP6, (B) OBJETOS ABANDONADO C Y REMOVIDOS A Y B VP6.....	24
FIGURA 19	(A) OBJETO ABANDONADO A VP7, (B) OBJETO ABANDONADO B VP7.....	25
FIGURA 20	(A) ABANDONO DE MALETA, (B) DETECCIÓN DEL OBJETO.....	25
FIGURA 21	(A) ESCENA DE INICIO, (B) INICIO DE DETECCIÓN.....	26
FIGURA 22	OBJETO DETECTADO.....	26
FIGURA 23	(A) ABANDONO Y PEATÓN EN CUESTIÓN, (B) DETECCIÓN DEL OBJETO Y DEL PEATÓN.....	27
FIGURA 24	(A) LEFT BAG, (B) LEFT BAG AT CHAIR.....	28
FIGURA 25	(A) LEFT BAG PICKED UP, (B) LEFT BOX.....	28

FIGURA 26 (A) IMAGEN INICIAL, (B) PRIMERAS DETECCIONES	29
FIGURA 27 DETECCIÓN DE UNA CAJA BLANCA.....	29
FIGURA 28 (A) IMAGEN INICIAL, (B) DETECCIÓN DE UNA CAJA BLANCA.....	29
FIGURA 29 (A) DETECCIÓN DE UN LIBRO Y UN CELULAR, (B) DETECCIÓN DE UNA SOMBRILLA	30
FIGURA 30 (A) ABANDONO OBJETO 1MINUTO 2:26, (B) DETECCIÓN OBJETO 1 MINUTO 2:55.....	30
FIGURA 31 (A) ABANDONO OBJETO 2 MINUTO 8:11, (B) SIN DETECCIÓN OBJETO 1 MINUTO 8:43	31
FIGURA 32 (A) ABANDONO OBJETO 1 MINUTO 3:35, (B) DETECCIÓN OBJETO 1 MINUTO 3:59	31
FIGURA 33 (A) ABANDONO OBJETO 2 MINUTO 7:35, (B) NO DETECCIÓN OBJETO 1 MINUTO 9:50	32
FIGURA 34 JUSTO ANTES DE ABANDONAR OBJETO 3.....	32
FIGURA 35 (A) ABANDONO OBJETO 3 MINUTO6:23, (B) OBJETO NO DETECTADO.....	33
FIGURA 37 (A) POSICIÓN ORIGINAL DE LA CÁMARA MINUTO 6:47, (B) CAMBIO DE POSICIÓN DE LA CÁMARA MINUTO 6:51	33
FIGURA 38 (A) RESPUESTA DEL SISTEMA MINUTO 7:49, (B) ADAPTACIÓN COMPLETA DE LA ESCENA MINUTO 11:33	34
FIGURA 39 (A) ABANDONO OBJETO MINUTO 6:2, (B) DETECCIÓN OBJETO MINUTO 6:47	34
FIGURA 40 (A) ABANDONO OBJETO 1 MINUTO 2:58, (B) DETECCIÓN OBJETO 1 MINUTO 3:28.....	35
FIGURA 41 (A) ABANDONO OBJETO 2 MINUTO 5:56, (B) DETECCIÓN OBJETO 2 MINUTO 6:38	35
FIGURA 42 (A) ABANDONO OBJETO MINUTO 7:18, (B) DETECCIÓN OBJETO MINUTO 10:04	36
FIGURA 43 (A) ABANDONO OBJETO 3 MINUTO 11:00, (B) DETECCIÓN OBJETO 3 MINUTO 11:36	36
FIGURA 44 (A) PRIMEROS DOS FALSOS POSITIVOS, (B) TERCER FALSO POSITIVO	37
FIGURA 45 (A) DETECCIÓN DEL OBJETO, (B) DETECCIÓN INTERMITENTE.....	37
FIGURA 46 SIN DETECCIÓN DEL OBJETO	37
FIGURA 47 (A) PRIMER FALSO POSITIVO, (B) SEGUNDO FALSO POSITIVO Y DETECCIÓN DEL OBJETO	38
FIGURA 48 (A) PRIMER FALSO POSITIVO, (B) SEGUNDO FALSO POSITIVO.....	38
FIGURA 49 DETECCIÓN DEL OBJETO.....	38
FIGURA 50 DIVISIÓN DE ESCENA	42

1.2 Índice de Tablas

TABLA 1 DATOS DE LAS PLATAFORMAS DE PROCESAMIENTO	9
TABLA 2 VENTANA DE TIEMPO.....	14
TABLA 3 BASE DE DATOS VDOA	18
TABLA 4 BASE DE DATOS AVSS 2007.....	19
TABLA 5 BASE DE DATOS CAVIAR	19
TABLA 6 FACTOR DE ESCALA Y PERIODO DE MUESTREO	20
TABLA 7 VELOCIDAD DE PROCESAMIENTO.....	21
TABLA 8 RESULTADOS VIDEO DE PRUEBA 1	23
TABLA 9 RESULTADOS VIDEO DE PRUEBA 4	24
TABLA 10 VIDEO DE PRUEBA 6	24
TABLA 11 VIDEO DE PRUEBA 7	25
TABLA 12 VIDEO AVSS_AB_EASY_DIVX	25
TABLA 13 VIDEO AVSS_AB_MEDIUM_DIVX.....	26
TABLA 14 VIDEO AVSS_AB_HARD_DIVX	27
TABLA 15 RESULTADOS CAVIAR	28
TABLA 16 RESULTADOS PRIMERA CAPTURA BEAGLE BOARD	29
TABLA 17 SEGUNDA CAPTURA SOBRE LA BEAGLE BOARD	30
TABLA 18 COMPARATIVA DE DESEMPEÑO.....	39

2 INTRODUCCIÓN:

Hoy en día los instrumentos tecnológicos que ayudan al ser humano a monitorear y controlar las variables de tráfico humano han tenido una gran acogida, en Bogotá, por ejemplo, el Fondo de Seguridad y Vigilancia de Bogotá, está desarrollando el proyecto SISTEMA DE VIDEO VIGILANCIA el cual tiene entre sus objetivos fortalecer el sistema de video vigilancia publica de la ciudad [1].

La detección de objetos abandonados es una de las áreas de interés en el ámbito de visión por computador, es una temática que involucra la prevención de atentados con artefactos explosivos, los cuales a lo largo de los años han cobrado vidas de seres humanos alrededor del mundo, tal como sucedió en España con los atentados del 11 de Marzo del 2004, en Bombay con los atentados del 11 de Julio de 2006 y en Colombia con el atentado al edificio de Caracol Radio el 12 de Agosto del 2010, entre otros.

Es una temática que pretende evitar situaciones catastróficas para el ser humano y su integridad, debido a los riesgos que un objeto abandonado sugiere [7] [8], diversas investigaciones alrededor del mundo han tomado lugar en este tema, de esta forma se ha convertido en el argumento para desarrollar diferentes soluciones que permitan detectar eficientemente un objeto abandonado [9][10].

Normalmente en este tipo de sistemas se suele encontrar un esquema general de solución, este consiste en cuatro etapas principalmente, segmentación del primer plano, detección de objetos, seguimiento de los mismos y detección del objeto abandonado. Alrededor de este esquema se ha desarrollado la mayoría de proyectos de investigación en el tema, aplicando en cada etapa diferentes soluciones y técnicas de tratamiento de imágenes. Una característica que resalta este esquema es el seguimiento de objetos, y la utilización de estructuras que almacenan las características de los objetos detectados. [6][8]

En algunas de las más recientes investigaciones, se ha encontrado una solución alternativa que no incluye una etapa de seguimiento de objetos o bloques del primer plano, suprimir este paso, permite que procesar el algoritmo con menor carga computacional, como se expondrá más adelante, una característica ideal para aplicaciones en tiempo real. En cambio, el algoritmo se enfoca en hacer la detección de regiones u objetos estáticos, después de hacer la segmentación de primer plano, y enseguida hacer la detección de objetos abandonados. [12][13]

Es por esta razón que los sistemas de Video Vigilancia Inteligente, implementan soluciones en el campo de visión por computador, ya que estas permiten que las tareas repetitivas que realiza el ser humano con el sentido de la vista, sean simplificadas o satisfechas [15], facilitando la tarea del operario, quien debido al cansancio visual, ocasionado por largas jornadas de trabajo, puede tener una pequeña pérdida en los detalles que puede reconocer. [3][4]

La implementación de este proyecto en un sistema real permite controlar y prevenir incidentes en espacios públicos de alta concurrencia de personas, previniendo situaciones de alta peligrosidad como atentados con artefactos explosivos. De igual forma se pueden recuperar elementos extraviados que pueden contener información importante, o elementos de alto valor y cuyo extravió pudiese representar una gran pérdida.

Este proyecto es un trabajo de investigación del grupo SIRP de la Pontificia Universidad Javeriana, en la temática de extracción de variables de tráfico urbano. La integración de varias soluciones en un mismo sistema, prometen a futuro una solución completa, sobre la cual se puede obtener rentabilidad económica.

La importancia de este tipo de sistemas radica en que un objeto abandonado en cualquier espacio abierto o cerrado, puede representar gran peligro potencial, ya que no se conoce quien es su dueño, quién es el responsable, y tampoco el contenido de este. En nuestro país y en varias ciudades del mundo ha habido gran cantidad de atentados terroristas a través de esta modalidad. Un individuo abandona un objeto cualquiera: un maletín, una caja, un regalo, etc. Y algún tiempo después el objeto resulta en un artefacto explosivo, causando lesiones físicas y psicológicas a las personas que se encuentran al rededor, y a su vez daños materiales que pueden representar perdidas catastróficas. Este tipo de ataques suelen ser inesperados, los individuos que planean este tipo de acciones no suelen ser evidentes a la hora de abandonar y accionar estos artefactos. El solo hecho de exponer la vida humana hace méritos para que esta clase de incidentes sean prevenidos.

El desarrollo y la investigación realizada durante la realización de este proyecto está guiado por el cumplimiento de lo siguientes objetivos:

- Diseñar e implementar un algoritmo de detección de objetos abandonados utilizando visión por computador (objetivo principal).
- Diseñar un algoritmo de visión por computador para la detección de objetos abandonados en una escena.
- Construir una base de datos para ser usada con algoritmos de detección de objetos abandonados en escena.
- Implementar el algoritmo de detección de objetos abandonados en escena, sobre un núcleo OMAP3.
- Especificar las condiciones mínimas de operación para un sistema que implemente el algoritmo diseñado.

En lo que sigue, este documento está escrito con el fin de mostrar el trabajo de investigación realizado. El lector podrá encontrar detalles del producto final, algunos términos clave para comprender la implementación, una descripción detallada de lo que es capaz de llegar a hacer el sistema, bajo qué condiciones y también la forma como se logra obtener este resultado. Finalmente el lector encontrará un compendio de resultados útiles que comprueban la descripción previamente dicha, y finalmente encontrará los logros obtenidos al desarrollar e implementar este proyecto.

3 MARCO TEÓRICO

Antes de hablar acerca del sistema y su desarrollo es preciso citar algunos conceptos teóricos relevantes para comprender la temática del proyecto.

3.1 Sistemas de Video Vigilancia Inteligente

Se trata de sistemas computacionales que intentan interpretar secuencias de imágenes, de forma automática, con el objetivo de extraer de ellas información efectiva, útil y fácil de comprender. Los sistemas de video vigilancia inteligente consisten en analizar el comportamiento de objetos y personas, con el fin de monitorearlos de manera eficiente, en sistemas de tiempo real. [21]

En aplicaciones de tiempo real para detectar amenazas, es necesario el acompañamiento de la interpretación de un operario entrenado para la tarea visual de monitoreo, quien debe interpretar cada señal que el sistema arroja, referente a una detección. [20]

Esta área de investigación de Visión por computador, es una de las más activas y tiene un amplio campo de aplicaciones, como control de accesos, detección de objetos abandonados, identificación de humanos, control del flujo de personas y análisis de congestiones, o detención de comportamientos anómalos, etc.

3.2 OpenCV

OpenCV (Open Source Computer Vision), es una biblioteca de visión por computador de código libre, para aplicaciones en tiempo real. Es de libre uso comercial y académico, bajo una licencia BSD. Trabaja con lenguajes C, C++, Python y próximamente con Java. Se puede ejecutar sobre los sistemas operativos, Windows, GNU/Linux y Mac OS. [38]

Es ampliamente utilizada en sistemas de visión por computador y está diseñada para aplicaciones de tiempo real con alta eficiencia computacional. Cuenta con más de 2500 algoritmos optimizados, brindando a los usuarios un conjunto completo de herramientas con las cuales desarrollar sistemas basados en visión por computador. Además contiene cerca de 500 funciones que abarcan un amplio rango de aplicaciones en visión, inspección de productos, imágenes médicas, seguridad, visión estéreo, robótica, entre otras. El uso de esta herramienta contribuye con el adelanto y perfeccionamiento de esta área de la ingeniería.

3.3 Captura del video

La captura del video consiste en tomar las imágenes de una cámara digital, bajo sus condiciones de muestreo, y resolución. Las imágenes se obtienen a través de la conexión USB de la plataforma de procesamiento. La captura también puede hacerse desde un archivo de video, previamente capturado.

3.4 Segmentación del Primer Plano

Es común encontrar que en visión por computador se hace necesario algún o algunos tipos de segmentación. En las temáticas involucradas con el estudio de objetos en la escena, en especial para aplicaciones de video vigilancia inteligente, es indispensable que el algoritmo logre hacer distinción de lo que corresponde al fondo y al primer plano de la imagen. Este es un paso esencial, ya que logra delimitar las zonas de la imagen que se deben procesar, facilitando el análisis de la captura de video. [19]

El fondo es todo aquello que hace parte del espacio como tal, es decir, las paredes, el piso, el techo, y los mobiliarios contenidos en él. En el caso de una sala de estar, la mesa del centro, las sillas, el florero y demás objetos propios de esta, son parte del fondo.

Y el primer plano es todo aquello que no pertenece al fondo, es decir, todo lo que se encuentra sobrepuesto a este y que a su vez lo obstruye en la imagen. En el mismo caso de la sala de estar, las personas sentadas, de pie o caminando, y sus objetos hacen parte del primer plano.

Usualmente el primer plano se extrae obteniendo una estimación del fondo, en seguida se debe comparar cada imagen que entra al sistema con el modelo y el resultado de esa comparación, se obtiene en una imagen binaria que divide el fondo y el primer plano en la misma imagen.

Existen diferentes estrategias para modelar el fondo, en algunas de estas no es posible adaptar cambios de iluminación, ni de movimiento, con facilidad, esto implica una estimación poco precisa, a su vez representa un problema directo en la extracción del primer plano. Su implementación para este proyecto resulta incompatible, puesto que podrían existir regiones estáticas conformadas por ruido, que conlleven a una detección errónea. Un ejemplo de esto es el algoritmo de Libro de Códigos [17], el cual construye un modelo de fondo a través Palabras de Código en un periodo de aprendizaje, una vez que terminado este periodo el sistema queda limitado a lo aprendido en este tiempo. A pesar de su eficiencia computacional, y de soportar ligeros cambio de iluminación [19], el modelo no es lo suficientemente adaptativo para la aplicación. [24]

Algunas de los modelos más usados en detección de objetos abandonados son, Diferencia Temporal, Modelos de Mezcla de Gaussianas (MMG), y finalmente el Método media móvil.

El método de Diferencia Temporal es el más simple, consiste en calcular diferencias cuadro a cuadro, después se umbraliza la imagen y así se registran los cambios de la escena, su mejor cualidad es que por su simplicidad el método tiene baja complejidad computacional, su debilidad consiste en que la aplicación no almacena ni recopila datos con los cuales comparar, además de que el modo de funcionamiento del sistema es vulnerable de incorporar demasiado ruido.

El método Modelos de Mezcla de Gaussianas, consiste en calcular una mezcla distribuciones gaussianas que describen el color para cada pixel en la imagen. El peso de los parámetros de la mezcla representa la proporción de tiempo que se encuentran presentes los colores en la escena. El algoritmo cuenta con un esquema de actualización selectivo. Este método ha sido ampliamente usado, su estimación de fondo permite manejar cambios de iluminación, movimientos periódicos, movimientos lentos, y ruido introducido por la cámara. [26] [29]

En [28] Zheng y Fan, establecen una comparación de la complejidad computacional de estos métodos y uno creado en ese trabajo. Los resultados indican que después de Diferencia Temporal, Media Móvil es el método con menor complejidad computacional, resultados que se obtuvieron con 2 videos de prueba citados allí.

El método de Media Móvil, consiste en construir el fondo a partir del promedio de los valores de cada pixel de las imágenes del video. Las medias móviles se calculan en cada iteración del algoritmo, y cada cálculo hace parte de una acumulación de valores con la cual se almacena la información. Esto implica que en todo momento se está construyendo el modelo de manera dinámica. Su modo de funcionamiento permite que cualquier cambio en la escena sea tomado en cuenta, esto lo hace completamente adaptativo. Además de esto es posible establecer la rapidez con la el algoritmo se actualiza, esto lo hace configurable y adaptable según la aplicación.

La Figura 1 Diagrama en Z de la Media Móvil, muestra cómo se actualiza la media:

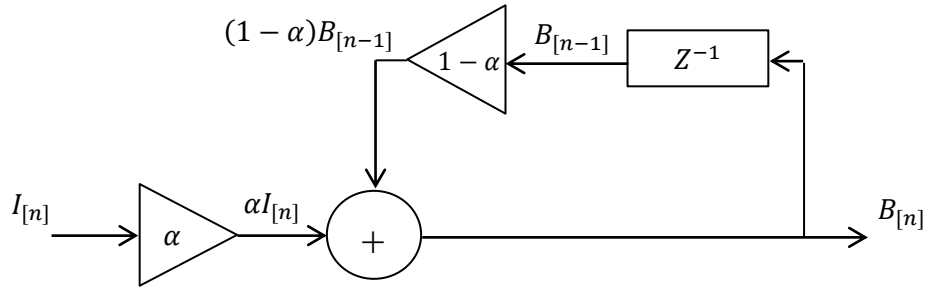


Figura 1 Diagrama en Z de la Media Móvil

Su respectiva ecuación será:

$$B_{[n]}(x, y) = (1 - \alpha) \cdot B_{[n-1]}(x, y) + \alpha \cdot I_{[n]}(x, y) \quad \text{ec. 1}$$

Donde B, corresponde al Fondo, I es la imagen actual, y α es la tasa de aprendizaje del fondo. El parámetro α establece esencialmente que tan rápido se actualiza el fondo. Desde otro punto de vista, este parámetro indica la memoria del sistema. Esta estructura es la misma de un filtro digital IIR de primer orden.

Este modelo se conoce también como promedio ponderado exponencial en movimiento o como originalmente se conoce, Media Móvil Ponderada Exponencial, por su comportamiento exponencial [27]. La escogencia del parámetro α , está determinada por lo que representa esta ecuación, y las condiciones bajo las cuales funciona de manera adecuada. En el Anexo 1, se describe el método para escoger este parámetro.

En recientes investigaciones se ha notado interés en la implementación este algoritmo, en la temática de detección de objetos abandonados, estas se han destacado por su notable reducción en carga computacional [12] [13], esencial en sistemas de video vigilancia. Este modelo resulta ser comparable en cuanto a la adaptabilidad que otorga, especialmente en espacios interiores, pero es más eficiente que este por su carga computacional.

4 ESPECIFICACIONES:

El grupo de investigación SIRP como parte de su investigación en la de extracción de variables de tráfico urbano, requiere un sistema en software que permita la implementación a la detección de objetos abandonados. La función principal del sistema es determinar por medio de un algoritmo de visión por computador, si un objeto ha sido abandonado en un espacio en particular, para esto se cuenta con una cámara fija, con una apertura de imagen fija. Que se encuentra conectada a una unidad de procesamiento. Sobre la misma se efectúa el tratamiento de imágenes que permite detectar dicho comportamiento. Por tal motivo lo que entra a la unidad de proceso es una secuencia de imágenes extraída de una cámara, y lo que sale del sistema es otra secuencia de imágenes que contiene información adicional indicando el objeto que ha sido abandonado, esta información está dada en forma de una imagen vistosa sobrepuesta a la imagen original, esto sirve de ayuda visual al operario que se encuentre en la estación de monitoreo [2] [15].

Para correcto funcionamiento del algoritmo, el espacio vigilado debe cumplir con una condición importante. Ya que el sistema está diseñado para actuar en espacios interiores, es indispensable que estos se encuentren correctamente iluminados. Esta característica luz en el área cubierta por la cámara, es necesaria debido a que cada cuadro que entra al sistema, sufre desde el inicio una transformación espacial de color, esta transformación consiste en pasar las imágenes del espacio RGB, a una Escala de Grises. Mientras que el espacio RGB está compuesto por tres dimensiones, que corresponden a los niveles de Rojo, Verde y Azul, la Escala de grises está compuesta por solo una dimensión, que corresponde a los niveles de grises desde el blanco hasta el negro, esto supone una pérdida importante de información ya que lo que queda después de la transformación son solamente las intensidades del color original. Una iluminación adecuada otorga una representación más exacta de los niveles de intensidad de los que se compone el espacio visual que se vigila. Por lo que es recomendable que el ambiente en el que se trabaje el sistema, cuente con la suficiente iluminación para que la imagen de la cámara que se esté usando de una representación correcta de la escena. Este parámetro se deja a criterio de la persona que implemente el sistema, dando como guía que si una persona puede ver los detalles del video sin esfuerzo el sistema propuesto también podrá hacerlo.

Por su parte en la ubicación de la cámara se debe tener en cuenta si el espacio cubierto por esta incluye zonas en las cuales haya personas que desempeñen labores en un mismo sitio, por ejemplo en un escritorio, donde una persona puede estar en la misma posición durante un tiempo importante, y después cambiar su posición. Entonces esa zona no es de interés pues se van a generar detecciones de forma aleatoria y sin ningún sentido.

El software de procesamiento de video, puede ser ejecutado sobre un computador, del tipo escritorio o portátil, o sobre una tarjeta de procesamiento integrado como la BeagleBoard, la cual se compone de, un microprocesador, memorias internas, puertos de interfaz como HDMI, S-VIDEO, USB, y SD, entre otros componentes. [33]

El sistema está diseñado para detectar objetos abandonados en interiores y hace frente a características del espacio abarcado por la cámara tales como cambios de iluminación progresivos, oclusiones no permanentes debido al tráfico dinámico de personas y objetos, y fondo variable, es decir en el caso en que un objeto que pertenece al fondo sea cambiado de sitio pero que sigue siendo parte de él.

Gracias a la naturaleza del sistema, es posible variar los tiempos que tarda el sistema en hacer la detección, y a su vez el tiempo que tarda la señalización del objeto detectado. Esta propiedad es útil, pues

no todos los espacios tienen la misma naturaleza de tráfico peatonal. Por ejemplo, no es lo mismo vigilar la zona de espera de ascensores que suben hasta un quinto piso, que la de unos que suben hasta un piso treinta, pues es muy probable que en el segundo caso las personas deban esperar más tiempo en la zona vigilada, que en el primer caso. De la misma forma podrían existir diferentes casos, en los que los cuales la escogencia de valores estándar para estos tiempos implicaría una detección distorsionada.

El algoritmo de detección de objetos abandonados puede representarse por 5 etapas:



Figura 2 Diagrama de Bloques General

Para hacer la detección de objetos en abandono, el sistema realiza una rutina que se encarga de capturar el video de la cámara, y verificar si existe en la escena un objeto que no es propio de ese espacio, enseguida, determina si este es un objeto que se ha quedado estático, si es así, entonces es un candidato de objeto abandonado, tras pasar por un filtro morfológico, consistente en erosión y dilatación, para separar el ruido, se obtienen los objetos abandonados, los cuales se hacen evidentes en la imagen mediante una tonalidad roja que cubre el objeto detectado.

El desarrollo de este algoritmo está cimentado en la estimación de fondo dual, que permite extraer dos versiones del primer plano sobre las cuales toman decisiones y se extrae toda la información relevante que conlleva recopilar las características adecuadas que revelan el estado abandonado de un objeto.

5 DESARROLLOS:

Los desarrollos del sistema constituyen la base sobre la cual se obtiene el resultado final y son el producto de la investigación bibliográfica y los aportes del desarrollador.

5.1 Entradas y Salidas

Las entradas del sistema consisten en las imágenes capturadas a través de la cámara, la tasa de muestreo, y la resolución con que lo hace. Sus salidas son las señales de alarma, consistentes en pigmentaciones rojas sobre la secuencia de entrada, que indican la detección.

5.2 Hardware del sistema

Con el fin de mostrar las propiedades físicas que expone el sistema es necesario conocer los componentes de hardware que constituyen el sistema, así como las entradas y salida del mismo. Fundamentalmente se requiere en el sistema una cámara de video, un hardware de adquisición y procesamiento, y una pantalla.

5.2.1 Cámara PlayStation Eye

La cámara utilizada para la captura de imágenes, es la cámara PlayStation Eye, esta cámara fue desarrollada y ha sido fabricada por Sony Computer Entertainment, Inc. (SCEI), para trabajar con la consola PlayStation 3, con el fin de permitir a los jugadores usar el movimiento de los controles para interactuar con los juegos. Para cumplir con este objetivo, el software de la cámara implementa tecnologías de Visión por computador y de reconocimiento de gestos [34].

La cámara PlayStation Eye cuenta con un chip que permite mayor efectividad en operación de poca luz, este chip cuenta con un sensor CMOS retroiluminado fue desarrollado en conjunto con OmniVision Technologies. Por otra parte la cámara puede trabajar a una frecuencia de muestreo de hasta 120 cuadros por segundo que permiten una calidad de video impecable a una resolución [35].

Respecto a la conectividad, el dispositivo se comunica vía USB 2.0. Mientras que su compatibilidad para trabajar en diversos sistemas operativos es posible gracias al controlador que trae el dispositivo, el cual funciona correctamente en distribuciones Windows, GNU/Linux y en especial es totalmente compatible con la librería de OpenCV.

Estas condiciones favorecen el funcionamiento del algoritmo de Detección de Objetos Abandonados, y en general el de sistemas basados en visión por computador que usan una cámara web para adquirir imágenes de video, como lo hace [14]. Es por esto que se escogió este dispositivo para realizar la captura de imágenes, además del módico costo que representa de acuerdo a sus beneficios.

En este proyecto la cámara es utilizada a una resolución de 640x480, tanto en la captura de tiempo real como en la captura de los videos de prueba.

5.2.2 Plataformas de Procesamiento

Por ser una aplicación de visión por computador es necesario tener un hardware de procesamiento que interprete las instrucciones del algoritmo, y permita hacer los cálculos pertinentes para interpretar las imágenes. Las plataformas con las cuales se desarrolló el sistema, son un computador portátil con un procesador Intel Core 2 Duo de arquitectura x64, y una tarjeta de procesamiento con procesador OMAP3530 de arquitectura ARM Cortex-A8. Algunas de las características de ambas plataformas se listan a continuación.

Tabla 1 Datos de las Plataformas de Procesamiento

Procesador	Intel Core 2 Duo P9600	OMAP3530DCBB72
Frecuencia de Reloj	2.66Ghz	720MHz
Memoria RWM	4,00GB @ 1333MHz	2Gb MDDR SDRAM x32 (256MB @ 166MHz)
GPU	Nvidia GeForce 9400M G	PowerVR SGX530 @ 200MHz
DSP	No tiene	TMS320C64x+@520MHz
PC	Dell Studio XPS 1340	Beagle Board - Rev C4
Sistema Operativo	Windows 7 Ultimate 64-bit	Angstrom 2010-3 Kernel 2.6.35-47
Entorno Grafico	Windows Aero	Enlightenment 16.1.0.2
Compilador	Microsoft Visual C++ 2010 Express	GCC 4.4.1 ARMv7

Ambas plataformas de procesamiento contienen el hardware de adquisición de video.

5.3 Software del Sistema

La contribución del software del sistema representa la funcionalidad del mismo, está constituido por el sistema operativo, el compilador.

El código fuente diseñó por completo sobre la distribución Windows 7 64-bit usando el compilador Microsoft Visual C++ 2010 Express, sobre el PC Dell XPS 1340. Una vez desarrollado, se implementó en la BeagleBoard, usando el compilador GCC 4.4.1 ARMv7 bajo el sistema operativo Angstrom GNU/Linux. Es posible desarrollar el mismo código portado en distintos sistemas operativos y distintos compiladores, gracias que la distribución 2.1 de la librería OpenCV es compatible tanto con los compiladores como con varios sistemas operativos.

La configuración de Software de la BeagleBoard tanto en su sistema operativo como en el compilador, y la librería OpenCV, es la misma que se utilizó en [14], para el mismo dispositivo.

5.4 Algoritmo del Sistema

Los algoritmos de detección de objetos abandonados cumplen la característica estándar de análisis de Video Vigilancia Inteligente que es, reconocer objetos, seguir su trayectoria de movimiento y finalmente realizar la operación deseada. El algoritmo propuesto, se sale de este esquema para dar paso a una nueva aproximación propuesta en [13], y que de manera similar se implementó en [12]. Estos trabajos se apartan del paradigma que establece como esencial la etapa de seguimiento de objetos en los sistemas de Video Vigilancia Inteligente. Y proponen un novedoso método basado en la estimación fondo dual adaptativo.

5.4.1 Conversiones de Color

Para desarrollar el procedimiento de detección de objetos abandonados, las imágenes originales que captura la cámara son transformadas a la entrada del sistema. Todas las imágenes son convertidas a imágenes en escala de grises, en este espacio de color se hace la construcción del fondo. Las demás imágenes en el algoritmo son binarias, algunas se obtienen de operaciones con otras imágenes del mismo tipo, pero la mayoría se obtiene tras aplicar una diferencia absoluta entre dos imágenes en escala de grises y umbralizar su resultado, el nivel de umbral depende de la etapa del algoritmo. [17][18]

Las imágenes en escala de grises son el resultado de una conversión que consiste en tomar los niveles de intensidad de las componentes R, G y B y operarlos tal como se establece en la siguiente ecuación:

$$Y = (0.299)R + (0.587)G + (0.114)B \quad ec.2$$

Las imágenes binarizadas se obtienen a partir de la siguiente ecuación:

$$imgBin = \begin{cases} 255 & imgGris > Umbral \\ 0 & otro\ caso \end{cases} \quad ec.3$$

[19]

Los parámetros de conversión seleccionados para este algoritmo se extrajeron a partir de heurísticas encontradas en [32]. Este es un tema que conlleva todo un desarrollo, pues establecer el umbral correcto depende de la aplicación, un umbral muy bajo conlleva a introducir ruido, y un umbral alto implicaría pérdida de información.

5.4.2 Diagrama en Bloques del Algoritmo

El siguiente diagrama en bloques muestra la estructura del algoritmo diseñado:



Figura 3 Diagrama de Bloques el Algoritmo Propuesto

La Figura 3 es una muestra de la estructura de las implementaciones basadas en estimación de fondo dual, adaptativo. A continuación se hace la descripción de cada bloque.

5.4.3 Actualización de Fondo

Esta etapa comprende la construcción del fondo y su actualización en el tiempo. Aquí se realiza la construcción del fondo Dual que da las herramientas para determinar lo estático de los objetos del primer plano, utilizando el método de Media Móvil previamente expuesto con brevedad en la sección 2.4.

A través de un fondo dual es fácil determinar si un objeto que se ha quedado estático, es un objeto nuevo en la escena, esta situación es de interés para detectar objetos abandonados y constituye la base sobre la cual está desarrollado el algoritmo. La forma en como esto sucede es haciendo que la tasa de aprendizaje de uno de los métodos proceda a actualizar rápidamente uno de los fondo, y lentamente el otro. De esta manera el sistema cuenta con un Fondo Actual y un Fondo Original, respectivamente. El Fondo Actual adapta en un corto tiempo los cambios de la escena, y permite registrar los cambios momentáneos o definitivos. El Fondo Original, adapta los cambios en un tiempo más grande, aquí solo se registran los cambios definitivos y gracias a esto se constituye el punto de comparación que le indica al sistema lo que había.

Las ecuaciones que en un principio definen el Fondo Actual y el Fondo Original son las siguientes:

$$FA_{[n]}(x, y) = (1 - \alpha) \cdot FA_{[n-1]}(x, y) + \alpha \cdot I_{[n]}(x, y) \quad ec. 4$$

$$FO_{[n]}(x, y) = (1 - \beta) \cdot FO_{[n-1]}(x, y) + \beta \cdot I_{[n]}(x, y) \quad ec. 5$$

α : tasa de aprendizaje rapido

β : tasa de aprendizaje lento

FA : fondo actual

FO : fondo original

I : imagen actual

Todas las imágenes en esta etapa, están en escala de grises.

La rapidez con la que se actualiza el fondo depende directamente de la tasa de aprendizaje como se hace evidente en las ecuaciones. Para escoger el valor de α y β , Ver anexo 1.

Existe una condición desfavorable en la construcción del fondo, que se da cuando la escena tiene mucho movimiento, esto se puede representar, por ejemplo, por un grupo grande de personas que pasan por el escenario durante un tiempo considerable. Esto podría causar que el fondo se llene de sombras que por supuesto no deberían encontrarse en una imagen de fondo, pues distorsiona la aproximación, al introducir ruido en la imagen.



Figura 4 (a) Fondo Actual Con Sombras, (b) Fondo Original Con Sombras

Para evitar esta situación el cálculo del fondo se limita a la cantidad de movimiento presente en la escena, de tal forma que si la escena contiene un valor de movimiento por encima del umbral M , no se efectúa el cálculo de ningún fondo y estos quedan tal y como se habían establecido previamente. La cantidad de movimiento en la escena es calculada en el Bloque de Detección de Movimiento, donde el sistema calcula la cantidad de píxeles móviles en un tiempo equivalente a 5 cuadros de video.

Por otra parte, se encontró que el Fondo original, toma demasiado tiempo en adaptar la escena al inicio de la ejecución, haciendo que se retrase el debido funcionamiento del mismo. Para evitar este retraso, se diseñó un método que consiste en asegurar que la escena se encuentra estable en cuanto al movimiento, para construir los fondos con tasas de aprendizaje rápidas, al inicio de la ejecución e ir cambiando su valor

a medida que se construye el fondo. A raíz de esto el Fondo Actual y el Fondo Original cuentan con tres tasas de aprendizaje.

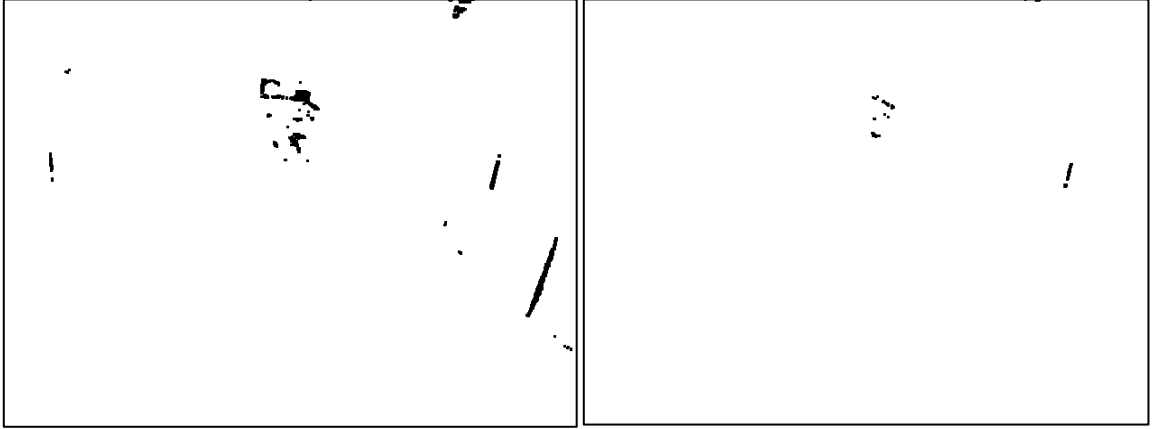


Figura 5 (a) Fondo Actual Mal Construido, (b) Fondo Original Mal Construido

Las ecuaciones de los Fondos, que incluyen estas mejoras son las siguientes:

$$FA_{[n]}(x, y) = \begin{cases} (1-\alpha) \cdot FA_{[n-1]}(x, y) + \alpha \cdot I_{[n]}(x, y) & \text{movimiento} < \text{MAXMOVIMIENTO} \\ FA_{[n-1]}(x, y) & \text{en otro caso} \end{cases} \quad \text{ec. 5}$$

$$FO_{[n]}(x, y) = \begin{cases} (1-\beta) \cdot FO_{[n-1]}(x, y) + \beta \cdot I_{[n]}(x, y) & \text{movimiento} < \text{MAXMOVIMIENTO} \\ FO_{[n-1]}(x, y) & \text{en otro caso} \end{cases} \quad \text{ec. 7}$$

$$\alpha = \begin{cases} a & \text{si } \# \text{Cuadros} < \mathcal{A} \\ b & \text{si } \mathcal{A} < \# \text{Cuadros} < \mathcal{B} \\ c & \text{si } \# \text{Cuadros} > \mathcal{B} \end{cases} \quad \text{ec. 8}$$

$$\beta = \begin{cases} a & \text{si } \# \text{Cuadros} < \mathcal{A} \\ b & \text{si } \mathcal{A} < \# \text{Cuadros} < \mathcal{B} \\ d & \text{si } \# \text{Cuadros} > \mathcal{B} \end{cases} \quad \text{ec. 9}$$

a : tasa de aprendizaje de inicio de FA y FO.

b : tasa de aprendizaje del medio de FA y FO.

c : tasa de aprendizaje Final de FA.

d : tasa de aprendizaje Final de FO.

\mathcal{A}, \mathcal{B} : Umbrales de Selección

En estas ecuaciones solo se incluyen las tasas de aprendizaje finales, ya que las demás son inherentes al sistema y solo tienen lugar al inicio del sistema.



Figura 6 (a) Fondo Actual Bien Construido, (b) Fondo Original Bien Construido

5.4.4 Bloque de Detección de Movimiento

Con el fin de determinar cuánto movimiento hay en la escena, este bloque cumple las funciones, de mapear los píxeles móviles de la escena en una imagen binaria, donde 1 equivale a los píxeles móviles y 0 a los píxeles estáticos. Y extraer un valor equivalente a los píxeles móviles que se halla calculando la norma L1 de la imagen binaria que se obtuvo.

El método aplicado consiste en tomar diferencias temporales cuadro a cuadro del video de entrada, binarizar la imagen resultante, cinco de estas muestras y unificar la información mediante una AND lógica, para obtener y así registrar movimiento en un tiempo equivalente a las muestras almacenadas.

Esta labor se ha desarrollado antes en [31], bajo el concepto *temporal template*, cuyo objetivo es justamente el de construir una representación del movimiento a través del tiempo.

Se podría pensar que con una simple diferencia temporal es suficiente para determinar la cantidad de movimiento en la escena, sin embargo esto depende de que tan grande sea la ventana de tiempo en la cual se desea obtener la representación del movimiento, y de la tasa de muestreo bajo la cual trabaja el algoritmo. La ventana de tiempo se puede describir con la siguiente ecuación:

$$V = nDif \cdot T_m \text{ ec. 10}$$

En esta ecuación $nDif$, equivale al número de muestras o diferencias y T_m , al periodo de muestreo.

La influencia en tiempo de un número de muestras igual a cinco, se muestra en la siguiente tabla:

Tabla 2 Ventana de Tiempo

$nDif = 5$	
$T_m (s)$	$V (s)$
1	5
1/5	1
1/10	0.50
1/20	0.25
1/25	0.20
1/30	0.16



Figura 7 (a) Píxeles Móviles, (b) Escena Original

Las anteriores imágenes fueron tomadas bajo el funcionamiento del algoritmo a una tasa de 20 cuadros por segundo.

5.4.5 Extracción de Primer Plano

La extracción del primer plano se hace mediante la diferencia de la imagen actual en escala de grises, con cada uno de los fondos, posteriormente las imágenes se binarizan con un umbral de 40 en el rango de 0 a 255. Así, se obtienen las imágenes del primer plano para cada fondo.

Cada una de las imágenes extraídas, se denomina según el fondo que corresponda. Por tal motivo el Primer Plano Actual (PPA), corresponde al primer plano del Fondo Actual, y el Primer Plano Original (PPO), corresponde al del Fondo Original.

Las siguientes ecuaciones representan matemáticamente las imágenes del primer plano:

$$PPA_{[n]} = I_{[n]} - FA_{[n]} \quad ec. 11$$

$$PPO_{[n]} = I_{[n]} - FO_{[n]} \quad ec. 12$$

Algunas imágenes de primer plano se muestran a continuación.



Figura 8 (a) Primer Plano Original, (b) Primer Plano Actual

5.4.6 Detección de Regiones Estáticas

Para determinar si un objeto está abandonado, previamente se debe establecer si este se ha quedado en una misma posición, para comprobar esta condición se extrae un mapa de regiones estáticas. Este mapa depende directamente del Primer Plano Actual (PPA) y del Primer Plano Original (PPO). Al comparar estas dos imágenes existen los siguientes casos:

$PPA(x, y) = 0$ y $PPO(x, y) = 0$, esto indica que los valores de los fondos es el mismo en las coordenadas (x, y) .

$PPA(x, y) = 0$ y $PPO(x, y) = 1$, esta es la condición de interés pues indica se trata de un pixel que es parte del fondo actual pero no del fondo original, lo cual indica que es algo nuevo, que no estaba antes. También puede indicar algo que estaba previamente y ya no está.

$PPA(x, y) = 1$ y $PPO(x, y) = 0$, esta condición no es de interés pues de la única forma en que se da es si existe una mala aproximación de alguno o ambos fondos, en una operación adecuada esta condición nunca se cumple.

$PPA(x, y) = 1$ y $PPO(x, y) = 1$, finalmente un 1 lógico en ambas imágenes, representa un objeto nuevo que entra en la escena que no ha sido adaptado en ningún fondo.

El mapa estático se construye desde una imagen nula e incrementa los valores de los pixeles progresivamente bajo la condición $PPA(x, y) = 0$ y $PPO(x, y) = 1$, hasta que llegan al máximo valor, es decir 255. Cuando se deja de cumplir esta condición, una región estática deja de serlo, los valores de los pixeles disminuyen su valor hasta llegar al valor mínimo es decir 0.

A continuación, la ecuación que describe el mapa de ecuaciones estáticas:

$$ME_{[n]}(x, y) = \begin{cases} ME_{[n-1]}(x, y) + \Delta t & PPA(x, y) = 0 \wedge PPO(x, y) = 1 \\ 255 & PPA(x, y) = 0 \wedge PPO(x, y) = 1 \wedge ME_{[n-1]} > 254 \\ ME_{[n-1]}(x, y) - 1 & PPA(x, y) \neq 0 \wedge PPO(x, y) \neq 1 \\ 0 & PPA(x, y) \neq 0 \wedge PPO(x, y) \neq 1 \wedge ME_{[n-1]} < 1 \end{cases} \quad ec. 13$$

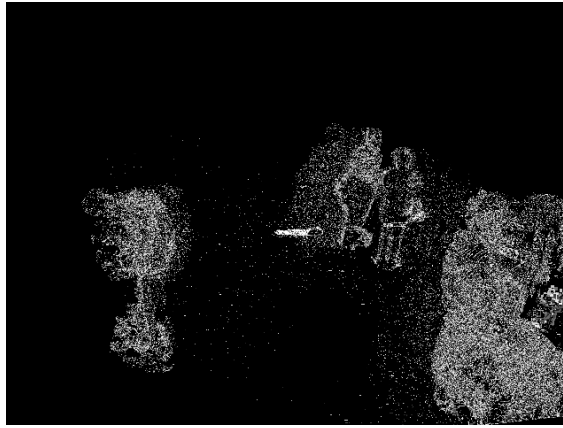


Figura 9 Mapa de Regiones Estáticas

5.4.7 Detección de Objetos Abandonados

Finalmente, en esta etapa se obtienen las coordenadas en las cuales habita un objeto abandonado. Esto sucede luego de obtener el mapa de regiones estáticas, los lugares donde este mapa haya tomado su máximo valor serán las regiones en donde habita un objeto abandonado, pues habrá pasado el tiempo suficiente para determinar que el objeto que se ha quedado en una misma posición es un objeto abandonado. Estos valores se almacenan en una nueva imagen denominada Mapa de Abandonados.

Una vez se han ubicado las coordenadas correspondientes, se hace un mapeo en la imagen original de captura, llevando la intensidad del canal R a su máximo valor y generando así la señal de alarma. De esta forma los objetos en la escena que estén cubiertos por una tonalidad de rojo intenso, serán los objetos abandonados, sobre los cuales el operario deberá llevar su atención.

La siguiente ecuación describe las señales de imagen, de la detección de objetos abandonados:

$$R(I_{[n]}(x, y)) = \begin{cases} 255 & ME_{[n]}(x, y) = 255 \\ R(I_{[n-1]}(x, y)) & ME_{[n]}(x, y) \neq 255 \end{cases} \quad ec. 14$$

$R(I_{[n]}(x, y))$: canal rojo de la imagen de entrada

$MA_{[n]}(x, y)$: mapa de abandonados

Las siguientes imágenes muestran objetos abandonados señalados por el algoritmo:



Figura 10 (a) Detección de Objetos Abandonados 1, (b) Detección de Objetos Abandonados 2

Como se logra ver en la Figura 10 el sistema no solo realiza la detección de objetos abandonados sino que por el modo en el que se efectúa esa detección, el algoritmo también detecta los objetos removidos, esto es, debido a que lo que detectar un objeto que no estaba, y en un momento determinado si, es equivalente a detectar algo que ya estaba, pero ha dejado de pertenecer a ese espacio.

5.4.8 Filtrado de Ruido

En la última etapa se hace un filtrado a la imagen de detección de objetos abandonados, con el fin de eliminar el ruido que puede generarse tras la etapa de detección de regiones estáticas. El ruido se manifiesta como ruido similar al sal y pimienta, el cual se suele eliminar con el uso de técnicas como erosión y dilatación una seguida de otra, o una contenida en otra como ocurre con la técnica de apertura y cierre. [15]

El filtro que se implementó se fundamenta en aplicar dos iteraciones de erosión y en seguida dos de dilatación. A pesar que no es directamente una operación de apertura si es muy similar, lo único en lo que difiere es el número de iteraciones de Erosión y Dilatación.

El elemento estructurante utilizado para realizar ambas operaciones es:

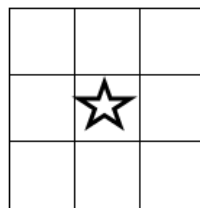


Figura 11 Elemento Estructurante

En la operación de dilatación, el punto central del elemento estructurante se ubica alrededor de la frontera de todas las islas presentes y calcula el máximo local en esa área haciendo que las islas se expandan. Lo contrario sucede en la operación de erosión, haciendo uso del mismo operador, la erosión calcula el mínimo local, logrando reducir el tamaño de las islas.

Una explicación más profunda de los operadores utilizados se presenta en [19].

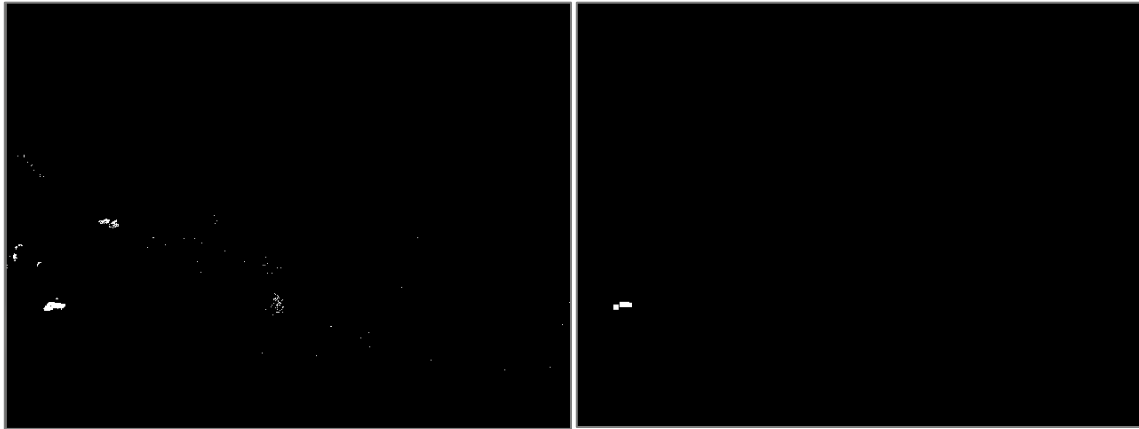


Figura 12 (a) Sin Filtrado, (b) Con Filtrado

Por facilidad para ver con detalle la acción del filtrado se las respectivas muestras de las imágenes a color, en imágenes binarias.

5.5 Bases de Datos

Con el fin de probar el desempeño del algoritmo propuesto, se construyó una base de datos de objetos abandonados, así mismo se consultaron dos de las bases de datos más usadas en esta temática.

5.5.1 VDOA

Con motivo de desarrollar y probar el algoritmo se tomaron 8 videos en diferentes instalaciones de la Pontificia Universidad Javeriana, en escenarios propicios para el análisis de resultados del algoritmo. La captura de video se realizó con la cámara PlayStation Eye, bajo una resolución de 640x480, y una tasa de muestreo de 30 cuadros por segundo, de los cuales 2 se utilizaron en la etapa de desarrollo, y los 6 restantes se utilizaron para probar el algoritmo.

De esta forma se construyó la base de datos VDOA (Videos de Detección de Objetos Abandonados), útil para establecer comparaciones con desarrollos posteriores en esta temática.

La siguiente tabla lista los videos de VDOA:

Tabla 3 Base de Datos VDOA

Videos de Detección de Objetos Abandonados (VDOA)			
Nombre	Descripción	Tipo de Espacio	Uso
Video de Prueba 1	Zona Baños	Exterior	Prueba
Video de Prueba 2	Zona de ascensores	Interior	Prueba
Video de Prueba 3	Pasillo CJFD	Interior	Diseño
Video de Prueba 4	Pasillo Biblioteca	Interior	Prueba
Video de Prueba 5	Pasillo edificio F. Barón	Interior	Diseño
Video de Prueba 6	Zona de estantería de libros	Interior	Prueba
Video de Prueba 7	Zona de escaleras en biblioteca	Interior	Prueba
Video de Prueba 8	2 Zona de escaleras en biblioteca	Interior	Prueba

5.5.2 Otras Bases de Datos

Para realizar comparaciones de los resultados obtenidos se consultaron las bases de datos AVSS 2007[40] y CAVIAR [41]. Las siguientes tablas listan los videos de estas bases de datos usados para probar el algoritmo.

Tabla 4 Base de Datos AVSS 2007

AVSS 2007
AVSS_AB_Easy_Divx
AVSS_AB_Medium_Divx
AVSS_AB_Hard_Divx

Tabla 5 Base de Datos CAVIAR

CAVIAR
LeftBag
LeftBag_AtChair
LeftBag_BehindChair
LeftBag_PickedUp
LeftBox

“AVSS 2007 ofrece datos de referencia a disposición del público para probar y evaluar algoritmos de detección y seguimiento. Los conjuntos de datos son gratuitas para los fines de investigación y educativos únicamente y pueden ser utilizados en las publicaciones científicas en la condición de respetar el reconocimiento de la citación solicitada”. Los videos de AVSS 2007 que se utilizaron en las pruebas son los correspondientes a *i-Lids bag and vehicle detection challenge* [40]. Las características de los videos se pueden consultar en la misma referencia.

“El objetivo principal de CAVIAR es hacer frente a la pregunta científica: ¿Puede una imagen local rica en descripciones foveal y otros sensores, seleccionados por un proceso jerárquico de atención visual y guiadas y procesadas con la tarea, la escena, la función y el conocimiento contextual del objeto mejorarlos procesos de reconocimiento basados en imágenes? Esto hacer frente a cuestiones claramente fundamentales para el enfoque de la visión cognitiva”. [42]

Estos videos se incluyen en el DVD anexo para su rápida supervisión y facilitar su acceso, pero su uso por parte de otros trabajos se encuentra limitado por las licencias específicas de cada uno, es recomendable entrar a la página web de cada proveedor para conocer más de las licencias que tienen estos videos para su uso.

5.6 Configuración del sistema

Este algoritmo de detección de objetos abandonados, es adaptable a la naturaleza del tráfico peatonal en la escena. Para poder adaptar el sistema es preciso analizar algunas características de la escena, y de acuerdo con estas, escoger los valores adecuados de cada parámetro.

5.6.1 Análisis de la Escena

Existen principalmente dos características que el usuario puede modificar en el sistema. La primera es el tiempo que de espera de un objeto nuevo en la escena para detectar un objeto en abandono. La segunda es el tiempo que debe transcurrir para que el sistema adapte completamente, los cambios en la escena.

El usuario debe determinar el tiempo máximo que en condiciones normales dura una persona en una posición de espera, por ejemplo, en una zona de ascensores de 5 pisos el tiempo máximo de espera por un ascensor es de 2 min. Este tiempo en adelante se denominara como *Tiempo de Espera*. Por encima de este valor, todo lo que se encuentre estático deberá detectarse como un objeto abandonado.

También es preciso determinar el tiempo que debe transcurrir para adaptar un cambio definitivo de la escena. Este tiempo será llamado *Tiempo de Adaptación*. Se debe tener en cuenta su valor debe ser

considerablemente mayor al del tiempo de espera, con el fin de dar el tiempo suficiente para que el operario se dé cuenta de las detecciones. El tiempo que dura la señal de alarma sobre los objetos abandonados es equivalente a la diferencia entre el *Tiempo de Adaptación* y el *Tiempo de Espera* y será denominado, *Tiempo de Detección*.

Así pues, el usuario debe analizar la escena y extraer los valores de tiempo, que den una mejor aproximación de la naturaleza del sistema.

5.6.2 Escogencia de Parámetros

El *Tiempo de Espera* depende directamente de dos parámetros.

El primero corresponde la tasa de aprendizaje α del FA. Entre menor es el valor de este parámetro más lenta es la actualización del FA, retrasando a su vez, que se cumpla la condición con la que se obtiene el mapa de regiones estáticas, y por tanto las coordenadas de los objetos abandonados.

Para determinar el valor adecuado de α es necesario consultar las gráficas que muestran los valores de α respecto al tiempo de adaptación, que se encuentran en el Anexo 1.

El segundo corresponde a γ , un parámetro proporcional al tiempo que se tarda el sistema en llegar al máximo valor en el mapa de regiones estáticas, el cual se explica a continuación.

Los incrementos del valor de los pixeles de cada iteración en la construcción del mapa estático son proporcionales al tiempo que un objeto estático tarda en pertenecer al mapa de regiones estáticas. Esta característica puede ser manipulada para variar el tiempo de detección de regiones estáticas pero también de objetos abandonados. Este parámetro es útil para la caracterización del algoritmo. Con el fin de ser utilizado como tal el algoritmo incorpora esta característica y la Tabla 5 muestra cómo se puede variar este tiempo.

Tabla 6 Factor de Escala y Periodo de Muestreo

T_m (s)	Δt (s)
1	$1 \cdot \gamma$
1/5	$5 \cdot \gamma$
1/10	$0.50 \cdot \gamma$
1/20	$0.25 \cdot \gamma$
1/25	$0.20 \cdot \gamma$
1/30	$0.16 \cdot \gamma$

γ : el factor de escala escogido por el usuario para un incremento Δt .

$$\gamma = 0,1,2,3 \dots$$

El tiempo total que representa cada incremento escogido equivale a:

$$T_{total} = 255/\Delta t \quad ec. 15$$

Este es el tiempo total que se demora una región estática en tomar su máximo valor.

El *Tiempo de Adaptación* solo depende del parámetro β , del FO. Su valor cumple exactamente la misma función que el parámetro α y para seleccionar su valor también es necesario remitirse al Anexo1.

5.7 Carga Computacional

La carga computacional es una cualidad del algoritmo necesaria para conocer su eficiencia. Su medida se hace mediante la estimación de la tasa de muestreo con la cual se procesa el algoritmo.

Para hallar este valor se mide el tiempo que tarda el algoritmo en procesar las imágenes en cada iteración del proceso, el inverso de este número proporciona la frecuencia de muestreo. Este procedimiento se desarrolló previamente en [14].

Esta medida es útil para realizar los cálculos de γ y la ventana temporal V (s).

Cabe resaltar que estos valores dependen de la implementación en software, del hardware y del uso de los recursos que el hardware en específico dispone. Por lo que estos valores fueron hallados bajo la misma implementación de algoritmo propuesto en este libro, con las opciones por defecto de compilación de OpenCV y de mejoras del compilador, sin hacer uso de hilos de proceso, usando solamente el procesador principal de las plataformas de prueba. La razón principal de obtener estos datos consiste en la calibración de los tiempos de operación del sistema, en otras palabras la obtención de la frecuencia de muestreo temporal, necesaria para fijar los demás parámetros de operación del sistema.

Los resultados en el XPS 1340 y en la BeagleBoard fueron:

Tabla 7 Velocidad De Procesamiento

PC	XPS 1340	BeagleBoard
Cuadros por Segundo	25	5

6 ANÁLISIS DE RESULTADOS:

En las primeras tres secciones de este capítulo se encontrara una serie de pruebas del algoritmo.

6.1 Pruebas con videos de archivo

En los apartados de esta sección se encontrara el resultado de las pruebas, obtenido con los videos VDOA destinados para este fin, y los obtenidos con AVSS 2007 y CAVIAR. Los valores de $a, b, c, d, \gamma, \mathcal{A}$ y \mathcal{B} para VDOA fueron los que se hallaron en la fase de desarrollo, con los videos dedicados para ello, por otro lado los que se usaron en los videos de AVSS 2007 y CAVIAR, fueron hallados estudiando el comportamiento de la escena, y posteriormente siguiendo las especificaciones del apartado 6.4.2.

Las características que será observadas en cada prueba serán, el número de objetos abandonados y de falsos positivos. De la misma forma que se hace en [13].

6.1.1 VDOA.

Se realizaron algunas pruebas ajustando los parámetros del sistema de la siguiente forma:

$$a = 0.05, b = 0.002, c = 0.002, d = 0.0003, \gamma = 4, \mathcal{A} = 200, \mathcal{B} = 800 \quad \text{ec. 16}$$

Estos parámetros corresponden a la caracterización a la que se llegó durante la etapa de diseño con los videos de la base de datos que se destinaron para esto (ver gráficas de Anexo 1).

En algunos casos no se detecta remoción por terminación del Video, sin embargo se asume que su remoción es efectiva, por haber existido una detección.

Video de Prueba 1:



Figura 13 (a) Objeto Abandonado VP1,(b) Ruido VP1

En la Figura 13 (a) se señala la primera detección, en la se abandona el segundo objeto, la Figura 13 (b) se señala también otra área, sin embargo hace parte del interior de la estructura y el interés está centrado en el exterior de la misma.



Figura 14 Objeto Removido

Finalmente no se reconoce el segundo objeto abandonado, en 6.3 se explicará, por qué.

Tabla 8 Resultados Video de Prueba 1

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Video de Prueba 1	Abandonado	2	1	0
	Removido	2	1	0

Video de prueba 4:



Figura 15 (a) Objeto Abandonado A VP4, (b) Objeto Abandonado B VP4



Figura 16 (a) Objetos Removidos A y B VP4, (b) Objetos Abandonado C y D y Removido A VP4

Tabla 9 Resultados Video de Prueba 4

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Video de Prueba 4	Abandonado	4	4	0
	Removido	4	4	0

Video de prueba 6:



Figura 17 (a) Objeto Abandonado A, (b) Objetos Abandonados A y B VP6



Figura 18 (a) Removido A VP6, (b) Objetos Abandonado C y Removidos A y B VP6

Tabla 10 Video de Prueba 6

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Video de Prueba 6	Abandonado	4	4	0
	Removido	4	4	0

Video de prueba 7:



Figura 19 (a) Objeto Abandonado A VP7, (b) Objeto Abandonado B VP7

Tabla 11 Video de Prueba 7

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Video de Prueba 7	Abandonado	2	2	0

En este video, sucede un cambio de escena el cual analizará más adelante en 5.4, los resultados corresponden a los obtenidos antes del mismo.

6.1.2 AVSS2007

Video AVSS_AB_Easy_Divx

$$a = 0.05, b = 0.001, c = 0.001, d = 0.0003, \gamma = 4, \mathcal{A} = 200, \mathcal{B} = 800 \quad ec. 17$$

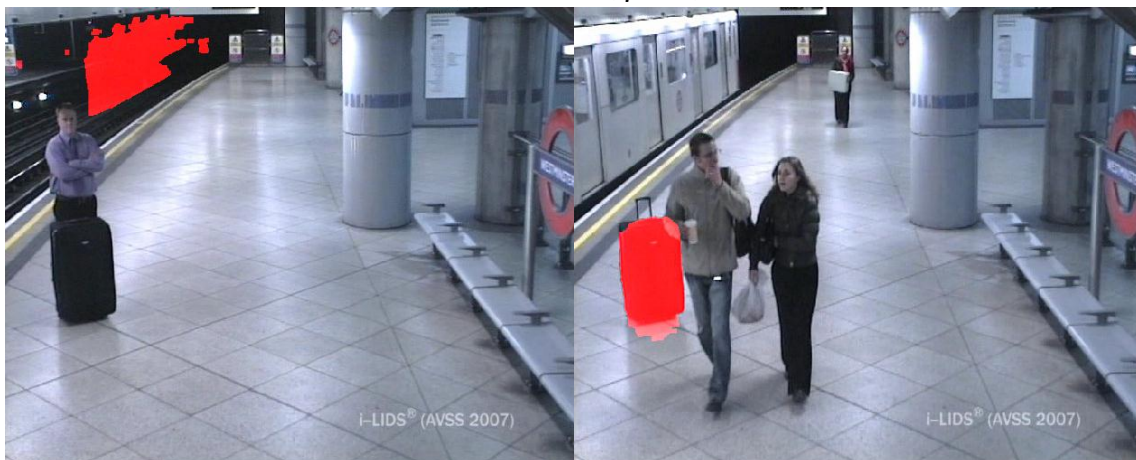


Figura 20 (a) Abandono de Maleta, (b) Detección del Objeto

Como se logra ver en la Figura 20 (a) existe una falsa detección en la zona del tren esta zona no se tiene en cuenta en los falsos positivos pues tal zona no es de interés para la detección de objetos en abandono. Por su parte la Figura 20 (b) muestra la detección del objeto en abandono.

Tabla 12 Video AVSS_AB_Easy_Divx

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
AVSS_AB_Easy_Divx	Abandonado	1	1	0

Video AVSS_AB_Medium_Divx

$a = 0.05, b = 0.002, c = 0.0006, d = 0.0003, \gamma = 4$ ec. 18



Figura 21 (a) Escena de Inicio, (b) Inicio de Detección

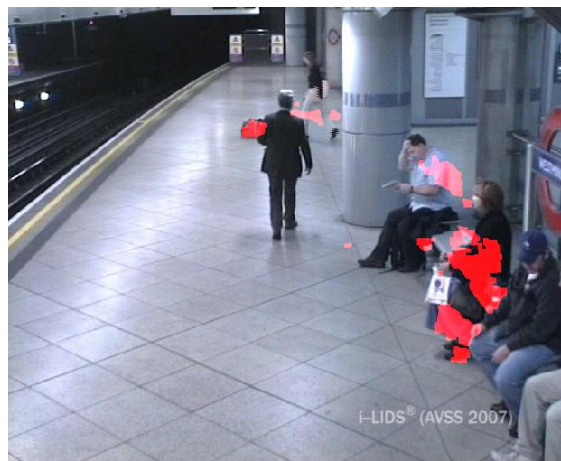


Figura 22 Objeto Detectado

En el segundo video de AVSS 2007 se hace la detección del objeto sin problema, sin embargo la escena muestra también algunas detecciones que no deberían estar allí, esto se da porque la primera escena mostrada en la Figura 21(a) dura alrededor de un minuto y medio, debido a que el algoritmo construye un fondo a partir de las muestras que toma en el tiempo, constituyendo el fondo en esta imagen, lo cual causa que ante la ausencia de personas y objetos pertenecientes al FO, se detecte su ausencia como si fuese un objeto removido. Esto es evidente al comparar la Figura 21 (a) y la Figura 21(b), en especial donde se encuentra señalado.

Tabla 13 Video AVSS_AB_Medium_Divx

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
AVSS_AB_Medium_Divx	Abandonado	1	1	---

Video AVSS_AB_Hard_Divx

$a = 0.05, b = 0.002, c = 0.0004, d = 0.0003, \gamma = 4, \mathcal{A} = 200, \mathcal{B} = 800$ ec. 19



Figura 23 (a) Abandono y Peatón en Cuestión, (b) Detección del Objeto y del Peatón

En este video sucede algo parecido al anterior, pero además de eso, entre las características inherentes, el *Tiempo de Espera* y el *Tiempo de Adaptación*, son de 3:30 y 3:10 respectivamente lo cual implica un rango muy estrecho en el cual hacer la detección, debido a esto, el peatón que se encuentra sentado en la Figura 23 (b), es detectado como un objeto abandonado. En la Figura 23(a) se logra comprobar que tanto el objeto de estudio como el peatón ingresan a la escena al tiempo. Se podría pensar en incrementar un poco el *Tiempo de Espera*, pero esto resultaría en prolongar no solo la detección del peatón, sino también la del objeto. Esta decisión podría ser viable o no, todo depende de lo que el operario desee que el sistema informe. De acuerdo a las condiciones de funcionamiento del sistema el peatón y sus objetos son una detección efectiva.

Tabla 14 Video AVSS_AB_Hard_Divx

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
AVSS_AB_Hard_Divx	Abandonado	2	2	0

6.1.3 CAVIAR

La característica de los Videos de esta base de datos es que todas las acciones suceden muy rápido, el *Tiempo de Espera* debe ser muy corto pues lo máximo que duran los objetos en una misma posición es 17 segundos y lo mínimo es 12 segundos, en este punto además de modificar, α, β , y γ , se deben modificar las *tasas de aprendizaje de inicio y del medio*. Además de lo mencionado los videos tienen una menor resolución, y por tanto el costo computacional es menor. Con una resolución de 384x288 la frecuencia de muestreo fue de 80 cuadros por segundo frente a 640x480 a 20 cuadros por segundo. Para probar esta base de datos fue necesario configurar el sistema de tal forma que se pudiesen hallar las constantes, esto consiste en dar un tiempo de espera a la ejecución del código, de tal forma que la tasa de muestreo sea equivalente a 25 cuadros por segundo.

$a = 0.1, b = 0.06, c = 0.04, d = 0.0003, \gamma = 5, \mathcal{A} = 200, \mathcal{B} = 300$ ec. 20

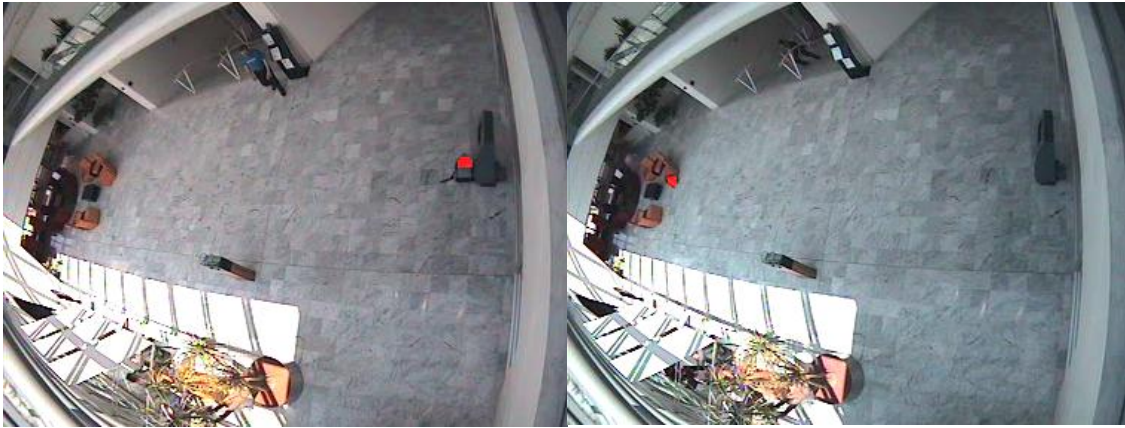


Figura 24 (a) Left Bag, (b) Left Bag at Chair



Figura 25 (a) Left Bag Picked Up, (b) Left Box

Tabla 15 Resultados CAVIAR

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
LeftBag	Abandonado	1	1	0
LeftBag_AtChair	Abandonado	1	1	0
LeftBag_PickedUp	Abandonado	1	1	1
LeftBox	Abandonado	1	1	0

6.2 Pruebas sobre la BeagleBoard

Las pruebas sobre este dispositivo se realizaron directamente con la captura de la cámara PlayStation Eye, para la correcta ejecución del programa se midió la tasa de muestreo, la cual se establece en aproximadamente 5 cuadros por segundo, con esta información se escogieron los valores de a, b, c, d, γ de tal forma que su funcionamiento fuese similar al establecido en la etapa de desarrollo, su elección consistió en buscar los valores de estos parámetros, más aproximados en tiempo, para la tasa de muestreo de 5 cuadros por segundo. Ya que la tasa de muestreo del sistema sobre el PC XPS1340, en la etapa de desarrollo es de 25 cuadros por segundo. Por su parte los valores de \mathcal{A} y \mathcal{B} , se dejaron igual a los de la etapa de desarrollo.

$$a = 0.2, b = 0.008, c = 0.008, d = 0.001, \gamma =, \mathcal{A} = 200, \mathcal{B} = 800 \quad ec. 20$$

Primer Captura Beagle Board:



Figura 26 (a) Imagen Inicial, (b) Primeras Detecciones



Figura 27 Detección de una Caja Blanca

La secuencia de imágenes revela la detección de dos objetos abandonados en la Figura 26 (b), de los cuales, uno ya pertenecía al fondo (flecha naranja) y ahora se detecta como removido en la misma imagen (flecha verde).

Tabla 16 Resultados Primera Captura Beagle Board

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Primer Captura Beagle Board:	Abandonado	3	3	0
	Removido	1	1	0

Segunda Captura Beagle Board:



Figura 28 (a) Imagen Inicial, (b) Detección de una Caja Blanca

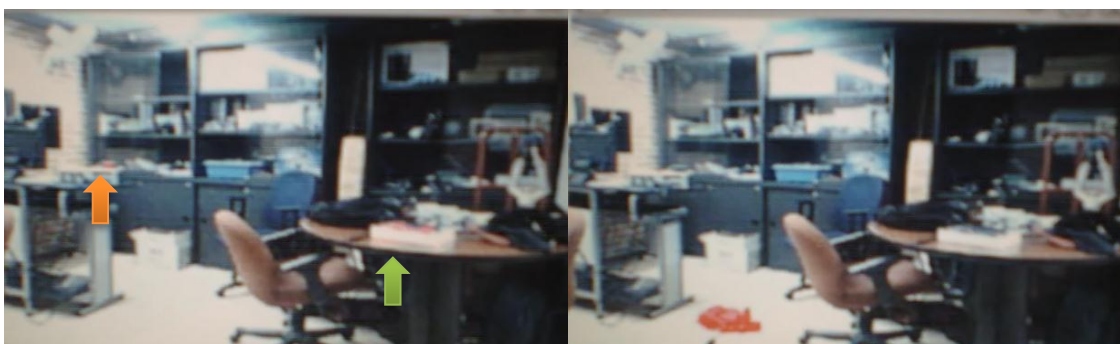


Figura 29 (a) Detección de un Libro y un Celular, (b) Detección de una Sombrilla

En la Figura 29 se detectan dos objetos abandonados, de izquierda a derecha, un celular de dimensiones 90.0x52.0x17.0 mm (flecha naranja) y un libro (flecha verde).

Tabla 17 Segunda Captura Sobre la Beagle Board

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
Segunda Captura Beagle Board:	Abandonado	3	3	0

A pesar que el sistema funciona a una tasa de muestreo menor que con la arquitectura Intel Core 2 Duo, su obtuvieron resultados que cumplen las condiciones de diseño tal como lo hace esa arquitectura.

6.3 Condiciones de Iluminación

Las condiciones de iluminación presentes en la escena son una parte fundamental en la detección de objetos en abandono, esta propiedad espacial se hace evidente en los siguientes resultados.

6.3.1 Exteriores

Este video, es el único que se tomó en un exterior, con el fin de probar la validez del algoritmo en exteriores, habiéndose diseñado para interiores.

Video de prueba 1



Figura 30 (a) Abandono Objeto 1 Minuto 2:26, (b) Detección Objeto 1 Minuto 2:55

En el primer abandono de este video se detecta el objeto abandonado después de 30 segundos. Sin embargo su detección no es tan clara ya que como se ve en la Figura 30, la región roja es parcial sobre el objeto.



Figura 31 (a) Abandono Objeto 2 Minuto 8:11, (b) Sin detección Objeto 1 Minuto 8:43

No se detecta el segundo objeto abandonado, pese a que el algoritmo reconoce la ausencia del primer objeto abandonado después de 30 segundos.

6.3.2 Interiores

Los siguientes videos se tomaron exclusivamente en interiores.

Video de prueba 8



Figura 32 (a) Abandono Objeto 1 Minuto 3:35, (b) Detección Objeto 1 Minuto 3:59

Como se ve en la Figura 32 (b), el algoritmo logra hacer una detección parcial en la parte inferior del objeto, esto se debe a que el color que existe en ese lugar antes de poner del abandono es un tono gris medio, mientras que en el resto de la ubicación del objeto es gris oscuro proporcionado por la sombra de la esquina.

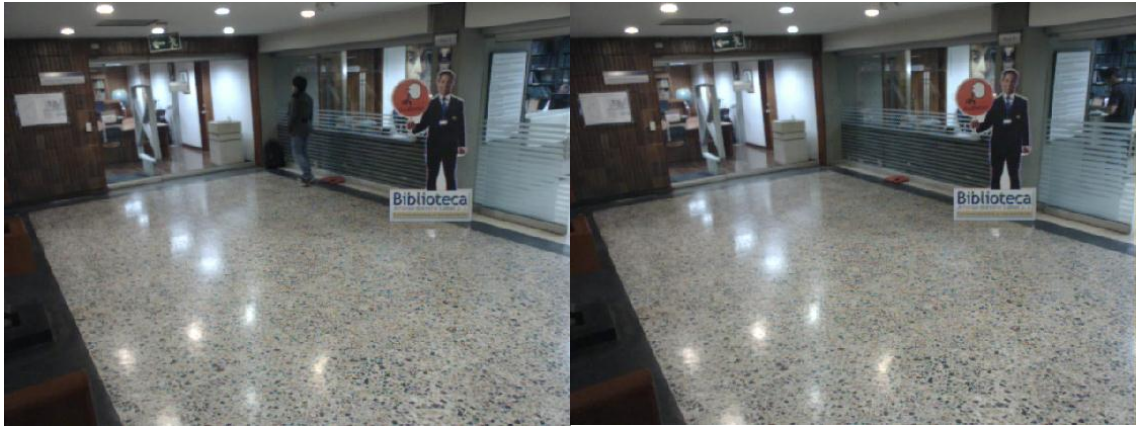


Figura 33 (a) Abandono Objeto 2 Minuto 7:35, (b) No detección Objeto 1 Minuto 9:50

Es esta ocasión el sistema tiene un objeto rojo que no se detecta, esto es debido a la misma razón del caso anterior, el nivel de intensidad de color entre lo que hay antes del objeto abandonado y cuando este se encuentra presente, no es muy distinto lo que ocasiona que se pase por alto el objeto.

A través de los anteriores resultados se comprueba el buen funcionamiento del algoritmo bajo una condición de buena iluminación, así como los posibles falsos negativos que son ignorados gracias a la ausencia de luz o la pérdida parcial del área del objeto abandonado.

Video de prueba 6:



Figura 34 Justo Antes de Abandonar Objeto 3



Figura 35 (a) Abandono Objeto 3 Minuto6:23, (b) Objeto No Detectado

En este caso el objeto abandonado 3 señalado en la Figura 35 (b), no es identificado como abandonado, cabe resaltar que es el único de 4 objetos abandonados, pues los 3 restantes fueron detectados de forma exitosa. Como en los anteriores casos, si se comparan la Figura 35 (b) y la Figura 34 Justo Antes de Abandonar Objeto 3 el espacio donde está abandonado el objeto no cambia de manera drástica en la intensidad del color, causando esta pérdida de información.

6.4 Cambio de Escena

Esta es una condición de interés en el caso de algún movimiento accidental sobre la cámara, o de un cambio total en la escena. En el siguiente video se muestra cual es el impacto de variar la posición de la cámara, y cómo responde el sistema ante esta situación.

Cabe resaltar que el algoritmo fue diseñado para operar con una cámara fija, y esto suponía una limitante de operación, sin embargo el algoritmo se adapta con el tiempo a estas situaciones como se muestra en la Figura 36 y en la Figura 37.

Video de prueba 7



Figura 36 (a) Posición Original De La Cámara Minuto 6:47, (b) Cambio De Posición De La Cámara Minuto 6:51



Figura 37 (a) Respuesta Del Sistema Minuto 7:49, (b) Adaptación Completa De La Escena Minuto 11:33

Como se muestra en las imágenes el sistema es capaz de recuperar el funcionamiento normal, después de cumplirse el *Tiempo de Adaptación*, que es el tiempo necesario para que el fondo original se construya completamente. Esto es una ventaja pues en el caso en el que suceda un cambio de este tipo no es necesario reiniciar ni interceder para que el algoritmo retome el funcionamiento natural.

6.5 Objetos Pequeños

Una de las dudas que surge es que tan grande debe ser un objeto para ser reconocido. En uno de los videos se probó si el algoritmo detectaba un artefacto de dimensiones 90.0x52.0x17.0 mm, a una distancia de 4 metros aproximadamente. Los resultados obtenidos son los siguientes:

Video de prueba 7



Figura 38 (a) Abandono Objeto Minuto 6:2, (b) Detección Objeto Minuto 6:47

Como muestran la Figura 38 (a), la Figura 38 (b), y la flecha color naranja, el objeto localizado en la escultura que es apenas visible para un operario, fue detectado con éxito por el algoritmo. La pigmentación del objeto es totalmente nítida y cubre el objeto completamente, el hecho que destaca la buena detección es el contraste entre el objeto y el fondo el cual permite distinguirlo fácilmente. El área del objeto en la imagen es de aproximadamente, 15x13 pixeles. Por tanto se puede inferir que el algoritmo es capaz de detectar objetos abandonados cuya área en la imagen sea mayor o igual a 195 pixeles cuadrados.

6.6 Condiciones Tiempo

La naturaleza del espacio que se analiza juega un papel fundamental, en el presente video, se hace evidente que no todos los espacios tienen las mismas características de tráfico peatonal, y que por esta razón el algoritmo puede funcionar de forma adecuada en algunos espacios, pero en otros no tan bien.

Video de prueba 2



Figura 39 (a) Abandono Objeto 1 Minuto 2:58, (b) Detección Objeto 1 Minuto 3:28

La primera detección se hace sin ningún problema ni contratiempo.



Figura 40 (a) Abandono Objeto 2 Minuto 5:56, (b) Detección Objeto 2 Minuto 6:38

La segunda detección también se hace sin ningún problema ni contratiempo. Se presenta un poco de ruido en la imagen.

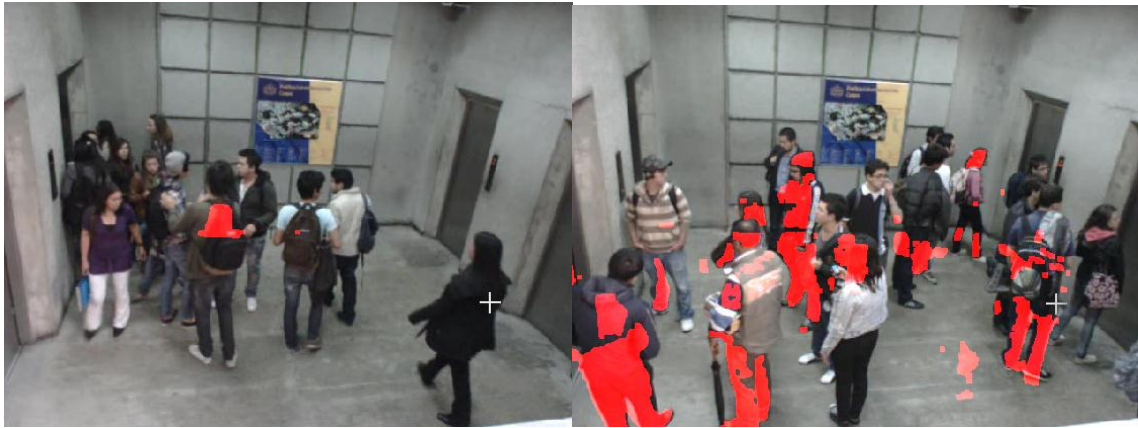


Figura 41 (a) Abandono Objeto Minuto 7:18, (b) Detección Objeto Minuto 10:04

En la Figura 41(a), pese a que existe un alto tráfico peatonal la detección del objeto sigue mostrándose, lo cual comprueba la robustez del sistema frente a obstrucciones. En la Figura 41(b), el algoritmo reconoce como objetos abandonados a diferentes individuos o parte de estos como objetos abandonados, este hecho destaca la importancia de analizar el espacio vigilado, para seleccionar de manera adecuada los valores de *Tiempo de Espera* y *Tiempo de Adaptación*. En este caso lo que sucede es que el *Tiempo de Espera* es muy corto. El tiempo de espera debe ser superior al tiempo máximo en promedio que esperan los peatones por un ascensor.



Figura 42 (a) Abandono Objeto 3 Minuto 11:00, (b) Detección Objeto 3 Minuto 11:36

En la última detección al igual que en la segunda el sistema reconoce el objeto con un poco de ruido en la imagen.

6.7 Abandoned Object Detection de Matlab

Abandoned Object Detection es un demo de Toolbox de Visión por Computador de Matlab [10] [11], es la única implementación que se encontró con la cual es posible comparar el desempeño del algoritmo desarrollado.

Los videos con los cuales se probó esta implementación son los de la base de datos CAVIAR, proveen características similares al video de prueba original de Abandoned Object Detection. Que principalmente se basa en detecciones en muy corto tiempo.

LeftBag

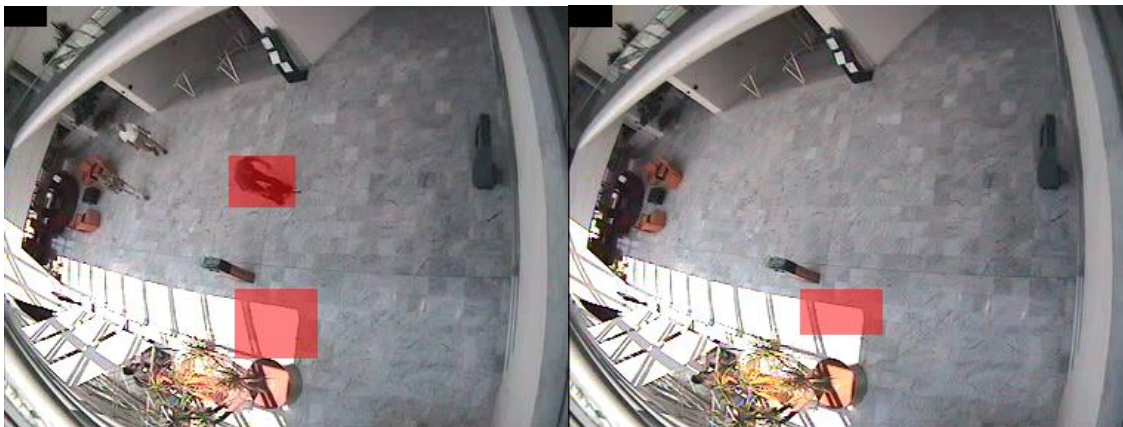


Figura 43 (a) Primeros Dos Falsos Positivos, (b) Tercer Falso Positivo

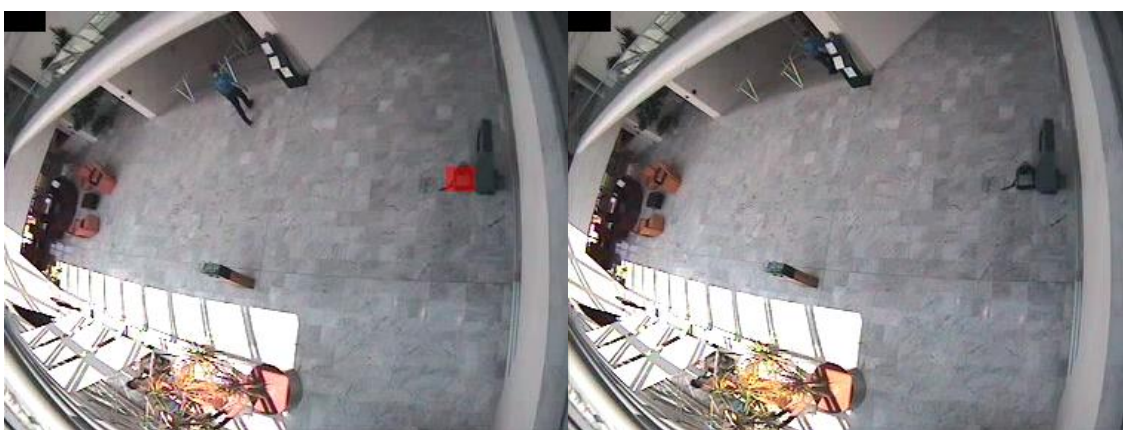


Figura 44 (a) Detección del Objeto, (b) Detección Intermitente

LeftBag_AtChair



Figura 45 Sin Detección del Objeto

LeftBag_PickedUp



Figura 46 (a) Primer Falso Positivo, (b) Segundo Falso Positivo y Detección del Objeto

LeftBox

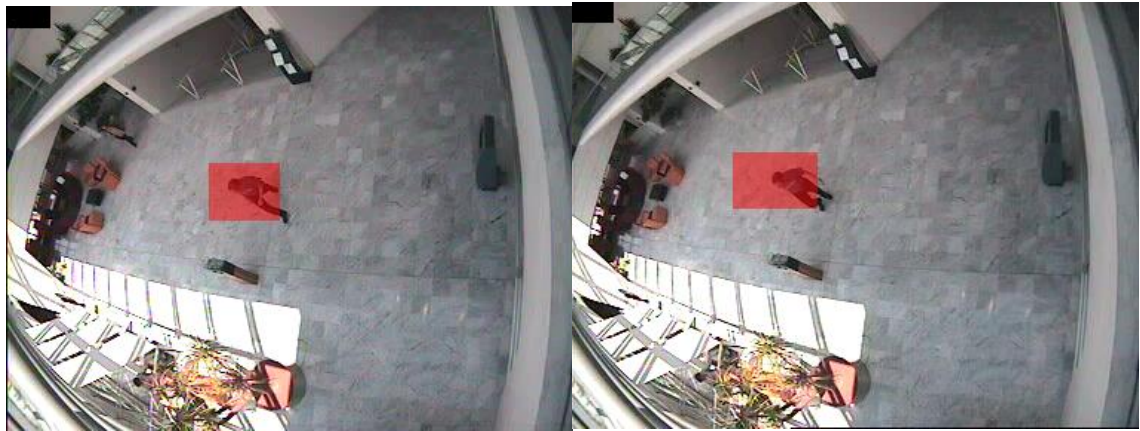


Figura 47 (a) Primer Falso Positivo, (b) Segundo Falso Positivo



Figura 48 Detección del Objeto

Con los videos de CAVIAR, se lograba ver que la implementación Abandoned Object Detection no permite oclusiones en la detección, ya que en el momento en que algún peatón intentaba cruzar por una región detectada, esta dejaba de ser una región de detección.

La Tabla 18 muestra que el desempeño del algoritmo desarrollado cuenta con menor cantidad de falsos positivos.

Tabla 18 Comparativa de Desempeño

Secuencia de Prueba		Numero de Objetos	Detecciones Acertadas	Falsos Positivos
LeftBag	Detección de Objetos Abandonados	1	1	0
	Abandoned Object Detection	1	1	2
LeftBag_AtChair	Detección de Objetos Abandonados	1	1	0
	Abandoned Object Detection	1	1	1
LeftBag_PickedUp	Detección de Objetos Abandonados	1	1	1
	Abandoned Object Detection	1	1	2
LeftBox	Detección de Objetos Abandonados	1	1	0
	Abandoned Object Detection	1	0	2

7 CONCLUSIONES:

En los siguientes apartados se resaltarán los cumplimientos del proyecto y el desenlace de la investigación. También las recomendaciones consideradas como primordiales para continuar con la temática.

7.1 Conclusiones

- El sistema de detección de objetos Abandonados, se ha diseñado, implementado, probado y comparado. La base del desarrollo del algoritmo consistió en la aplicación de conceptos propios de visión por computador, comprobando que los sistemas basados en este campo traen consigo la capacidad de automatizar procesos en áreas como seguridad, medicina, entretenimiento, aplicaciones militares, industria, entre otras. Es un campo en desarrollo constante y está consolidándose como herramienta eficaz en cada una de estas áreas. Los resultados de este proyecto, son también una muestra y una justificación de por qué este es un campo que crece día a día.
- El algoritmo de Detección de Objetos Abandonados está en la capacidad de hacer detecciones en escena, esto supone que el sistema trabaja en tiempo real, gracias a la eficiencia lograda con esta implementación. Una de las características más comunes en sistemas de Video Vigilancia es la proyección de diferentes puntos de interés capturados con las cámaras de video. Desarrollar este tipo de algoritmos da la posibilidad, de convertir un sistema estándar de video vigilancia en un Sistema de Video Vigilancia Inteligente, donde el operador tiene la oportunidad de monitorear todo lo que sucede al instante y además, recibir también señales de alarma que se activan automáticamente y brindan soporte al operador, optimizando la labor de inspección que el mismo desempeña.
- Construir una base de datos en visión artificial, es importante para el desarrollo de proyectos actuales y futuros, pues constituye una misma fuente de material de prueba de distintos proyectos cuyos resultados se pueden contrastar, con el fin de obtener cada vez un resultado mejor. Por tanto, la base de datos Videos de Detección de Objetos Abandonados (VDOA) es un pilar que hace parte de grupo SIRP, para posteriores investigaciones en esta temática.
- Los sistemas de video vigilancia inteligente suponen más de una plataforma hardware de procesamiento en especial cuando el sistema está conformado por dos o más cámaras. La tarjeta de procesamiento Beagle Board, es una plataforma que por sus características de procesamiento y conectividad, puestas a prueba en este proyecto, encaja convenientemente en un sistema de video vigilancia inteligente, para procesar imágenes de video, con algoritmos basados en Visión por Computador y entregar la información resultante al servidor central del sistema.
- En la etapa de desarrollo se planteó que el uso de una etapa de seguimiento no era necesaria para cumplir con el propósito de detección de objetos abandonados, de acuerdo con las pruebas hechas sobre las bases de datos CAVIAR y AVSS 2007, no hubo ningún objeto abandonado que no fuese detectado. Aunque con algunos videos de VDOA, se dio esta circunstancia, cabe resaltar que esta se debe principalmente a los umbrales que tienen que ver en el proceso de binarización. Pues los objetos que no fueron detectados se ubicaron en espacios donde quedaron camuflados.
Por otra parte se esperaría que un sistema basado en detección de objetos abandonados por seguimiento detectara objetos que se abandonan tras un objeto perteneciente al fondo, sin embargo al probar la implementación Abandoned Object Detection de Matlab con el video de

CAVIAR LeftBag_BehindChair, no se obtuvo ninguna detección. Este hecho soporta la idea, que no es necesario incluir una etapa de seguimiento en un algoritmo de detección de objetos abandonados.

- La eficiencia de un sistema depende de cuánto beneficio se obtiene a pesar del costo, esta implementación obtiene una eficiencia en la carga computacional que aplica al hardware de procesamiento, implementando modelos que demandan poca complejidad computacional, sin embargo esta no es la única eficiencia que logra el algoritmo, al poder ser ejecutado en tiempos más cortos cumple fácilmente el requisito detectar objetos en escena. Ahora bien, detectar un objeto en escena depende de qué tan rápido sucedan cambios en la misma. Por ejemplo, si se sabe que lo que está contenido en la escena no tiene cambios significativos respecto al tiempo, en un tiempo menor a un segundo, el mínimo requerido para que el sistema haga la detección en escena es igual a este valor, los tiempos logrados por debajo de este valor, serían considerados como ganancia. Tal ganancia podría ser aprovechada mediante un control del tiempo de ejecución del algoritmo, este consistiría en poner en espera la ejecución del código desde que se cumple la ejecución del algoritmo hasta que se llega al mínimo requerido, entre más grande sea esta diferencia, mayor será el tiempo en espera, y menor el consumo de energía. De esta forma el sistema no solo sería más eficiente en cuanto a la carga computacional, sino también en cuanto al consumo energético, que representa un menor gasto de recursos monetarios y recursos naturales que contribuyen a la conservación del planeta.

7.2 Recomendaciones

- Hoy en día los procesadores cuentan con arquitecturas que proveen la opción de ejecutar más de una tarea a la vez, esta propiedad puede ser de gran provecho para implantar este algoritmo, ya que algunas labores que desarrolla pueden ser llevadas a cabo en el mismo tiempo de ejecución, paralelizando así la ejecución del algoritmo. Esto conducirá a una implementación aún más eficiente.
- En secuencias de video como AVSS_AB_Hard_Divx y AVSS_AB_Hard_Divx se observó que el comportamiento de los peatones cambia según la zona de la escena, esto supone dos naturalezas de escena distintas en una misma. Una forma de solucionar este inconveniente es caracterizar el sistema para delimitar una o varias zonas de la imagen sobre las cuales el *Tiempo de Espera* y el *Tiempo de Adaptación* tengan los valores adecuados para cada caso. Esto reduciría el ruido y lograría mayor efectividad.
- La etapa de detección de movimiento se implementó con el fin de evitar sombras en la estimación del fondo. Este procedimiento podría ser más preciso si en vez de involucrar el movimiento de toda la escena, para construir el fondo, se dividiera la escena y se construyera el fondo respecto a la misma división dependiendo del movimiento en cada sección. Por ejemplo, si la imagen de video se divide como en la Figura 49, el sistema no actualizaría los dos cuadrantes de abajo si en cada uno de ellos hay demasiado movimiento, de esta forma el sistema ganaría eficiencia y precisión en esta etapa del sistema.



Figura 49 División de Escena

8 BIBLIOGRAFÍA:

- [1] SISTEMA DE VIDEOVIGILANCIA. FONDO DE VIGILANCIA Y SEGURIDAD DE BOGOTÁ © 2010.
http://www.fvs.gov.co/portal/index.php?option=com_content&view=article&id=71:sistema-de-videovigilancia&catid=43:politicas-planes-yo-lineas-estrategicas. Consultado el 5 de diciembre de 2011
- [2] Monitoring Human Activity. <http://mha.cs.umn.edu/>.
- [3] Ergonomía Básica. <http://www.eduteka.org/ErgonomiaBasica.php>.
- [4] Síndrome Visual Del Computador.
http://www.compumedicina.com/oftalmologia/oft_150101.htm.
- [5] Paolo Spagnolo, Andrea Caroppo, Marco Leo, Tommaso Martiriggiano and Tiziana D’Orazio. An “Abandoned/Removed Objects Detection Algorithm and Its Evaluation on PETS Datasets”. Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS’06), 2006
- [6] Juan Carlos San Miguel, José M. Martínez. “Robust unattended and stolen object detection by fusing simple algorithms”. Grupo de Tratamiento de Imágenes. Escuela Politécnica Superior, Universidad Autónoma de Madrid, SPAIN
- [7] Álvaro Bayona. “Detección de objetos abandonados/robados en secuencias de vídeo-seguridad”. Universidad Autónoma De Madrid. Escuela Politécnica Superior, 2009.
- [8] Nathaniel Bird, Stefan Atev, Nicolas Caramelli, Robert Martin, Osama Masoud, Nikolaos Papanikolopoulos. “Real Time, Online Detection Of Abandoned Objects In Public Areas”. Department Of Computer Science And Engineering. University Of Minnesota. IEEE International Conference On Robotics And Automation, 2006.
- [9] M. D. Beynon, D. J. Van Hook, M. Seibert, A. Peacock, and D. Dudgeon, “Detecting abandoned packages in a multi-camera video surveillance system,” Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 221-228, July 2003.
- [10] Computer Vision System Toolbox. Abandoned Object Detection.
<http://www.mathworks.com/products/computer-vision/demos.html?file=/products/demos/shipping/vision/videoabandonedobj.html>. 2011 The MathWorks, Inc. consultado el 5 de diciembre de 2011
- [11] Computer Vision System Toolbox 4.0.
<http://www.mathworks.com/products/datasheets/pdf/computer-vision-system-toolbox.pdf>. 2011 The MathWorks, Inc. consultado el 5 de diciembre de 2011
- [12] Weihua Wang Zhijing Liu. A New Approach For Real-time Detection Of Abandoned And Stolen Objects School of Computer Science and Technology 2010 International Conference on Electrical and Control Engineering.
- [13] Qiuji Li, Yaobin Mao, Zhiquan Wang, Wenbo Xiang. “Robust Real-Time Detection Of Abandoned And Removed Objects*?”. IEEE computer society. Fifth International Conference On Image And Graphics, 2009.
- [14] ING. FRANCISCO CARLOS CALDERÓN BOCANEGRA. Sistema de adquisición de parámetros de tráfico vehicular, PONTIFICIA UNIVERSIDAD JAVERIANA, BOGOTÁ DC, 2010.
- [15] Ana González Marcos, Francisco Javier Martínez de Pisón Ascacibar, Alpha Verónica Pernía Espinoza, Fernando Alba Elías, Manuel Castejón Limas, Joaquín Ordieres Meré y Eliseo Vergara González (Integrantes del Grupo de Investigación EDMANS). TÉCNICAS Y

ALGORITMOS BÁSICOS DE VISIÓN ARTIFICIAL. Servicio de publicaciones, Universidad de la Rioja 2006

- [16] José Francisco Vélez Serrano. Visión por computador. Universidad Rey Juan Carlos 2003
- [17] Julián Quiroga Sepúlveda. PROC. DE IMÁGENES: Conceptos Básicos. Apuntes de Clase Señales e Imágenes, Pontificia Universidad Javeriana 2010.
https://docs.google.com/viewer?a=v&pid=explorer&chrome=true&srcid=0B7IdP8eYshy8ZTImYjhlYzMtZGY4Ny00ZmIwLWE1YzYtM2E0YTA5NmMwN2M4&hl=en_US. consultado el 30 de noviembre de 2011
- [18] Julián Quiroga Sepúlveda. PROC. DE IMÁGENES: Imágenes Binarias. Apuntes de Clase Señales e Imágenes, Pontificia Universidad Javeriana 2010.
https://docs.google.com/viewer?a=v&pid=explorer&chrome=true&srcid=0B7IdP8eYshy8YTQ1ZWQ0M2EtZTkYNS00NWFMlWE5NjEtODI0MjYwNjI4NzI2&hl=en_US. consultado el 30 de noviembre de 2011
- [19] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, Sebastopol, Canada: O'Reilly, 2008.
- [20] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, Norman Haas, Max Lu, Hans Merkl, Sharath Pankanti, Andrew Senior, Chiao-Fe Shu, and Ying Li Tian. Smart Video Surveillance. IEEE SIGNAL PROCESSING MAGAZINE [38] MARCH 2005
- [21] Arun Hampapur, Lisa Brown, Jonathan Connell, Sharath Pankanti, Andrew Senior and Yingli Tian. IBM T.J. Watson Research Center, Smart Surveillance: Applications, Technologies and Implications. Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on
- [22] IBM Smart Surveillance System (Previous PeopleVision Project).
<http://www.research.ibm.com/peoplevision/>.
- [23] Análisis Inteligente De Video (IVA 3.5).
http://www.boschsecurity.es/content/language1/html/2737_ESN_XHTML.asp.
- [24] K. KIM, T. CHALIDABHONGSE, D. HARWOOD, and L. DAVIS, "Real-time foreground-background segmentation using codebook model," Real-Time Imaging, vol. 11, 2005, pp. 172-185.
- [25] Kyungnam Kima,_, Thanarat H. Chalidabhongseb, David Harwooda, Larry Davisa Real-time foreground-background segmentation using codebook model. 2005 Elsevier Ltd.
- [26] Massimo Piccardi. Background subtraction techniques: a review. 2004 IEEE International Conference on Systems, Man and Cybernetics.
- [27] NIST/SEMATECH e-Handbook of Statistical Methods,
<http://www.itl.nist.gov/div898/handbook/>, date.
<http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm>
- [28] Zheng Yi ; Fan Liangzhong ; Moving object detection based on running average background and temporal difference. Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on.
- [29] P. KaewTraKulPong and R. Bowden .An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01. Sept 2001
- [30] Qiujie Li, Yaobin Mao, Zhiquan Wang, Wenbo Xiang. Robust Real-time Detection of Abandoned and Removed Objects*. 2009 Fifth International Conference on Image and Graphics

- [31] J. Davis and A. Bobick, "The representation and recognition of action using temporal templates" (Technical Report 402), MIT Media Lab, Cambridge, MA, 1997.
- [32] Paul L. Rosin. Thresholding for Change Detection. Department of Information Systems and Computing, Brunel University, Uxbridge.
- [33] BeagleBoard.org - hardware-xM. <http://beagleboard.org/hardware-xM>.
- [34] PlayStation Eye. http://en.wikipedia.org/wiki/PlayStation_Eye. Consultado el 22 de noviembre de 2011
- [35] Cámara PlayStation® Eye. <http://co.playstation.com/ps3/accesorios/scph-98047.html>. Consultado el 22 de noviembre de 2011
- [36] The Ångström Distribution. <http://www.angstrom-distribution.org/>.
- [37] Tutorial de OpenCV. http://docencia-eupt.unizar.es/ctmedra/tutorial_opencv.pdf
- [38] FullOpenCVWiki. <http://opencv.willowgarage.com/wiki/FullOpenCVWiki>. Consultado el 22 de noviembre de 2011
- [39] BeagleBoard System Reference Manual Rev C4. http://beagleboard.org/static/BBSRM_latest.pdf. Consultado el 29 de noviembre de 2011
- [40] i-Lids dataset for AVSS 2007. AVSS 2007, 2007 IEEE International Conference on Advanced Video and Signal based Surveillance. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html. Consultado el 1 de diciembre de 2011
- [41] Prof. Robert Fisher, Dr. Jose Santos-Victor, Prof. James Crowley, Institut National Polytechnique de Grenoble, France. CAVIAR Test Case Scenarios. http://homepages.inf.ed.ac.uk/rbf/CAVIAR_DATA1/ Consultado el 1 de diciembre de 2011
- [42] Robert Fisher CAVIAR: Context Aware Vision using Image-based Active Recognition. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>. Consultado el 1 de diciembre de 2011
- [43]