

INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA
ESCUELA DE INGENIERÍA Y GESTIÓN



CONTEO DE PERSONAS

AUTORES: Fariña, Segundo (Leg. N° 56176)
Rodríguez, Gastón (Leg. N° 56020)

DOCENTE TITULAR: Fierens, Pablo

**TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA**

Lugar: Buenos Aires, Argentina
Fecha: 10/12/2020

Índice

1. Resumen	3
2. Introducción	4
2.1 Objetivo del trabajo	4
2.2 Posibles aplicaciones	5
2.3 Desarrollo del trabajo	5
2.4 Trabajos Existentes	6
2.5 Anatomía de un video	7
2.6 Sistema	9
3. Análisis de casos	10
3.1 Dos individuos que cruzan juntos la barrera	10
3.2 Individuos con mochilas o equipajes	11
3.3 Guardia de seguridad	11
3.4 Cambios de iluminación	11
3.5 Resolución de la imagen	12
4. Algoritmo	13
4.1 Detección y sustracción del fondo	13
4.1.1 Detección del fondo	13
4.1.2 Eliminación del ruido en la imagen	14
4.1.3 Agrupación de zonas blancas	16
4.2 Detección de individuos	17
4.2.1 Detección de contornos	17
4.2.2 Filtrado de contornos espurios	17
4.2.3 División de contornos muy grandes	18
4.2.4 Cálculo de centroides	19
4.3 Seguimiento de personas	19
4.3.1 Asociación de centroide con persona	19
4.3.2 Determinación de la dirección de cruce	21
4.3.3 Registro de barrera cruzada	24
5. Ajustes de parámetros	25
5.1 Umbrales para la detección de una persona	25
5.2 Barreras de cruce	25
5.3 Configuración	26
6. Resultados	28
6.1 Dos individuos que cruzan juntos la barrera	28
6.2 Individuos con mochilas o equipajes	29
6.3 Guardia de seguridad	30

6.4 Cambios de iluminación	31
6.5 Resolución de la imagen	32
7. Conclusión	33
8. Apéndice	34
8.1 Uso del programa	35
8.1.1 Instalación	35
8.1.2 Ejecución por línea de comandos	35
8.1.3 Ejecución por interfaz gráfica	36
Referencias	37

1. Resumen

El objetivo del presente trabajo es el desarrollo un sistema de conteo de personas utilizando una cámara desde una vista cenital. Un caso de uso típico es el de un local comercial con la cámara registrando el acceso al mismo de manera que el sistema permite obtener estadísticas del flujo de clientes a lo largo de una jornada. En este sentido, el sistema no sólo cuenta la cantidad de personas en el área de interés, sino que también analiza la dirección de las mismas. Un requerimiento del sistema es que tenga un muy bajo costo, por lo que se dispone que las imágenes sean tomadas por una cámara web convencional.

Este proyecto fue motivado por las necesidades de una empresa que brindaba servicios de conteo de personas, fundamentalmente, a locales comerciales. Dicha empresa contaba con un sistema funcionando, pero que presentaba dificultades en ciertos casos particulares. En estos casos de interés es donde se focaliza el trabajo, a saber: la contabilización correcta de dos individuos que aparecen en las imágenes juntos, muy cerca entre sí; la contabilización como una sola unidad de personas con mochilas o equipajes; evitar el conteo de gente que no atravesase efectivamente la escena en una dirección predeterminada (por ej., individuos que son registrados por la cámara pero que no entran ni salen del local comercial); adaptar adecuadamente el sistema a variaciones de la iluminación y de la resolución del video.

El diseño del programa tuvo en cuenta estos casos particulares desde el inicio. Para comprobar el correcto funcionamiento del mismo se realizaron experimentos especialmente diseñados para representar cada situación. En base a estas pruebas, se muestra que el sistema desarrollado es robusto, adaptándose correctamente a todos los escenarios planteados, y con un bajo costo computacional que hace posible el funcionamiento en tiempo real y su implementación en un hardware del tipo *single board computer*.

2. Introducción

El conteo de personas es el proceso mediante el cual se determina la cantidad de individuos en un área de interés. La forma más elemental de proceder es contar de manera manual (por ej., como en ocasiones hace la tripulación de un vuelo comercial). Por otro lado, existen soluciones que hacen uso de la tecnología para poder realizar el conteo de manera automática. Esto puede lograrse con distintos dispositivos, como pisos inteligentes con sensores de presión, rayos infrarrojos, imágenes térmicas, rastreadores WiFi o visión por computadora. Estrechamente relacionado con el conteo de individuos es el problema de la determinación del flujo de personas en diversas direcciones. En este caso no sólo es de interés estimar la cantidad de individuos, sino también fijar desde y hacia dónde se dirigen. En este proyecto, se trabaja sobre un caso particular de estimación de flujos, en sólo una dirección y dos sentidos posibles, sobre la base de visión por computadora.

2.1 Objetivo del trabajo

El conteo de personas es un problema desafiante con muchos ángulos desde los que se lo puede atacar. En este trabajo realizamos un contador por medio de visión de computadora, el cual consiste en un sistema de análisis de imágenes en tiempo real capaz de detectar cuándo y en qué dirección una persona atraviesa una barrera virtual.

El objetivo es lograr una solución económica, disponiendo únicamente de una cámara web no muy sofisticada colocada sobre el área que se desea supervisar. Asimismo, se requiere que una vez instalado y se realice una configuración inicial, sea capaz de ajustarse automáticamente a ciertos cambios del entorno, como pueden ser variaciones en la iluminación o aparición y desaparición de objetos estáticos (e.g., movimiento de mobiliario).

Además del diseño básico del sistema, el presente trabajo se enfoca en la resolución de ciertas problemáticas comunes (ver Sección 3) de las soluciones ya existentes. Para poder evaluar el funcionamiento del sistema propuesto, se diseñan experimentos específicos que reproduzcan las condiciones que generan dichas dificultades.

2.2 Posibles aplicaciones

El conteo de personas puede tener diversas aplicaciones prácticas en la vida cotidiana. En recintos con aforos que requieren de un control, como por ejemplo locales bailables [1], canchas de fútbol o espacios para conciertos [2], puede utilizarse para asegurar el cumplimiento de la reglamentación vigente. En las circunstancias actuales que

atraviesa el mundo debido al COVID-19, puede ser una aplicación muy útil para controlar el acceso a recintos que no excedan la cuota máxima de gente por metro cuadrado habilitada por el gobierno [3].

En tiendas departamentales se pueden conocer los horarios pico de sus negocios y cómo fluctúa el tránsito de gente a lo largo del día, el tiempo que pasa cada cliente dentro del negocio y su relación con la facturación evaluada como el porcentaje de conversión, esto es, el porcentaje de total de visitas contra el número de ventas, entre otras muchas variables de interés.

2.3 Desarrollo del trabajo

Originalmente el trabajo iba a ser desarrollado basándonos únicamente en los videos obtenidos de un prototipo instalado en el Laboratorio de Optoelectrónica en la sede del ITBA de Puerto Madero. El mismo es un arco metálico, con dos cámaras web montadas en la parte horizontal apuntando hacia el piso, de manera tal que obtenga una imagen de las personas pasando por debajo de este (ver Fig. 1). Ambas cámaras son de la misma marca y modelo: *TP Link NC220* , con una resolución de 640x480 píxeles.

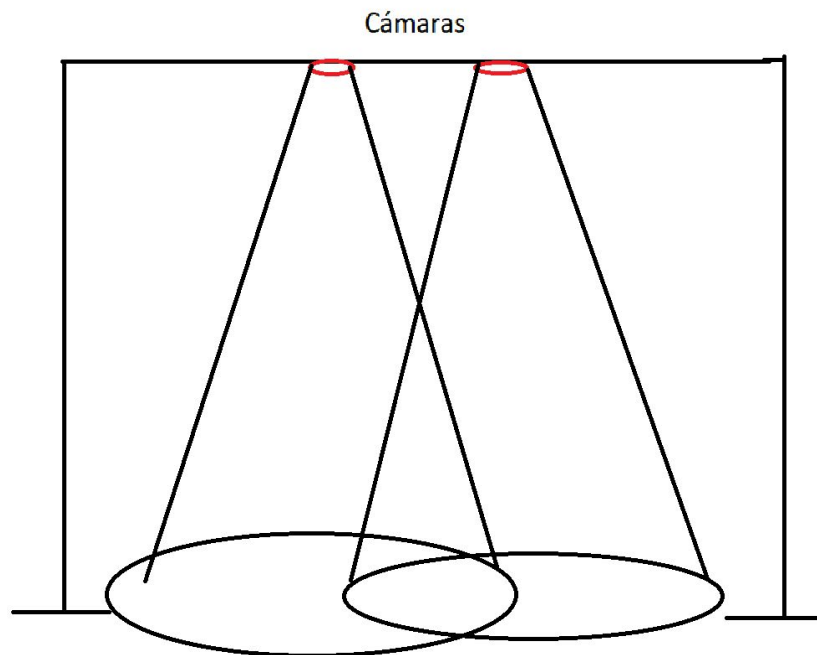


Figura 1: Esbozo del prototipo

Esta idea debió ser adaptada por las restricciones impuestas por el Aislamiento Social Preventivo Obligatorio (ASPO) por la pandemia de COVID-19. Se cuenta con una única grabación utilizando esta estructura. Si bien en dicha grabación se reprodujeron varias de las condiciones de interés (ver Sección 3), no todas pudieron ser tenidas en cuenta. Debido a estas circunstancias, se recurrió a soluciones caseras utilizando celulares y colocándolos en nuestras casas. La resolución de los videos fue reducida para poder simular las cámaras web que se utilizan para realizar este tipo de tareas.

2.4 Trabajos Existentes

Se debe destacar que existen soluciones comerciales a este problema, como la de la empresa *Urbix Technologies* quien donó la estructura al laboratorio del *ITBA*. También existen trabajos en la literatura científica que plantean diversas soluciones. En este sentido, no es el objetivo del presente proyecto realizar un aporte novedoso desde el punto de vista científico, sino que, basándonos en la literatura existente, se procura desarrollar un sistema eficiente de conteo de personas. Se hace foco particularmente en situaciones difíciles para el procesamiento de imágenes, algunas de las cuales se describirán más adelante.

Los métodos expuestos en la literatura científica sobre el tema siguen en términos generales dos escuelas de procesamiento: técnicas clásicas de análisis y tratamiento de imágenes y el uso del Aprendizaje Automático o *Machine Learning*. Esta distinción no es definida, ya que los sistemas basados en Machine Learning suelen usar varias técnicas clásicas durante el preprocesamiento de las imágenes.

Los *papers* que se inclinan por utilizar Aprendizaje Automático [4][5] utilizan el mismo para crear un modelo sobre cómo se ven las personas desde un ángulo determinado de la cámara, y luego usar algún otro algoritmo para determinar si las personas pasan o no el área vigilada por la cámara en alguna dirección. Este método de detección de personas debe ser entrenado con un robusto *dataset* de imágenes mostrando los posibles contextos en los cuales se podría instalar el sistema, y debiera ser lo suficientemente variado en cantidad de localizaciones y alturas a las cuales se dispone para llegar a resultados satisfactorios. Este entrenamiento suele ser computacionalmente muy costoso, pero una vez que se tiene el modelo entrenado ejecutarlo no requiere semejante capacidad de cómputo.

Por otro lado, los *papers* que utilizan técnicas de procesamiento de imágenes clásicas [6][7] suelen seguir las mismas ideas principales entre ellos, y varían en los detalles finos de las implementaciones de los grandes algoritmos. Lo que se observa en la literatura es que se siguen los siguientes pasos: se detecta el fondo de la escena, se hace la separación fondo-frente de la imagen para obtener los objetos en movimiento, y luego se define alguna heurística que determine cómo se mueven los objetos detectados en la imagen.

Después de consultar con la literatura disponible, nos inclinamos por implementar las técnicas clásicas de procesamiento de imágenes ya que tenemos experiencia en el área y, en pruebas preliminares, obtuvimos resultados satisfactorios.

Analizando el mercado existente, encontramos múltiples productos a la venta con el fin de recabar información sobre el movimiento de las personas en locales comerciales [8][9][10]. Estos productos son en su mayoría dispositivos físicos con más de un sensor, siempre cuentan con al menos una cámara común, y luego se le agrega un detector de profundidad o una cámara infrarroja. Estos dispositivos toman el *feed* de los sensores y reconstruyen un mapa 3D de la escena con el cual arman una representación más fiel a la realidad. Si bien no están disponibles los detalles de implementación, observando las especificaciones de los productos y la información promocional correspondiente, pareciera que las soluciones en el mercado se inclinaron por usar Aprendizaje Automático para reconstruir las escenas que capturan sus sensores. Estos productos suelen ofrecer,

además del conteo de personas, *heatmaps* de la cantidad de gente por zona de los locales, manejo y *analytics* de las filas en las cajas, detección de qué objetos son los más llamativos para los clientes, etcétera.

Nuestro trabajo toma ideas de tanto las soluciones delineadas en la literatura científica como en el desarrollo profesional de la empresa *Urbix Technologies*. De esta manera logramos combinar las mejores ideas disponibles, culminando en un programa de relativo bajo computacional y buena precisión en nuestros casos de uso.

2.5 Anatomía de un video

Como se mencionó previamente, el trabajo se basa en el análisis de videos, ya sea de forma *offline* cargándolo para ser analizado o con un feed de imágenes en tiempo real. A fin de dar un entendimiento básico se explicará cómo es la estructura de datos de un video digital del tipo que se usa en este proyecto. Cabe destacar que esta no es la única estructura de datos posible pero no está en los objetivos del trabajo explicarlas todas sino las que usaremos para el análisis del mismo.

Un video es una secuencia de imágenes o frames. Una imagen es una matriz con un alto y ancho determinados donde cada celda de la matriz representa un píxel en el cual se guarda el color del mismo. Para poder formar un color se puede representar en intensidades de rojo, verde y azul, haciendo que la combinación de estos genere el color deseado. Una imagen a color *RGB* (por la sigla en inglés *Red, Green and Blue*) está compuesta por tres matrices en las que las dimensiones están definidas por la altura y el ancho de la imagen. Se tiene una matriz por cada color *RGB*, donde cada elemento determina la intensidad de ese color en un píxel. Estos valores de intensidad pueden tomar un valor entre 0 y 255, ya que se representan por un número de 8 *bits*. Reiteramos que esta no es la única forma de representación de una imagen pero es la que usamos en nuestro caso.

En el caso de un frame en escala de grises puede representarse como una única matriz donde los valores van de 0 a 255 y representan la intensidad del blanco en el píxel, siendo 0 completamente negro, 255 completamente blanco y los valores en el medio el espectro de grises visibles.

Existen varias características propias del video que cambian en la calidad, y tamaño del mismo. Una de ellas es la tasa de refresco o cuadros por segundo, esto es, la cantidad de frames que se muestran en un segundo. A más cuadros por segundo se obtiene un video mucho más fluido, pero por otro lado se deben almacenar más *frames* impactando en el tamaño total del video. Para lograr la ilusión de una imagen en movimiento, la velocidad mínima de carga de las imágenes es de unas quince imágenes por segundo, sin embargo el ojo humano puede distinguir movimiento mucho más fluido por encima de los 48 fotogramas por segundo. Por otro lado, para optimizar el tamaño del archivo se usan distintas técnicas de compresión con pérdida de información que afecta la calidad del video.

Existen múltiples maneras de realizar representaciones digitales de imágenes y videos, saliéndose de las matrices *RGB* con representaciones de 8 bits. Es necesario remarcar que existen distintas maneras de representar los colores (por ej., *HSV, HSL*), se pueden utilizar más colores apelando a más de ocho *bits* por cada componente, y se utilizan diversos algoritmos de compresión con pérdida de la imagen. Sin embargo, todos estos aspectos exceden el alcance del presente proyecto y no forman parte de nuestro trabajo.

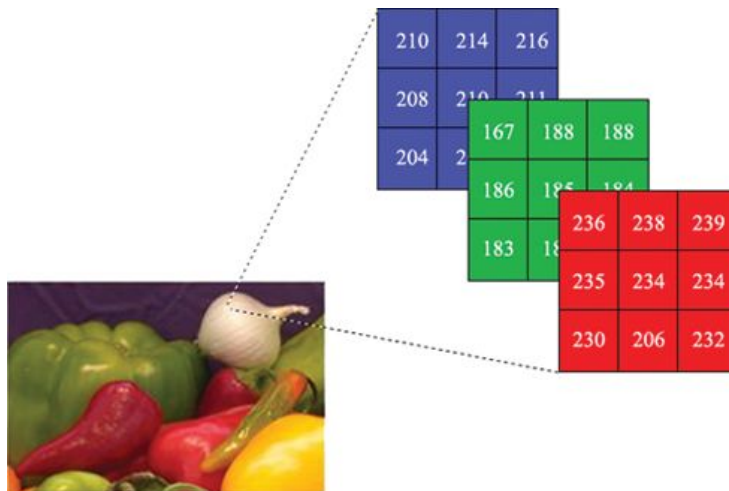


Figura 2: Demostración visual de la representación digital de los píxeles de una imagen. Imagen tomada de [11].

2.6 Sistema

El sistema se encuentra escrito en el lenguaje de programación *python3*. El análisis y tratamiento de imágenes tiene una comunidad de desarrolladores de código abierto muy grande en este lenguaje, por lo que existen bibliotecas con las últimas metodologías de análisis de imágenes ya implementadas, lo cual fue óptimo para nuestro desarrollo , permitiéndonos prototipar y descartar ideas con rapidez.

Se hace uso de bibliotecas externas como *opencv*, que implementa ciertos algoritmos conocidos para el procesamiento de imágenes, *numpy* para manipulación de matrices y *tinker* para la interfaz gráfica.

3. Análisis de casos

El principal foco de este proyecto es el análisis de casos particulares que causan problemas en los sistemas existentes. Los mismos fueron explicados por la empresa *Urbix Technologies* (comunicación privada). El objetivo es poder analizarlos uno a uno y tratar de buscar mejoras al algoritmo para que pueda funcionar como es esperado.

Antes de entrar en el detalle de los casos, es necesario destacar cuáles son algunas de las limitantes del desarrollo. Se utilizan cámaras web de muy bajo costo, con poca definición y un bajo *frame-rate*. Esto significa no sólo que las imágenes cuentan con una baja definición, sino que la cantidad de *frames* por segundo será menor que el de una cámara común, lo que puede ocasionar grandes saltos entre un *frame* y el siguiente, hecho que el sistema debe considerar especialmente. Muchas soluciones comerciales utilizan cámaras estereoscópicas que permiten tener una idea de profundidad y ayudan a resolver algunos de los problemas que se describen a continuación. Asimismo, la iluminación en el área no es ideal: es fácil confundir sombras con personas en ausencia de algún elemento que permita distinguir profundidades. Finalmente, se procura utilizar algoritmos de bajo nivel de cómputo con el objetivo deseable de incorporarlos a una *single-board computer*. Los casos se asumen confinados a una estructura como la de la Fig. 1 a Fig. 1, por la cual las personas atraviesan un portal vertical en relación a la imagen tomada por la cámara, y con un máximo de dos individuos pasando simultáneamente una al lado de la otra por la escena.

Los casos a analizar se describen a continuación.

3.1 Dos individuos que cruzan juntos la barrera

Analizando sistemas ya implementados, se pudo ver que una de las fallas de los mismos es cuando dos individuos cruzan la barrera muy juntos (tanto uno al costado del otro como uno delante del otro). El sistema contabiliza solo una persona que cruza la barrera cuando tendría que identificar a dos. La Fig. 3 muestra ejemplos de esta situación.



Figura 3: Individuos cruzando juntos la barrera uno detrás del otro (izquierda) y lado a lado (derecha). Imágenes tomadas de los videos de prueba en el Laboratorio de Optoelectrónica del ITBA.

3.2 Individuos con mochilas o equipajes

Análogamente al caso anterior, se presentaron situaciones en modelos ya implementados donde individuos con coches de bebés o mochilas podían ser detectados como dos personas y ser contados dos veces al pasar la barrera. Se quiere analizar este caso y solo contabilizarlo como un único individuo. La Fig. 4 muestra ejemplos de esta situación.



Figura 4: Individuos con equipajes. Imágenes tomadas de los videos de prueba en la casa de un integrante del grupo.

3.3 Guardia de seguridad

Un caso especial es cuando un individuo se encuentra en la zona de visión de la cámara, cruzando varias veces la barrera de manera errática. Éste no debería influir en el conteo ya que no es de interés para el sistema. Ciertamente, esto no se trata sólo de un problema de procesamiento de imágenes, pero es de interés. Un caso típico es el de un guardia de seguridad en la puerta de un local comercial, atento a la entrada y salida de clientes. La Fig. 5 muestra un ejemplo en el cual un individuo no atraviesa dirección perpendicular la barrera (virtualmente horizontal), sino que camina en forma paralela a la misma.



Figura 5: Guardia de seguridad caminando horizontalmente por la escena.

3.4 Cambios de iluminación

Un aspecto importante del sistema será su versatilidad ya que esperamos que el mismo sea capaz de adaptarse a diversos entornos. Una de las variables a controlar es el cambio de iluminación en la escena. Se desea que el sistema sea capaz de funcionar correctamente bajo condiciones de poca iluminación o con luces que generan sombras que pueden llevar a confusión. En condiciones reales, la iluminación puede cambiar a lo largo del día. La Fig. 6 muestra ejemplos con alta y baja iluminación.



Figura 6: Cambios de iluminación a lo largo del día

3.5 Resolución de la imagen

Siguiendo con la versatilidad del sistema, es deseable que sea capaz de funcionar correctamente con diversas cámaras, con distintos tipos de resolución y relación de aspecto. La Fig. 7 muestra ejemplos de imágenes con distintas relaciones de aspecto.

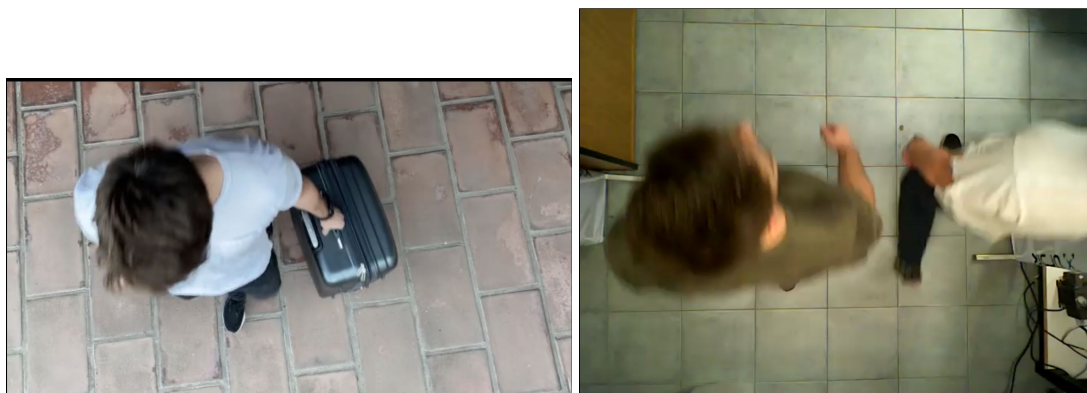


Figura 7: Cambios en la relación de aspecto de la imagen (16:9 vs. 4:3)

4. Algoritmo

El algoritmo puede ser dividido en tres grandes pilares organizados de manera secuencial, de forma que cada uno deja la información lista para el próximo. Estos tres pilares son:

- Detección y sustracción del fondo (Sección 4.1): el procesamiento del *frame* para extraer una imagen donde se detecten los objetos que no pertenecen al fondo en ella.
- Detección de individuos (Sección 4.2): una vez obtenidas las imágenes con sólo los objetos pertinentes en ellas, hay que determinar si son personas o no, y dónde se hallan en la imagen.
- Seguimiento de personas (Sección 4.3): una vez que se sabe en qué lugar se ubican las personas en la imagen, hay que hacer un seguimiento a través de los *frames* para determinar su recorrido a lo largo del tiempo..

4.1 Detección y sustracción del fondo

4.1.1 Detección del fondo

Para hacer la detección de personas en un video, se debe realizar una distinción entre el fondo y los objetos en movimiento. Con ese objetivo, usamos el detector y sustractor de fondos *MOG2* [12][13].

Este algoritmo mantiene un modelo de mezcla de distribuciones normales para la intensidad de cada píxel y lo adapta mediante una memoria implícita que tiende a olvidar la información de imágenes más lejanas en el tiempo. A la hora de determinar si un píxel pertenece al fondo de una imagen, se evalúa la probabilidad de dicha pertenencia sobre la base del modelo. Una forma de representar esta probabilidad es asignando al píxel, en una imagen en escala de grises, un valor de intensidad inversamente proporcional a la misma (cuanto menos probable sea, más brillante será el píxel). La Fig. 8 muestra un ejemplo de esta representación.



Figura 8: Frame original (izquierda) y luego de sustraer el fondo (derecha). Obsérvese que la intensidad es inversamente proporcional a la probabilidad estimada de pertenecer al fondo. Por lo tanto, cuanto más claro es un píxel, menor es la probabilidad de ser parte del mismo.

4.1.2 Eliminación del ruido en la imagen

Una vez ejecutado el sustractor de fondos, se le aplica una operación de *binarización*, de manera tal que solo los objetos más brillantes queden en blanco, y todo lo demás en negro. Luego de experimentar con distintos métodos de binarización, optamos por el más simple de todos, *threshold binarization*, ya que los distintos algoritmos conocidos intentan balancear la cantidad de blancos y negros que quedan en la imagen final, mientras que a nosotros nos interesan solo los valores más altos de intensidad en la imagen, ya que asumimos que los grises son debidos a cambios de iluminación o sombras. La Fig. 9 muestra el resultado de la binarización (comparar con la Fig. 8).



Figura 9: Frame luego de binarización

Como se puede observar en la Fig. 9, la imagen aún contiene mucho ruido debido a los cambios de iluminación en la escena que son falsamente detectados como objetos en

vez de seguir siendo parte del fondo. Para mitigar este problema se realizan operaciones morfológicas matemáticas sobre la imagen. La Fig. 11 muestra cómo la imagen ruidosa que resulta del sustractor de fondo binarizado (izquierda) da lugar a una con menor ruido luego de dichas operaciones (derecha).

Las operaciones que se realizan sobre la imagen son las de erosión y dilatación, realizando lo que se conoce como una “apertura” de la imagen (ver Fig. 10).

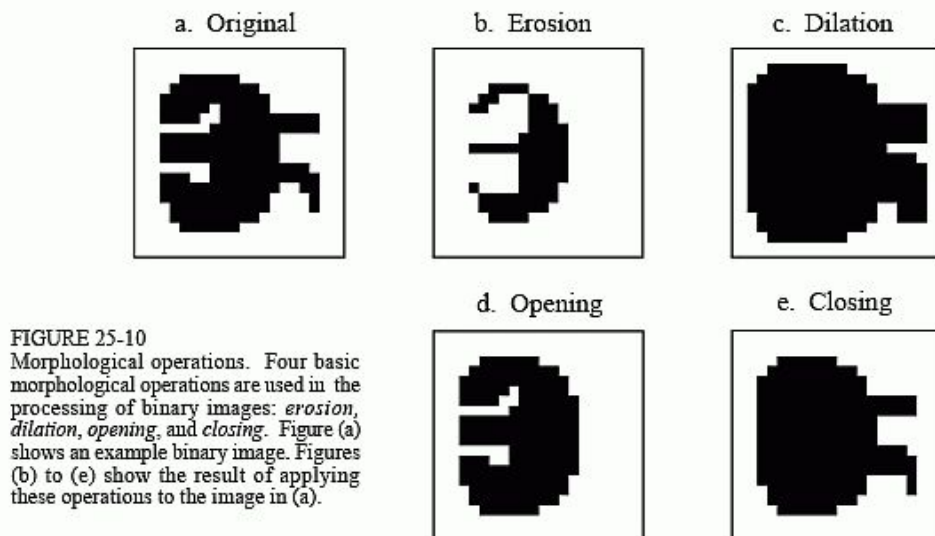


Figura 10: Operaciones morfológicas. imagen tomada de [14].

Una operación de erosión (Fig. 10b) se aplica sobre una imagen binarizada (siendo los píxeles de color blanco el fondo y color negro el objeto) de la siguiente manera: por cada píxel de la imagen que sea parte del objeto, si el mismo está tocando con un píxel que es parte del fondo se convierte el mismo en un píxel del fondo.

La operación de dilatación es la inversa a la de erosión, al ejecutarla se transforman todos los píxeles del fondo que estén en contacto con el objeto en píxeles del objeto. En la Fig. 10c se puede ver esta operación en acción. Se puede observar cómo se rellenaron las zonas blancas que antes eran parte del fondo.

La operación que se usa para la eliminación de ruido es llamada de apertura, y consiste en aplicar primero una operación de erosión seguido de una de dilatación, lo que provoca esto es que las zonas de ruido con partes muy pequeñas que parezcan ser objetos sean eliminadas junto con los bordes de los objetos más grandes, después con la operación de dilatación se trata de reconstruir la imagen original volviendo a generar los bordes eliminados no así el ruido (ver Fig. 10d).

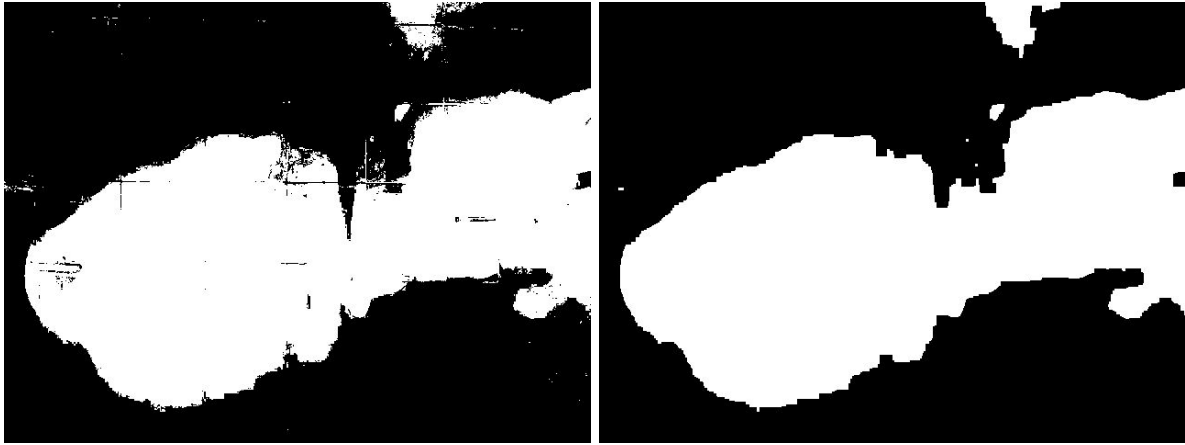


Figura 11: A la izquierda la imagen binarizada, a la derecha luego de eliminar el ruido

4.1.3 Agrupación de zonas blancas

Aún después de la operación de apertura, en la Fig. 11 (derecha) se puede observar que todavía quedan pequeñas líneas blancas dentro de los objetos detectados. En este caso se puede ver que se trata de las líneas de las baldosas del piso (ver Fig. 8 o Fig. 12, izquierda) que generan divisiones mínimas en los objetos. Para poder unir estas zonas blancas divididas por pequeñas líneas se hace una operación inversa a la que se hizo para eliminar el ruido, esto es, una operación de cerrado, la cual consiste en aplicar una operación de dilatación y luego una de erosión. El resultado es la eliminación de las líneas blancas y que queden agrupadas las zonas blancas.

En la Fig. 12 se puede observar la imagen original (izquierda) junto con la imagen que resulta de todo el proceso de sustracción de fondo.



Figura 12: Frame original (izquierda) vs. resultado de detección y sustracción de fondo (derecha).

4.2 Detección de individuos

4.2.1 Detección de contornos

Una vez que se cuenta con la imagen en blanco (objetos) y negro (fondo), se buscan los bordes de las figuras blancas mediante un método que consiste en encontrar píxeles de borde (píxel blanco con vecinos negros), y seguir los vecinos que también sean de borde, hasta encontrarse con el píxel original (habiendo circunscrito toda la figura blanca) o con un lado de la imagen [15]. De esta forma se obtienen los contornos de los objetos en movimiento dentro de nuestra escena. La Fig. 13 muestra el resultado de la detección de bordes. Obsérvese que el contorno del individuo de la derecha está limitado por el lado derecho de la imagen. Más interesante aún es el hecho que los dos individuos son envueltos por un mismo contorno, como si formaran parte del mismo objeto. Aparecen, además, varios contornos espurios que no se corresponden con ningún individuo: sombras en la remera de una persona, el talón de una zapatilla, regiones de iluminación irregular, etc.



Figura 13: Detección de contornos, marcados en verde en la imagen de la izquierda.

4.2.2 Filtrado de contornos espurios

Una vez obtenidos los contornos de la imagen, se deben descartar aquellos que no se corresponden a individuos. Una forma simple es filtrar los que son demasiado pequeños para ser una persona dentro del contexto en el que está ubicada la cámara. Para ello se analiza el área contenida por cada borde y se la compara con un umbral que puede ser ajustado en cada situación. El resultado de la aplicación de este filtro se observa en la Fig. 14. Si bien desaparecen los contornos espurios, aún la pareja de individuos es reconocida como una sola persona.

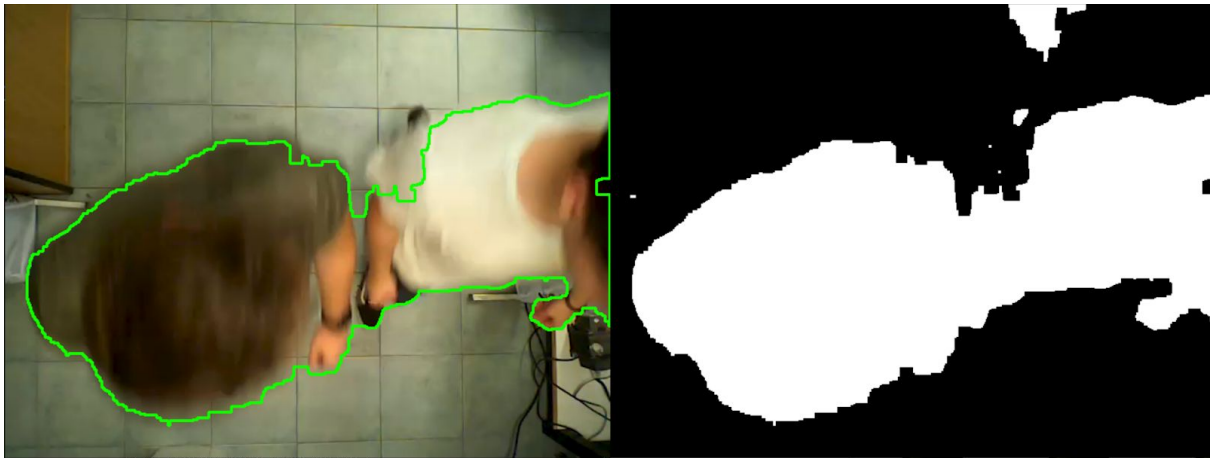


Figura 14: Filtrado de contornos espurios. En la imagen de la izquierda se puede observar que ya no aparecen los contornos correspondientes a las regiones blancas más pequeñas de la imagen de la derecha.

4.2.3 División de contornos muy grandes

Así como se descartan los contornos que son demasiado pequeños para ser considerados una persona, también hay contornos que son muy grandes como para ser considerados un único individuo. Si el área de un contorno excede un determinado valor parametrizado, entonces se considera que se trata de dos personas caminando en una gran cercanía, por lo cual dividimos el contorno por la mitad, haciendo un corte horizontal si el mismo es más alto que ancho, y uno vertical en caso contrario.

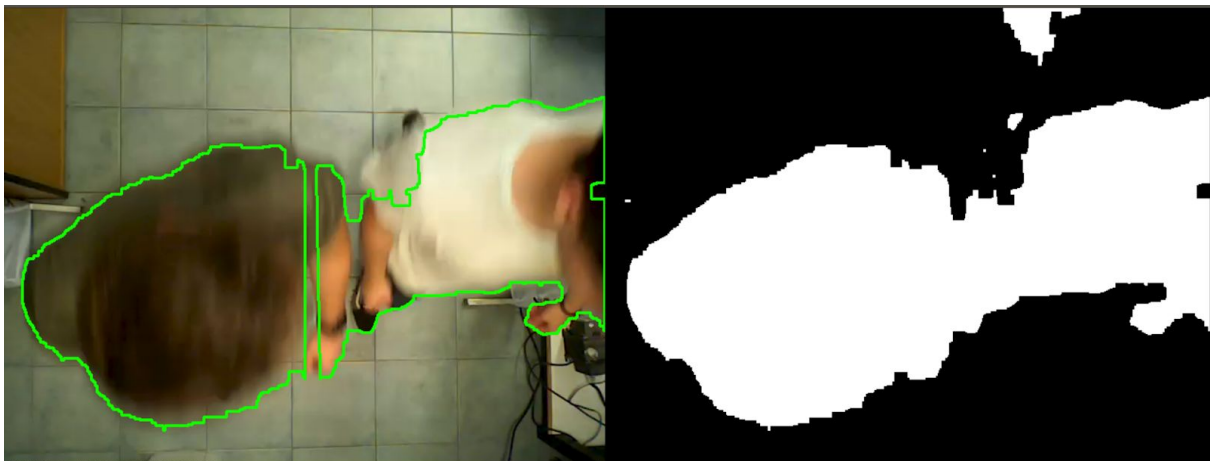


Figura 15: Corte vertical por la mitad a un contorno más ancho que alto. Se observa que el corte no divide a ambas personas de manera precisa.

El corte se realiza exactamente a la mitad del área del contorno ya que no es de nuestro interés obtener la figura exacta de la persona, sino saber qué área ocupa. Como se puede ver en la Fig. 15, el contorno fue seccionado por encima del hombro del individuo de la izquierda y no exactamente por donde las dos personas se separan, ya que la figura que crea el contorno tiene su punto medio en el eje horizontal sobre el hombro del mismo. Sin embargo, esta falta de precisión en la distinción entre dos personas que se encuentran juntas no parece tener una gran influencia en el conteo final del sistema.

4.2.4 Cálculo de centroides

Una vez obtenido el contorno de cada persona, se calcula el centroide del mismo. El objetivo es simplificar el seguimiento de los individuos reduciéndolo al seguimiento de los centroides. Para obtenerlas coordenadas del centroide se toma el promedio de las coordenadas de los puntos blancos dentro del contorno, como se observa en la Fig. 16. La Fig. 17 muestra el mismo resultado, pero con cada contorno esquematizado como una región rectangular con el mismo centroide.

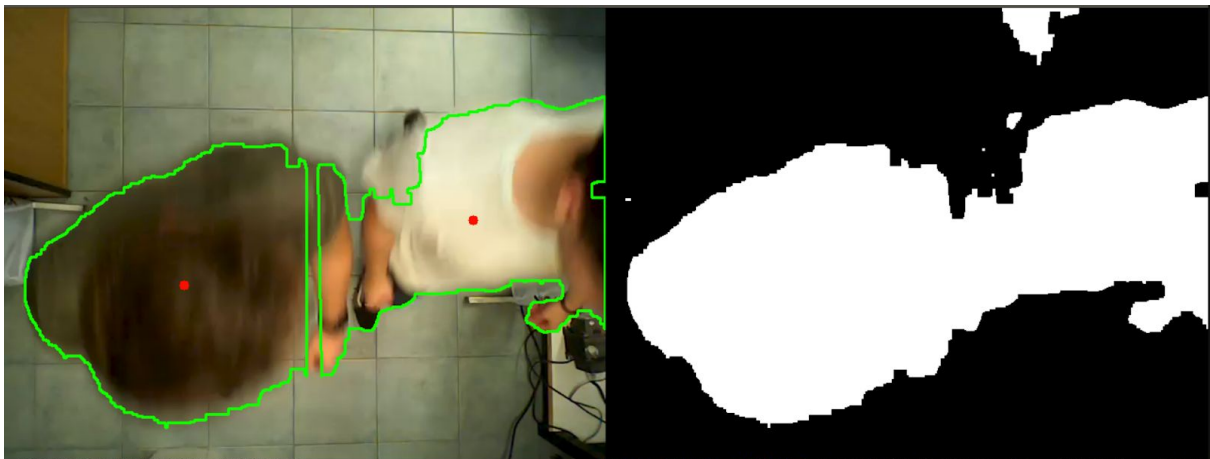


Figura 16: Cálculo de los centroides pertenecientes a los contornos

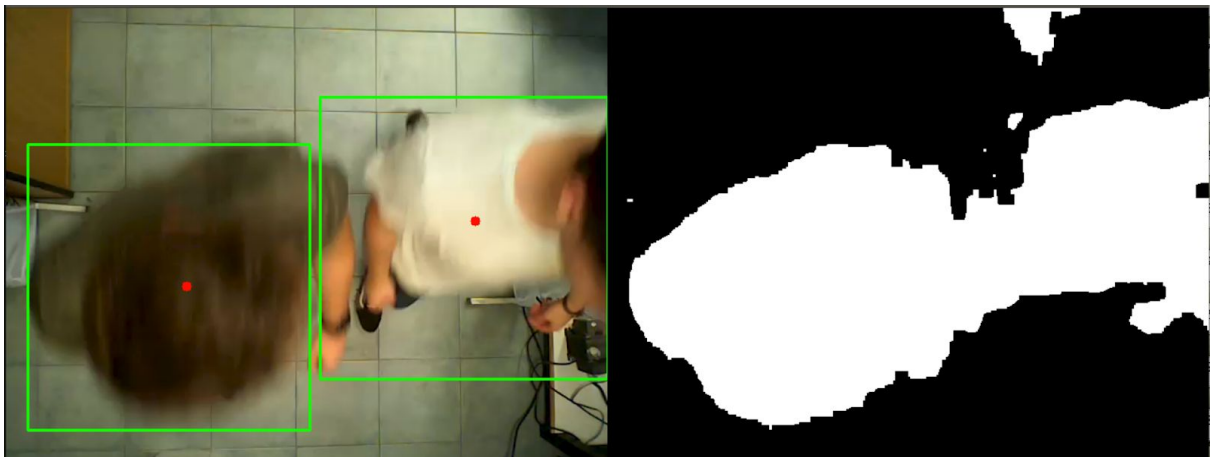


Figura 17: Producto final de la detección de personas. Se esquematiza a cada individuo con un rectángulo cuyo centroide es el calculado en la Fig. 16.

4.3 Seguimiento de personas

4.3.1 Asociación de centroide con persona

Al identificar a todas las personas con un centroide, el seguimiento del recorrido de cada individuo se restringe al seguimiento del movimiento del centroide. A todas las

personas (centroides) se les asigna un *id* e información sobre dónde está en el *frame* actual y dónde estuvo en los *frames* anteriores.

Para lograr un seguimiento de cada individuo, hay que relacionar los centroides en cada *frame* con los del *frame* anterior. Con este fin, a cada centroide en el *frame* actual, se le busca el centroide más cercano en el *frame* anterior, que esté dentro del bounding-box del centroide actual y se asume que pertenece a la misma persona, por lo que se actualiza la posición de la persona. En caso de que haya varias personas detectadas anteriormente dentro del bounding box se toma la de menor distancia. Una situación que se pueda dar es aquella en la que aparecen dos individuos, con sus respectivos centroides, en el *frame* actual, pero un solo individuo y centroide en el *frame* anterior. En este caso, el centroide del nuevo *frame* más cercano al del anterior se identifica como perteneciente a la misma persona y el otro centroide se asume como el de un nuevo individuo que aparece en la imagen.

Otra situación de interés se da cuando el número de centroides o individuos detectados disminuye de un *frame* al siguiente. Una vez que una persona desapareció por cinco frames se asume que abandonó el marco de la imagen y se lo quita de la lista de personas. Se usa este número de cinco frames para mitigar la no detección de la persona en alguna trama. Esta no detección puede deberse a diversos motivos, por ejemplo que el área computada para una persona sea menor que el umbral mínimo prefijado o que dos individuos se hayan acercado mucho en un frame y no se haya podido distinguirlos correctamente. Estos casos son muy pocos en la práctica, pero hemos notado que esta simple regla heurística es efectiva ya que no se llegan a superar los 2 o 3 frames de no detección en el material producido para la evaluación del funcionamiento. Existe otra forma de entender este umbral de cinco frames. Si se asume una tasa baja de frames, por ej., 15 frames por segundo, 5 frames corresponde a $\frac{1}{3}$ segundo. A un paso normal de un peatón de 3 km/h, esto es equivalente a un desplazamiento de algo menos de 30 cm. Es difícil, por tanto, que una persona desaparezca de la escena si se ha movido menos que esta pequeña longitud, siendo más probable que haya habido un problema de detección.

El pseudo-código del seguimiento de personas se encuentra en el Algoritmo 1. La Fig. 18 muestra la asociación de centroides con dos individuos en tres tramas consecutivas.

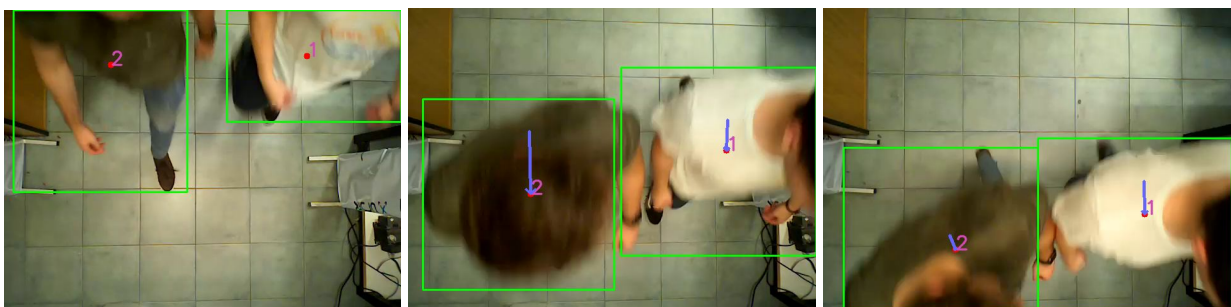


Figura 18: Asociación de centroides a las personas en la escena con respecto al *frame* anterior. Cada individuo se encuentra marcado con un id distinto.

Algoritmo 1

```
personas = []
for frame in frames:
    nuevos_centroides = obtenerCentroides(frame)
    personas_frame_anterior = personas[-1]
    personas_este_frame = []
    for persona in personas_frame_anterior:
        centroide = persona.centroideMasCercano(nuevos_centroides)
        if centroide is Null:
            persona.edad += 1
            continue
        persona.guardarPosicion(centroide)
        persona.edad = 0
        personas_este_frame[] = persona
        nuevos_centroides.remove(centroide)
    if len(nuevos_centroides) > 0:
        for centroide in nuevos_centroides:
            personas_este_frame[] = new Persona(centroide)
    personas_este_frame.filter(p ==> p.edad > 5)
    personas[] = personas_este_frame
```

4.3.2 Determinación de la dirección de cruce

Recordemos que la aplicación típica es la medición de los flujos de entrada y salida de un local comercial. Por lo tanto, no solo es necesario determinar el número de individuos, sino también la dirección de su movimiento. Para saber hacia dónde se dirige una persona se colocaron dos barreras para medir el movimiento en cada sentido. Estas dos barreras sirven para determinar si la persona estuvo en la zona superior o inferior de la imagen, y en qué orden hicieron el viaje.

En cada *frame* se recorren todas las personas presentes, se analiza su posición y se verifica su lugar en la escena con respecto a nuestras barreras lógicas. Si se encuentra por encima de la barrera de **límite inferior** (celeste en las figuras, ver Fig. 19) se marca a la persona como que ingresó por la parte superior. Luego, si en algún frame siguiente se encuentra por debajo de la barrera **principal inferior** (azul en las figuras, ver Fig. 20), se contabilizará que la persona hizo un recorrido hacia abajo en nuestra imagen.

Para contabilizar los recorridos hacia arriba, se procede de manera análoga, pero con las barreras de **límite y principal superiores** (rosa y roja, respectivamente). Así, si un individuo es detectado por debajo de la barrera de límite superior (rosa) y en un frame subsiguiente se encuentra por encima de la barrera principal superior (roja), se asume que se ha dirigido de abajo hacia arriba en la imagen.

La utilización de una barrera límite y otra principal permite evitar contabilizar varios pases, por pequeños errores en la determinación de los centroides, que se darían en caso de utilizar una única barrera. No sólo errores en el cálculo de los centroides pueden generar

falsos cruces si hubiera una barrera única, sino que también habría falsos cruces en el caso que se denominó “guardia de seguridad”. De manera análoga, las barreras límite y principal inferiores son distintas de las superiores para evitar falsos cambios de sentido.

En las Figs. 19-21 se pueden ver los distintos puntos donde se marca a la persona para contabilizar como un cruce hacia abajo.



Figura 19: Las dos personas se encuentran por arriba de la barrera de límite inferior (línea celeste).

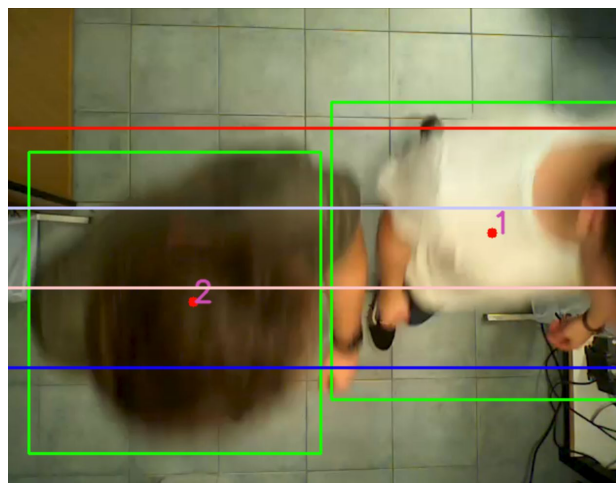


Figura 20: Las dos personas cruzaron la barrera límite inferior.

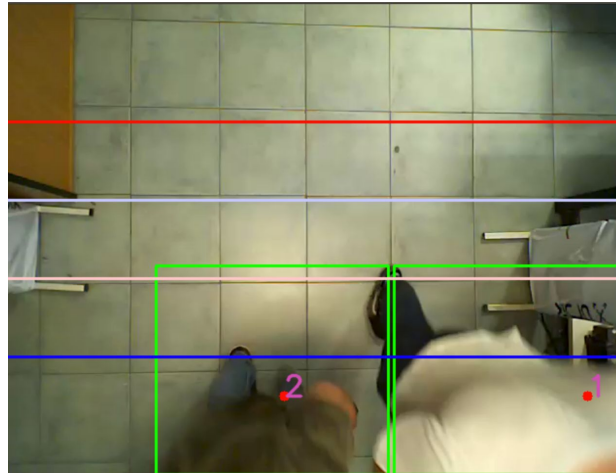


Figura 21: Las dos personas se encuentran debajo de la barrera principal inferior (línea azul). Como fueron registradas por encima de la barrera límite inferior (celeste), se contabilizan como si hubieran hecho el recorrido hacia abajo.

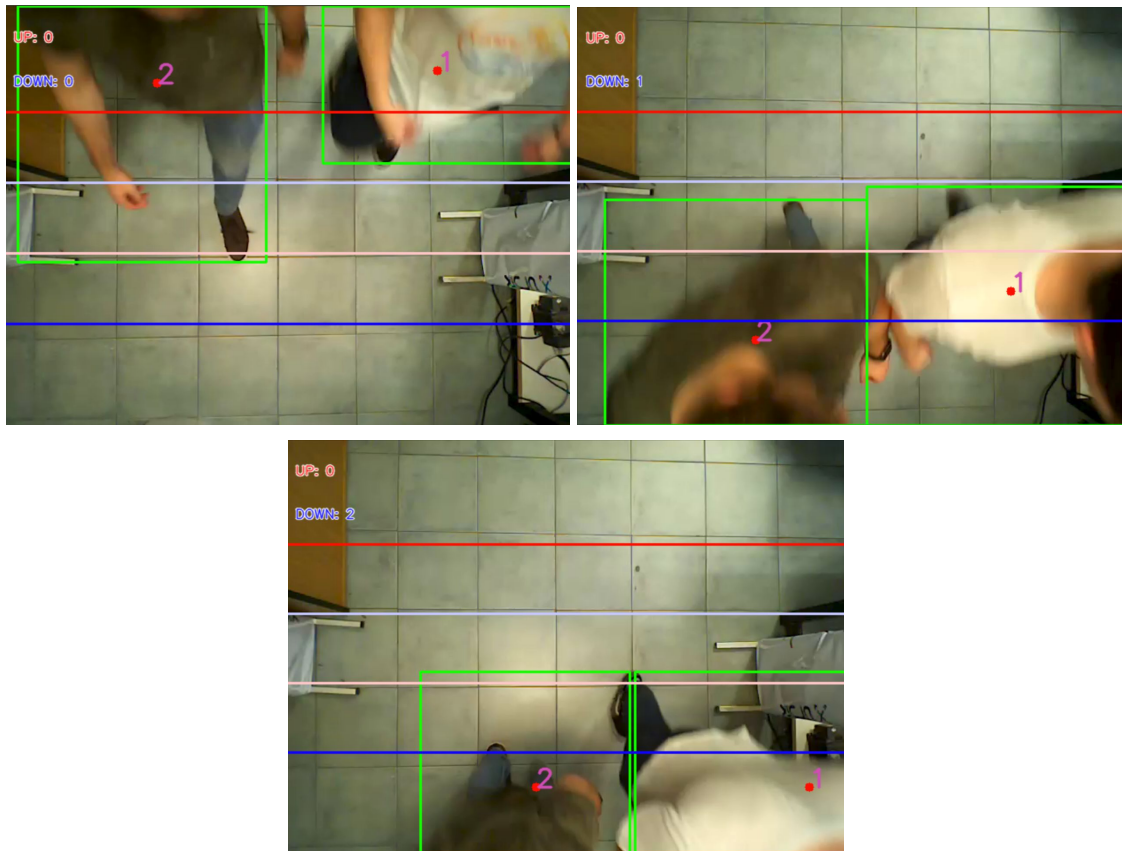


Figura 22: Registro barrera cruzada. En el frame superior izquierdo, hay dos individuos por arriba de la barrera límite inferior (línea celeste). En el frame superior derecho, una de las personas ya pasó la barrera principal inferior (línea azul) y el contador pasos hacia abajo (DOWN) marca 1. En el frame inferior, ambos individuos han atravesado la barrera principal inferior y el contador marca 2.

4.3.3 Registro de barrera cruzada

Basándonos en el sistema de seguimiento de recorrido, se tienen dos contadores, inicializados en cero, que se usan para contar los cruces en las dos direcciones posibles. Por cada cruce registrado en la Sección anterior se incrementa el contador correspondiente. Luego se imprimen estos contadores en la esquina superior izquierda del frame para tener una confirmación visual. La Fig. 22 muestra el funcionamiento de estos contadores.

Según la configuración de la ejecución del programa, la información de cuántas personas pasaron y en qué dirección se puede guardar en un archivo de texto en un formato *ad-hoc* .

5. Ajustes de parámetros

Debido a la diversidad de contextos en los que puede ser instalado nuestro sistema, nos fue necesario parametrizar ciertas propiedades para asegurar la flexibilidad del mismo y así lograr que con unos simples ajustes el programa pueda ser utilizado en distintos entornos. Cuando hablamos de los contextos en los cuales podría ser instalado nuestro programa, consideremos los siguientes casos:

- La altura de la cámara con respecto al piso: Cuanto más alta esté, más pequeñas se verán las personas en relación al tamaño total de la imagen y más recorrido tendrán que hacer los individuos para atravesar toda el área capturada.
- La calidad de la cámara: Las diferentes resoluciones darán distintas proporciones en píxeles de las personas, y también afectará el tamaño de estas.

Considerando estas dos variables decidimos tomar unas heurísticas para cada escenario; como ya pudimos ver en las figuras que usamos de ejemplo, tomamos vídeos en un laboratorio del *ITBA*, y en la casa de uno de los autores del trabajo para asegurarnos la diversidad de casos de uso y la robustez del programa, y cada uno de estos entornos tiene su propio *set* de parámetros que hace que el programa funcione correctamente. Los mismos se detallan a continuación.

5.1 Umbrales para la detección de una persona

Según la cámara que se use y la ubicación de la misma, se deben ajustar los umbrales de área mínima y máxima para considerar un objeto en movimiento como un individuo. Recordemos que a los objetos con áreas que son demasiado pequeñas para ser una persona se las descarta, y a las áreas que son demasiado grandes se las divide en dos según sus dimensiones.

El incorrecto ajuste de estos parámetros puede llevar a confundir cambios en la iluminación fuertes como personas (por ej., si alguien abre una puerta y entra un haz de luz), o a dos personas que caminan juntas como si fuese sólo una.

5.2 Barreras de cruce

En la Sección “*Seguimiento de personas*” mencionamos el uso de unas barreras lógicas para detectar el cruce de una persona y determinar su dirección. Las posiciones de estas barreras pueden ser cambiadas. El incorrecto ajuste de estos parámetros puede llevar a contar de menos por no detectar que la persona efectivamente estuvo de los dos lados de la barrera, o puede causar que se haga un doble conteo de la persona si las barreras de límite y principal (los pares azul-celeste y rojo-rosa en las Figs. 19-21) están muy cercanas una de la otra.

Un claro ejemplo donde la correcta parametrización de las líneas de cruce es vital, es en el caso del “guardia de seguridad”: el mismo puede caminar horizontalmente a lo

largo de la escena pero no debe ser contabilizado como yendo en ningún sentido determinado.

También se encontraron casos en los que una persona da zancadas largas al pasar por el área vigilada, por lo que se puede registrar el centroide de la persona en un *frame* por debajo (o por arriba) de donde estaba en el *frame* anterior, causando un movimiento del centroide en sentido contrario a la dirección en la cual se estaba moviendo la persona. En los casos en los cuales sucede sobre una de las barreras podría contarse como si la persona estuviese dirigiéndose en una dirección incorrecta.

5.3 Configuración

Todos estos parámetros son configurables en nuestro proyecto mediante archivos en formato *.json* alojados en el directorio del proyecto en la carpeta *./config/*.

Los parámetros son:

Áreas

Se tienen dos parámetros que se usan para delimitar el área posible de una persona:

- `maximum area threshold` (**área mínima**): se usa para descartar objetos detectados muy pequeños. Valor entre 0 y 1 que se multiplica por el área total del frame.
- `minimum area threshold` (**área máxima**): se usa para dividir contornos muy grandes. Valor entre 0 y 1 que se multiplica por el área total del frame.

Líneas

- `thickness` (**Espesor de la línea**): Distancia entre la barrera límite y la barrera principal. Se usa la misma distancia ya sea para el par de barreras que hacen el conteo hacia arriba o el par hacia abajo. Valor entre 0 y 1 que se multiplica por la altura del frame.
- `line up` (**Barrera hacia arriba**): Posición de la barrera límite superior que hace el conteo hacia arriba. Valor entre 0 y 1 que luego se multiplica por la altura del frame.
- `line down` (**Barrera hacia abajo**): Posición de la barrera límite inferior que hace el conteo hacia abajo. Valor entre 0 y 1 que luego se multiplica por la altura del frame.

La manera de configurar los parámetros es asignando un numerador y denominador. Esto se debe a que no trabajamos con valores absolutos sino con valores en relación al tamaño total de la imagen. Por ejemplo, para la máxima área que se espera que ocupe un individuo, se define un numerador y un denominador y el sistema calcula el área absoluta como

$$\text{maximum area threshold} = (\text{Ancho de la imagen}) \times (\text{Altura de la imagen}) \times \frac{\text{numerador}}{\text{denominador}}$$

A continuación mostramos el ejemplo del archivo de configuración para los videos del laboratorio:

```
{
  "areas": {
    "maximum_area_threshold": {
      "numerator": 1,
      "denominator": 8
    },
    "minimum_area_threshold": {
      "numerator": 1,
      "denominator": 20
    }
  },
  "lines": {
    "thickness": {
      "numerator": 1,
      "denominator": 5
    },
    "line_up": {
      "numerator": 4,
      "denominator": 9
    },
    "line_down": {
      "numerator": 5,
      "denominator": 9
    }
  }
}
```

6. Resultados

Como fue mencionado anteriormente, para analizar el comportamiento del sistema se realizaron experimentos en dos escenarios distintos: uno en el laboratorio y otro hecho de forma casera. En esta sección se analiza el funcionamiento del sistema en cada uno de los casos planteados en la Sección 3. Es necesario destacar que se realizaron experimentos que fueron especialmente planeados para representar cada una de las situaciones planteadas en dicha sección.

Los resultados producidos por nuestro trabajo son correctos en todos los entornos en los que fue probado, y logramos solucionar los problemas que nos propusimos con nuestros casos especiales. Contamos con trece minutos de rodaje en total, de los cuales dos minutos corresponden a videos grabados en el Laboratorio de Optoelectrónica, y el resto en la casa de uno de los desarrolladores. Además contamos con nuestras pruebas hechas en vivo, alimentando al algoritmo con un *feed* de video en tiempo real donde también se observó un correcto funcionamiento.

6.1 Dos individuos que cruzan juntos la barrera

En el caso que dos individuos cruzan juntos la barrera y no estén en contacto uno con el otro ni a una distancia muy pequeña el programa no presenta grandes dificultades. Un ejemplo de este caso se observa en la Fig. 24.

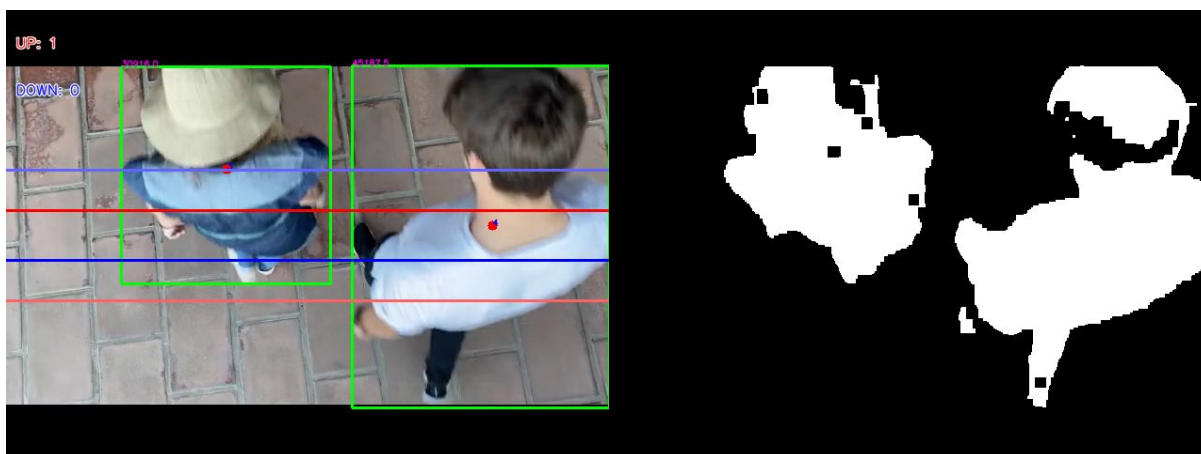


Figura 23: Dos individuos cruzando el portal a una distancia considerable

Cuando las dos personas, sea que se dirigen en la misma dirección o no, están en contacto uno con el otro o la distancia entre ellos es muy corta, sí se encontraron algunas dificultades. Como se observa en la Fig. 25, en este caso el detector de fondo agrupa a ambos individuos como un único objeto, por lo que al hacer la detección de contornos sólo hay un único contorno detectado cuando tendría que haber dos. Para entender las consecuencias de este problema, tomemos como ejemplo el video de ~2 minutos registrado

en el Laboratorio de Optoelectrónica. En este caso, hay un total de 11 cruces en cada dirección. Sin embargo, sólo se detectaron 8 hacia arriba y 8 hacia abajo.

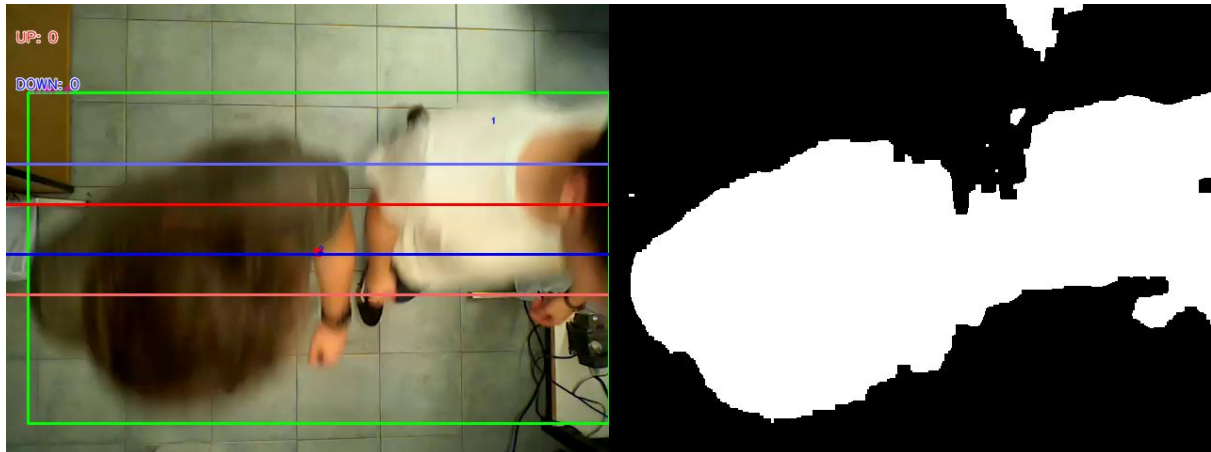


Figura 24: Dos individuos cruzando el portal en gran cercanía uno del otro, sin división de contornos

Para poder resolver este problema, se limitaron los posibles tamaños que puede tener un contorno que representa a una persona. Usando los parámetros de área mínima y máxima de una persona de tal manera que si el área de un contorno está por arriba de el área máxima el mismo es dividido como fue explicado en la Sección 4.2.3 (ver Fig. 26). Esta simple metodología permitió contabilizar perfectamente los cruces en todos los videos de prueba.

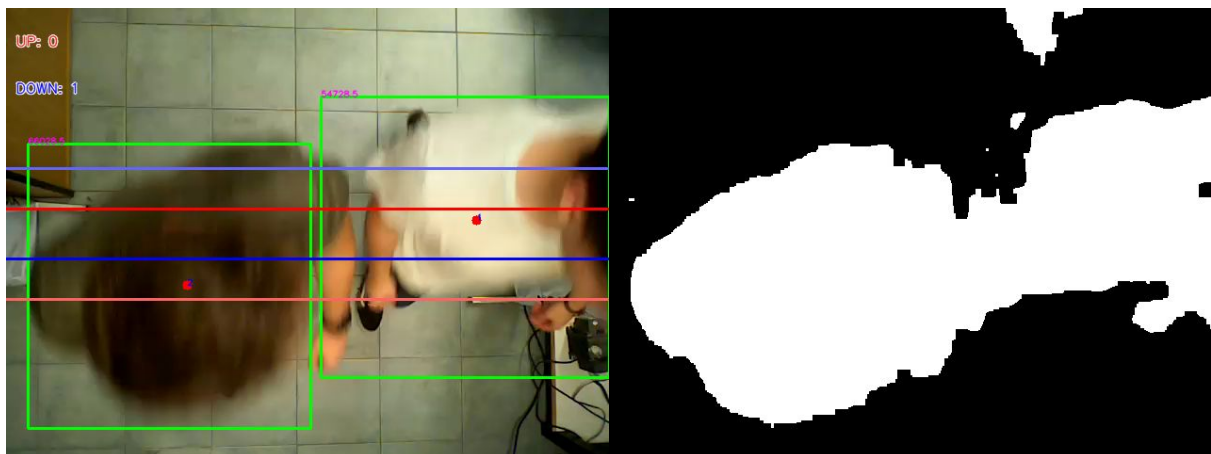


Figura 25: El mismo *frame* que la Fig. 24, pero con la división de contornos activada.

Se tomó el caso del laboratorio y se hizo un análisis de la eficacia del programa según la parametrización de los tamaños esperados de las personas. Se compararon los resultados de conteos hacia arriba y hacia abajo los cuales se muestran en la Tabla 1. En la primera ejecución se modificó el tamaño máximo que puede ocupar un único individuo de manera tal que nunca se realice la división del contorno, y en la siguiente corrida se utilizó un tamaño máximo adecuado correspondiente al escenario del video.

Conteos	Parametrización Incorrecta	Parametrización Correcta	Resultado real
Arriba	8	11	11
Abajo	8	11	11

Tabla 1: Comparación de los resultados del programa según la parametrización utilizada

6.2 Individuos con mochilas o equipajes

En la situación en la que los individuos cruzan con mochilas o equipajes no hubieron mayores complicaciones gracias al diseño de la aplicación. Como se mencionó anteriormente se hace uso de los parámetros de área mínima y área máxima de una persona. Estos parámetros contemplan a individuos con mochilas o equipajes por lo que a la hora de hacer pruebas se pudo detectar correctamente a los individuos y equipajes como una sola persona. La Fig. 27 muestra un ejemplo de la correcta detección de un individuo y su equipaje como una sola unidad a ser contabilizada.

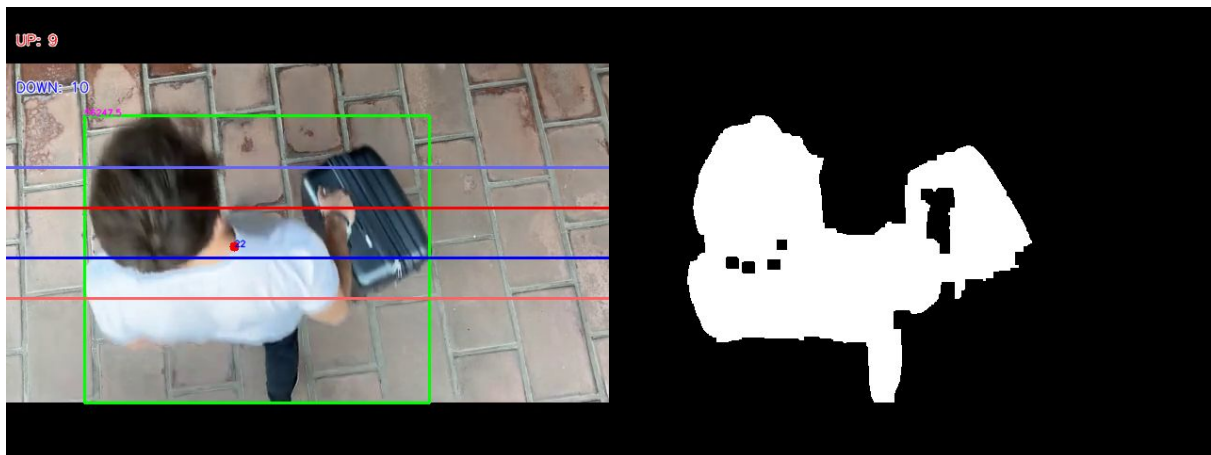


Figura 26: Individuo cruzando el portal rodando un equipaje de mano.

Es importante notar también que una incorrecta configuración de las áreas mínimas y máximas, puede llevar a un mal conteo y mal funcionamiento del programa. Si se configura el área máxima de una persona demasiado chica, el algoritmo dividirá a la persona en dos de forma incorrecta. Por lo que es importante poner un límite máximo de dos personas donde sea seguro decir que no es el área de una sola persona. En la Fig. 28 se puede observar un caso en el que la mala configuración de los parámetros del programa llevan a la incorrecta consideración de una pieza de equipaje como un individuo más en la escena

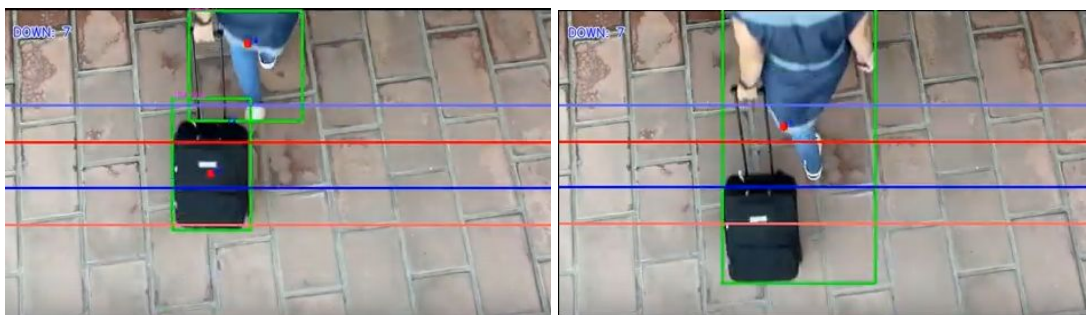


Figura 27: Resultados con parámetros mal configurados a la izquierda, y bien configurados a la derecha.

Se usó un video de prueba para comprar una parametrización correcta de la división contornos contra otra incorrecta. En este caso, los parámetros variados fueron tanto el tamaño máximo como mínimo esperado de una persona, esperando causar una división de los contornos de las personas con su equipaje cuando excedan el tamaño máximo, y que interprete al equipaje como un individuo propio y no sea filtrado como un contorno más pequeño que el mínimo.

Conteos	Parametrización Incorrecta	Parametrización Correcta	Resultado real
Arriba	11	10	10
Abajo	10	10	10

Tabla 2: Comparación de la eficacia del programa según la parametrización utilizada

6.3 Guardia de Seguridad

Recordemos que el caso del “Guardia de Seguridad” es aquel en que una persona camina en la zona de visión de la cámara, pero no debe ser contabilizada. En este caso se pueden mitigar los conteos indeseados ajustando las barreras de cruce. Como se describió anteriormente, tenemos dos pares de barreras distintas, uno para contar los cruces hacia arriba y otras dos para contar los cruces hacia abajo. Las barreras de cada par (celeste-azul o rosa-rojo en las Figs. 19-21) se encuentran a una distancia parametrizable: un individuo es contado sólo si cruza ambas barreras. Esta separación genera “zonas muertas” en las que se puede caminar tranquilamente hacia arriba o abajo sin disparar conteos indebidos. De esta forma, se pueden ajustar las líneas de cruce para que estas zonas muertas coincidan con aquellas en las que se desplaza habitualmente el guardia de tal modo que queden dentro de esta zona en la que no se harán conteos indebidos.

Las Figs. 28 y 29 son tomadas de los experimentos realizados para probar el funcionamiento en este caso: un individuo caminando de izquierda a derecha en la escena, pero sin atravesarla verticalmente de manera definida. La Fig. 28 muestra el caso en el cual ambas barreras coinciden, mientras que la Fig. 29 se corresponde con el caso en el que las barreras están separadas. El análisis del video muestra que, cuando las barreras coinciden, se contabilizan diez cruces, pero ningún cruce es contabilizado cuando las barreras son

separadas por una distancia adecuada como en la Fig. 29. Es importante destacar que la parametrización de las líneas no necesita ser demasiado precisa para poder llegar al resultado óptimo, y tan sólo el hecho de tener una distancia razonable entre barreras mitiga la gran parte de los conteos indebidos.

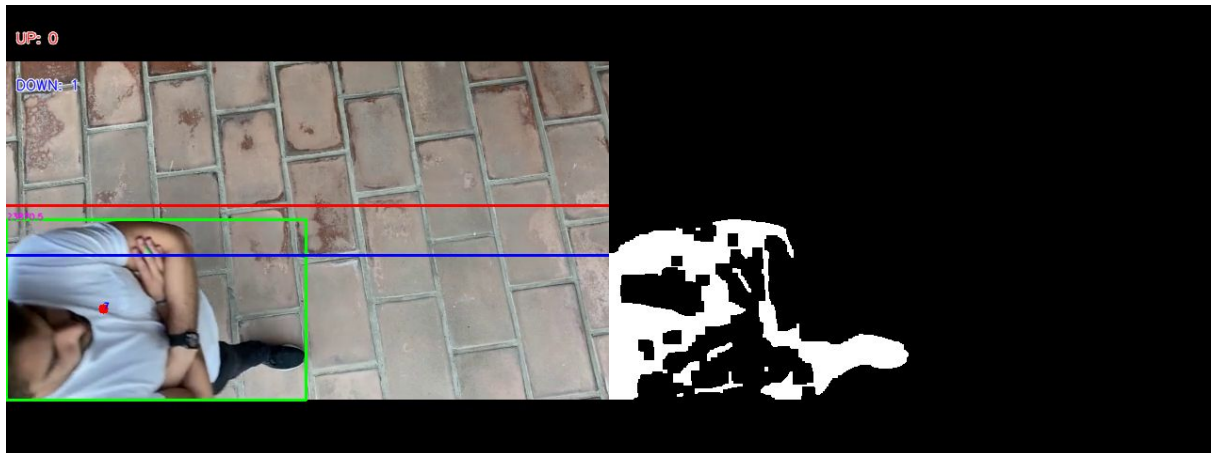


Figura 28: Guardia de seguridad cruzando de izquierda a derecha sin las barreras límite

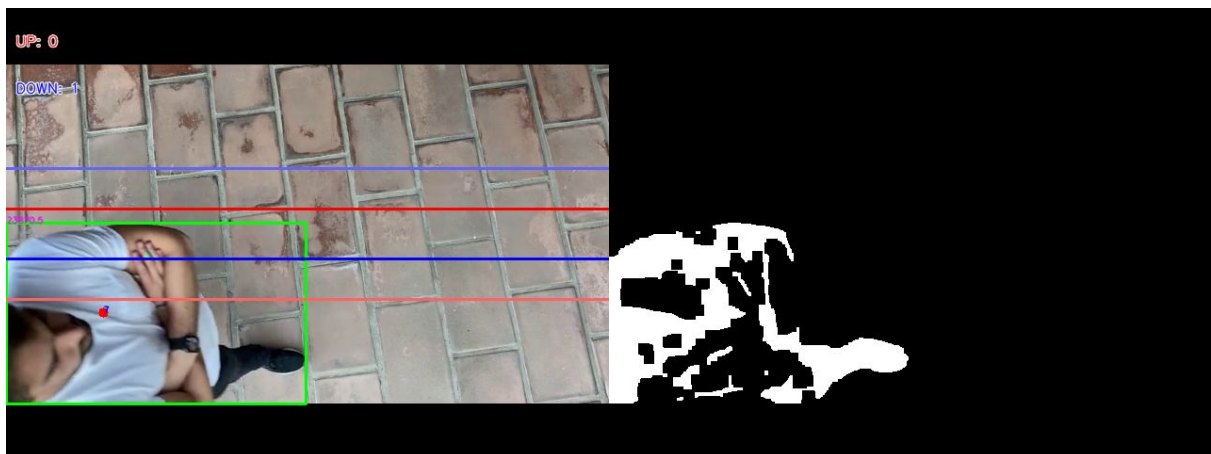


Figura 29: Guardia de seguridad cruzando de izquierda a derecha con las barreras límite

Inicialmente se implementó un sistema de barreras simple donde solo se comparaba la posición del individuo en el *frame* anterior con el actual, y al detectarse el cruce de una barrera se contabilizaba como un viaje en una dirección determinada. El caso del Guardia de Seguridad expone las debilidades de este sistema, por lo que se debió implementar una solución más robusta. A fin de demostrar la mejora introducida por las *barreras límite*, se comparan los resultados del programa con el sistema de barreras simples con los resultados obtenidos con barreras compuestas en la Tabla 3.

Conteos	Barreras simples	Barreras compuestas	Resultado real
Arriba	4	0	0
Abajo	5	1	1

Tabla 3: Comparación de los resultados del programa una vez implementado el sistema de barreras apropiado

6.4 Cambios de iluminación

Se busca que el programa sea tolerante a los cambios que pueda haber en la luz natural a lo largo del día. Nuestro sistema contempla esta posibilidad implícitamente debido a la elección del algoritmo de detección de fondos, el cual funciona en base a un historial de los *frames* más recientes, permitiendo el cambio gradual del escenario a lo largo del tiempo sin provocar desajustes en el mismo.

Para comprobar el adecuado funcionamiento del sistema en este caso, se realizaron pruebas en el mismo escenario, pero con distinta iluminación. Por un lado, se tomó un video poco pasado el mediodía con condiciones de luz óptimas (ver Fig. 30). Por el otro, se realizó un video entrando la noche, con luz reducida (ver Fig. 31). Para ambos casos se mantuvieron exactamente los mismos parámetros de configuración. De esta manera, se simuló que ambos videos fueran uno solo. Los resultados fueron excelentes. En efecto, no se observaron dificultades y se contabilizaron todos los cruces de forma correcta.

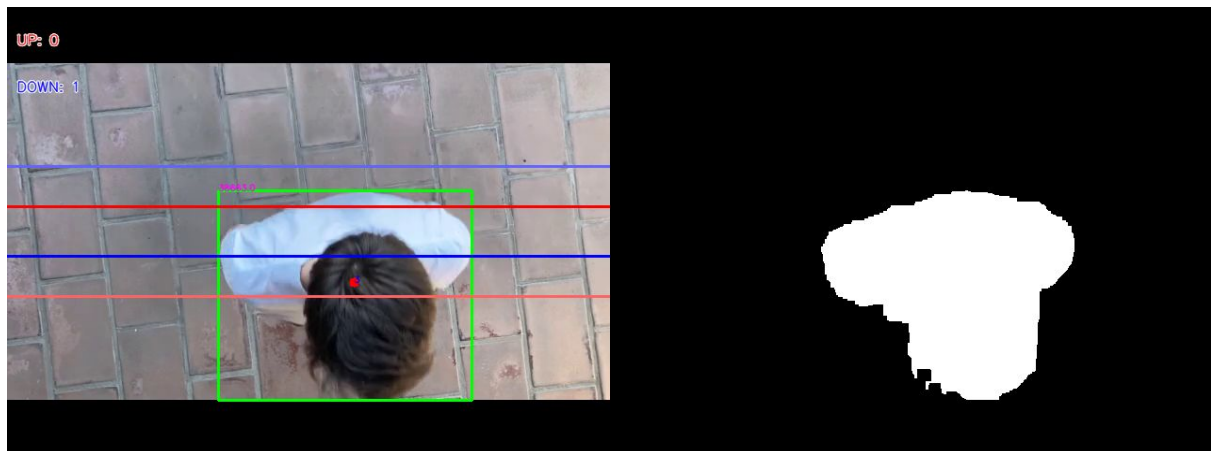


Figura 30: Persona cruzando el portal con una alta iluminación natural, poco después del mediodía.

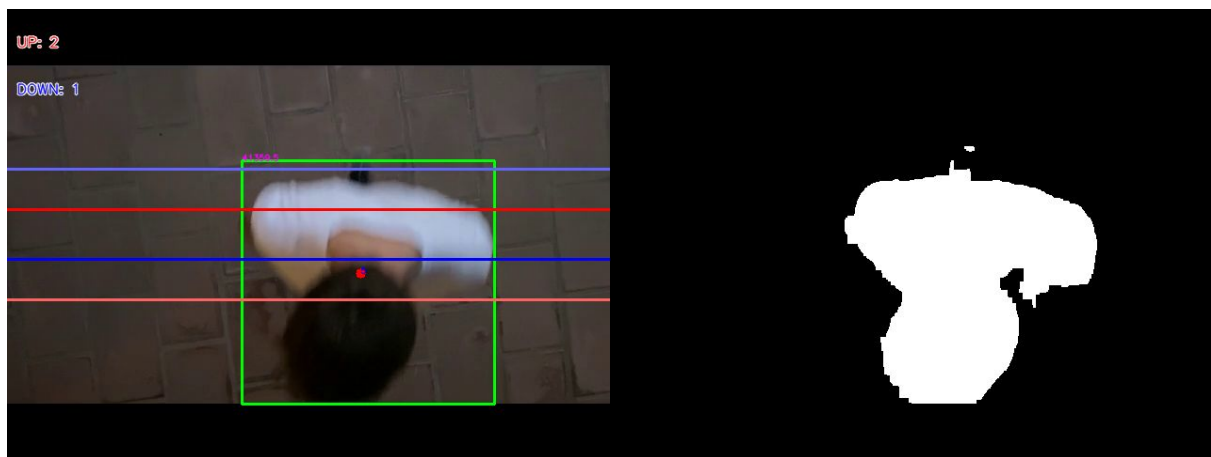


Figura 31: Persona cruzando el portal con poca iluminación, poco antes de anochecer.

6.5 Resolución de la imagen

Nuestra parametrización del programa permite configurar al algoritmo en base a los valores relativos de la imagen, por lo que los cambios en resolución y *ratio* entre altura y ancho son considerados automáticamente.

Los experimentos realizados en el domicilio de uno de los autores se basaron en filmaciones con un teléfono con alta definición. Para todas las pruebas presentadas hasta aquí, se bajó la calidad del video para asemejarla a aquella producida por una cámara web. En este sentido, se utilizó una resolución de 640x480 y la compresión fijada de manera de alcanzar un *bitrate* de 1200 kbps. Aún con esta baja calidad se obtuvieron resultados correctos, tal como se ha descrito en las secciones anteriores.

Dado que el sistema funciona con videos de baja calidad, puede cuestionarse si funciona adecuadamente con videos de mayor calidad. Para ello, se ajustó el *bitrate* a 6000 kbps. Sin realizar ajustes en el sistema, esto es, en los parámetros presentados en la Sección 5, se obtuvieron resultados correctos, demostrando la robustez y versatilidad del sistema.

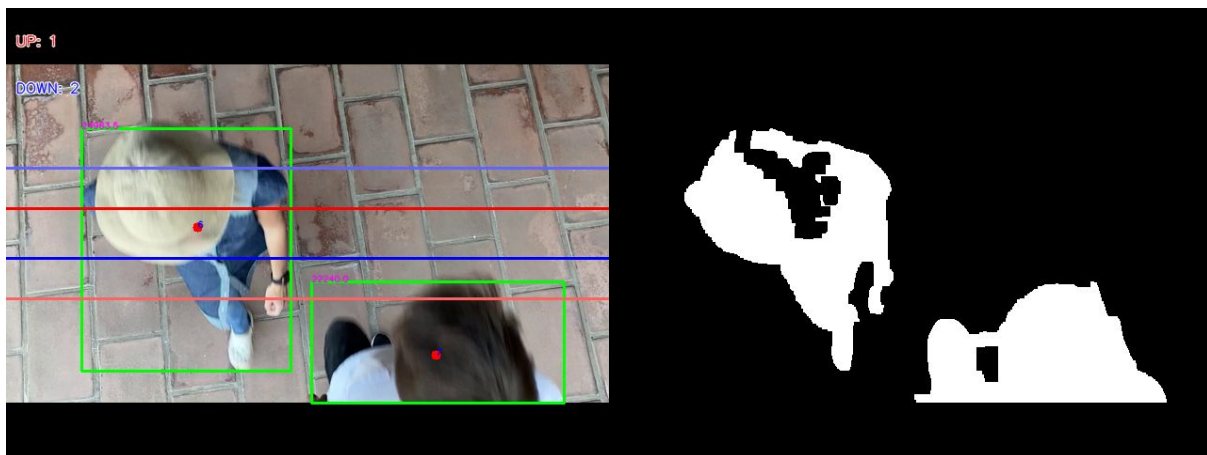


Figura 32: Captura de uno de los videos convertidos a una mayor resolución y con menor compresión.

7. Conclusión

El conteo de personas es un problema desafiante sobre el cual siguen publicándose nuevas técnicas y dispositivos que lo solucionan, incorporando distintos sensores y maneras de reconstruir las escenas para contabilizar el flujo de personas en un área determinada. Si bien las soluciones comerciales suelen utilizar algoritmos de Aprendizaje Automático, demostramos que es posible obtener una solución confiable con técnicas de análisis y tratamiento de imágenes tradicionales sin necesidad de sensores especiales ni cámaras sofisticadas.

Se lograron cumplir los objetivos propuestos. En primer lugar, se desarrolló el sistema de conteo haciendo uso de dispositivos de bajo costo. En segundo lugar, se demostró la robustez del sistema en los casos difíciles que se habían propuesto. La demostración se basó en el análisis del funcionamiento del sistema en experimentos especialmente diseñados para ser representativos de dichos casos. El éxito obtenido fue el resultado de una adecuada selección de los algoritmos de procesamiento de imágenes y la introducción de ciertos parámetros que permiten adaptar el sistema a distintos escenarios.

Todo esto fue logrado gracias a un adecuado análisis del panorama actual en la tecnología de conteo de personas, y a una rápida iteración de ideas sobre cómo afrontar los distintos obstáculos. Nuestro propósito también era lograr un algoritmo lo suficientemente liviano como para poder ejecutarlo en una *single board computer*, y nuestras pruebas hechas con una Raspberry Pi 3 fueron exitosas, siendo ésta capaz de analizar los videos en tiempo real.

Aún así, nos hubiese gustado poder realizar un análisis sobre una mayor variedad de escenarios, usando un portal con más flujo de personas y poder dejar el algoritmo ejecutándose por varios días y verificar el correcto funcionamiento a lo largo del tiempo, ya que debido a las circunstancias actuales tuvimos que recurrir a escenarios artificiales .

8. Apéndice

8.1 Uso del programa

8.1.1 Instalación

Nuestro programa está escrito en el lenguaje de programación *python3*, por lo que será indispensable para correrlo tenerlo instalado. Una vez descargado nuestro [repositorio](#) [15] podrán instalar las dependencias de nuestro programa con el comando

```
$ pip3 install -r dependencies.txt
```

Si no se cuenta con el programa *pip3*, se puede instalar o buscar en el archivo *dependencies.txt* cuáles son, y encontrar la manera de instalarlas. Luego, el repositorio cuenta con varios videos de ejemplo en la carpeta *./Test Files/*.

8.1.2 Ejecución por línea de comandos

El sistema se puede ejecutar por línea de comandos escribiendo

```
$ python3 people_counter.py
```

El programa acepta distintos parámetros por consola para alterar la ejecución, aquí una descripción de las distintas opciones:

- h, --help show this help message and exit
- video *string* : *Path* al video que se desea analizar
- show *boolean* : Mostrar el video en tiempo real mientras se analiza. Se puede apagar para un menor tiempo de ejecución.
- save *boolean*: Guardar las *frames* del video a medida que se ejecuta. Se puede apagar para un menor tiempo de ejecución.
- print_logs *boolean*: Imprimir por salida estándar los logs a medida que el video va siendo procesado. Se puede apagar para un menor tiempo de ejecución.
- live *bool*: Obtener el *feed* del video por la webcam en tiempo real en vez de cargar un video en memoria. Ignora --video.
- config_path *string*: *Path* al archivo de configuración correspondiente al video.
- clean *bool*: Si es verdadero, el programa limpia todos los archivos generados en *./out/** y termina la ejecución

Todos estos parámetros tienen un valor *default* y no es necesario nombrarlos a todos cada vez que se ejecuta el programa. Aquí un ejemplo de cómo se vería correr el programa:

```
$ python3 start_no_gui.py --save=True  
--config_path="./config/config_laboratorio.json" --video="./Test  
Files/TestVideo.avi"
```

8.1.3 Ejecución por interfaz gráfica

Se realizó una interfaz gráfica con el objetivo de poder tener una herramienta más visual para poder hacer la configuración de parámetros de líneas de cruce, y ajustar las áreas mínimas y máximas de una persona. De este modo al usar un video nuevo es más fácil y rápido poder probar distintas configuraciones y tener una guía visual de donde se van a colocar las líneas

Para poder correr el programa por medio de la gui se debe correr el script *gui.py*

```
$ python3 gui.py
```

El programa abrirá una ventana pidiendo seleccionar el archivo de video a analizar y una vez seleccionado se mostrará una ventana para configurar los parámetros de ejecución (ver Fig. 33).

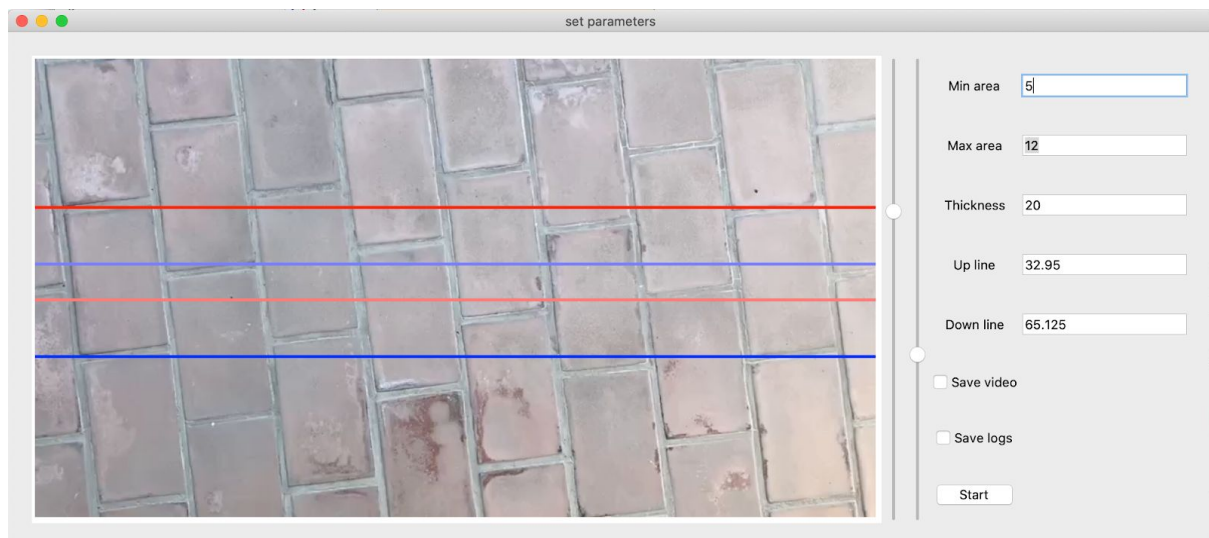


Figura 33: Interfaz gráfica

Como esta interfaz es para testeo y no para el uso real, siempre se muestra el feed de video analizando el mismo. Los parámetros a configurar son los siguientes:

- **Min area:** Porcentaje del total del área de un frame de video el cual se usa para determinar el área mínima de una persona (número entre 0 - 100).
- **Max área:** Porcentaje del total del área de un frame de video el cual se usa para determinar el área máxima de una persona (número entre 0 - 100).
- **Up line:** Porcentaje del alto de un frame de video el cual se usa para determinar

dónde se coloca la barrera límite que contabiliza una persona cruzó la barrera en dirección hacia arriba (número entre 0 - 100).

- **Down line:** Porcentaje del alto de un frame de video el cual se usa para determinar dónde se coloca la barrera límite que contabiliza una persona cruzó la barrera en dirección hacia abajo (número entre 0 - 100).
- **Thickness:** Porcentaje del alto de un frame de video el cual se usa para determinar la distancia entre las barreras límite y principal de cruce (número entre 0 - 100).

Referencias

- [1] La Nación, "Cinco días después del horror: las transgresiones que permitieron la tragedia. En una noche se violaron nueve normas". Online: <https://www.lanacion.com.ar/sociedad/en-una-noche-se-violaron-nueve-normas-nid668625/>. Accedido 16/11/2020.
- [2] La Nación, "El Indio Solari en Olavarría: la impresionante foto que muestra la cantidad de gente que asistió al recital". Online: <https://www.lanacion.com.ar/sociedad/el-indio-solari-en-olavarria-la-impresionante-foto-que-muestra-la-cantidad-de-gente-que-asistio-al-recital-nid1993260/>. Accedido: 16/11/2020.
- [3] La Gaceta de Salta, "Sin patios de comidas y con capacidad limitada reabren galerías y shopping". Online: <https://www.lagacetasalta.com.ar/nota/136553/actualidad/sin-patios-comidas-capacidad-limitada-reabren-galerias-shopping.html>. Accedido 16/11/2020.
- [4] C. Raghavachariy otros, "A Comparative Study of Vision Based Human Detection Techniques in People Counting Applications", *Procedia Computer Science* 58, 461-469 (2015).
- [5] D. Roqueiro y V. A. Petrushin, "Counting People using Video Cameras", *International Journal of Parallel, Emergent and Distributed Systems* 22 (3), 193-209 (2007).
- [6] J-W. Kim y otros, "Real-time Vision-based People Counting System for the Security Door", en *Proceedings of the IEEK Conference 2002*, 1416-1419 (2002).
- [7] S. Yoshinaga, A. Shimada y R. Taniguchi, "Real-time people counting using blob descriptor", *Procedia - Social and Behavioral Sciences* 2 (1), 143-152 (2010).
- [8] V-Count, "Ultima AI. Precise. All-in-one. Plug&Play". Online: <https://v-count.com/products/ultima-ai-people-counting-tracking-sensor/>. Accedido: 16/11/2020.
- [9] Xovis, "PC2S". Online: <https://www.xovis.com/en/products/detail/pc2s/>. Accedido 16/11/2020.
- [10] MegaCount, "3D People Counter". Online: <https://www.megacount.io/stereo>. Accedido: 16/11/2020.
- [11] M. Sinecen, "Digital Image Processing with MATLAB", en "Applications from Engineering with MATLAB Concepts", Jan Valdman editor, IntechOpen (2016).
- [12] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction", en *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, 28-31 (2004)
- [13] Z. Zivkovic y F. van der Heijdenb, "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction", *Pattern Recognition Letters* 27 (7), 773-780 (2006).
- [14] S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Pub (1997).
- [15] S. Suzuki y K. Abe, "Topological structural analysis of digitized binary images by border following", *Computer Vision, Graphics, and Image Processing* 30 (1), 32-46 (1985).
- [16] <https://bitbucket.org/itba/conteo-umbral/>