UNIVERSIDAD CARLOS III DE MADRID

INGENIERÍA INFORMÁTICA

Universidad
Carlos III de Madrid

Aplicación de técnicas de visión artificial para la seguridad en el acceso a sucursales bancarias

Autor: Álvaro Queipo de Llano Escribano

Tutor: José Manuel Molina López

Fecha: Julio 2014

Resumen

Debido a la rápida evolución de la tecnología, hoy en día tenemos muchísimas aplicaciones para facilitarnos el día a día, sin embargo esto es una hoja de doble filo, porque nos pueden ayudar a ejecutar nuestras tareas diarias con fines lícitos como delictivos. Es por esto por lo que es importante que las técnicas de seguridad evolucionen al mismo ritmo, ofreciéndonos una protección acorde con los avances.

Este proyecto pretende diseñar una solución para el control de accesos en sucursales bancarias que pueda ofrecer una seguridad preventiva y automática, facilitando el trabajo y otorgando seguridad adicional a los empleados y clientes de dichas sucursales.

Agradecimientos

Quiero agradecer en primer lugar a mi tutor de proyecto, el Profesor Don José Manuel Molina López, por haberme dado la oportunidad de desarrollar este proyecto bajo su supervisión y haber estado apoyándome durante todo el proceso de elaboración del mismo.

También quiero agradecer a mi empresa y compañeros, Ganetec Global Solutions, por haberme permitido disponer algo de tiempo para realizar este proyecto y apoyándome a través de un esfuerzo extra durante el tiempo que he necesitado para realizar este trabajo.

En especial, quiero agradecer el apoyo recibido por mi guía, consejero y amigo, Enrique, quien sin su apoyo y ánimo no podría haber acometido este proyecto.

Índice

Índice de Ilustraciones

Índice de Diagramas

Introducción

The video surveillance systems exist for 25 years. They began as analog systems to 100% and gradually were digitized. Systems today have come a long way since the advent of the first tube analog cameras connected to VCR.

La video vigilancia lleva existiendo desde hace 25 años. Comenzó como sistemas analógicos, y poco a poco se ha ido trasformando en 100% digitales.

Actualmente, estos sistemas están compuestos de cámaras y servidores, dedicados a la grabación de video digital. Aunque se tiende al mundo 100% digital, siguen existiendo sistemas mixtos en los que nos encontramos la unión entre el mundo analógico y digital.

1. Historia: Evolución de la video vigilancia

I. Sistemas de circuito cerrado de TV analógicos usando VCR

Un sistema CCTV (circuito cerrado de TV) analógico está formado por cámaras analógicas conectadas por un cableado coaxial a un VCR (grabador de vídeo) y a un monitor analógico, con el fin de poder visualizar y grabar en tiempo real. Según la escala de la instalación, pude ser necesario el uso de un multiplexor, que permite grabar múltiples cámaras simultáneamente.

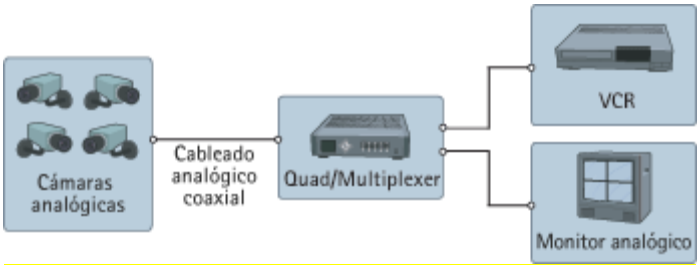


Ilustración 1: Esquema sistema analógico

Las grabaciones realizadas por el VCR se almacenan en cintas, sin posibilidad de aplicar ninguna técnica de compresión, limitando esto la autonomía del sistema y debiendo cambiar la cinta cada 8h como máximo.

II. Sistemas de circuito cerrado de TV analógicos usando DVR

Como comentábamos con anterioridad, existen sistemas mixtos, este es un ejemplo de ellos. Este sistema utiliza cámaras analógicas conectadas directamente (internamente integran un multiplexor) a un DVR (video grabador digital), permitiendo así pasar la imagen de analógico a digital en el mismo momento de la grabación. Estas grabaciones ya permiten realizar compresión y son almacenadas en un disco duro, lo que nos permite el almacenamiento de mayor tiempo, y por ende, mayor autonomía del sistema.

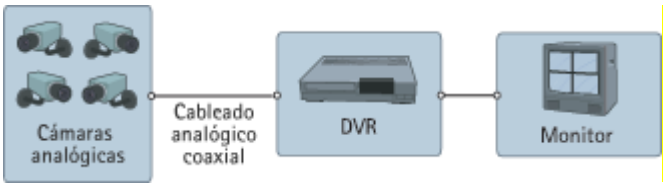


Ilustración 2: Esquema sistema analógico usando DVR

Para poder realizar visualización en tiempo real, en este sistema sigue siendo necesaria la conexión de un monitor analógico, ya que el DVR únicamente hace la conversión analógico-digital para la grabación.

El sistema DVR añade las siguientes ventajas:

- No es necesario cambiar las cintas
- Calidad de imagen constante

III. Sistemas de circuito cerrado de TV analógicos usando DVR de red

Este sistema es muy parecido al sistema anteriormente descrito, solo que el DVR tiene una funcionalidad adicional, transmite el video digitalizado por IP. Esta funcionalidad permite desprendernos del monitor analógico y distribuir el video digitalizado a través de una red IP, lo que facilita la visualización y grabación de video de forma remota.

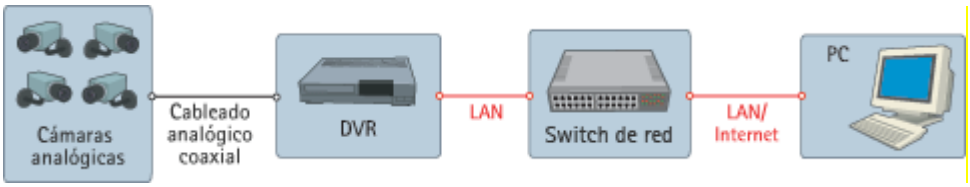


Ilustración 3: Esquema sistema analógico usando DVR en red

Dependiendo del sistema, esta salida de video IP puede ser visualizada a través de navegador o con necesidad de algún tipo de cliente capaz de interpretar la señal.

El sistema DVR IP añade las siguientes ventajas:

- Monitorización remota de vídeo a través de un PC
- Funcionamiento remoto del sistema

IV. Sistemas de vídeo IP que utilizan servidores de vídeo

Este sistema es el primero que empieza a tener ya una cantidad de componentes IP, podríamos decir que es casi un sistema de vigilancia IP con compatibilidad de cámaras analógicas.

En este caso, eliminamos el DVR de la ecuación pasando su función a un software que puede estar ubicado en un servidor remoto. La labor de la digitalización se la dejamos a unos sistemas dedicados específicamente a ello, llamados servidores de video o digitalizadores. Estos digitalizadores se encargan únicamente de la primera etapa del DVR, la digitalización, y transforman la imagen a protocolos estándar de imagen digital (JPEG, H264…) enviando el flujo de video por IP.

Un sistema de vídeo IP que utiliza digitalizadores añade las ventajas siguientes:

- Utilización de red estándar y hardware de servidor de PC para la grabación y gestión de vídeo
- El sistema es escalable en ampliaciones de una cámara cada vez
- Es posible la grabación fuera de las instalaciones
- Preparado para el futuro, ya que este sistema puede ampliarse fácilmente incorporando cámaras IP

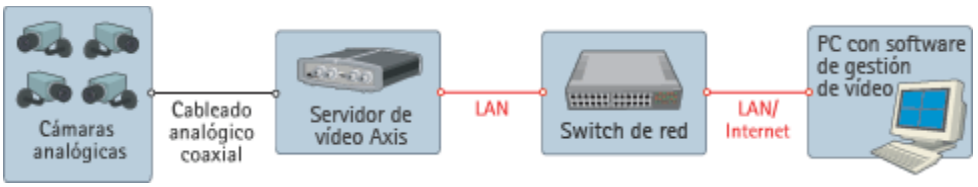


Ilustración 4: Esquema sistema analógico con servidor de video IP

Este diagrama muestra un verdadero sistema de vídeo IP, donde la información del vídeo se transmite de forma continua a través de una red IP. Utiliza un servidor de vídeo como elemento clave para migrar el sistema analógico de seguridad a una solución de vídeo IP.

V. *Sistemas de vídeo IP que utilizan cámaras IP*

Por último, ya encontramos un sistema 100% digital. En comparación con el sistema anteriormente descrito, este sistema utiliza cámaras IP.

Dentro de las cámaras denominadas IP existen dos tipos: cámaras 100% IP y cámaras con sensores analógicos y servidores de video integrados. Existen muchos fabricantes que aún utilizan el segundo tipo, sobre todo los fabricantes que llevan mucho tiempo en el mercado, que tienen sensores de alta calidad y que otorgan calidades muy símiles a los sensores 100% IP, invirtiendo en el servidor de video para ofrecer imágenes de alta resolución.

El procedimiento en este sistema es el siguiente: el vídeo se transmite a través de una red IP, mediante los conmutadores de red y se graba en un PC estándar con software de gestión de vídeo.

Un sistema de vídeo IP que utiliza cámaras IP añade las ventajas siguientes:

- Cámaras de alta resolución (megapíxel)
- Calidad de imagen constante
- Alimentación eléctrica a través de Ethernet y funcionalidad inalámbrica
- Funciones de Pan/tilt/zoom, audio, entradas y salidas digitales a través de IP, junto con el vídeo
- Flexibilidad y escalabilidad completas

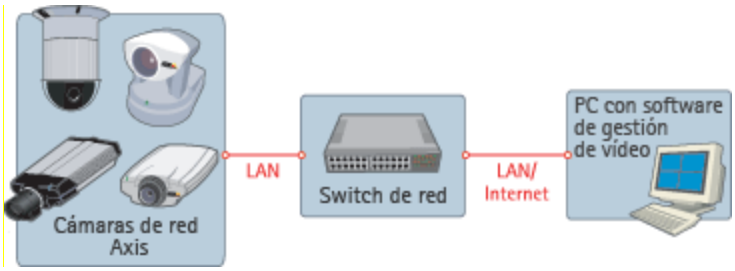


Ilustración 5: Esquema sistema IP

2. Video Analógico vs. Video Digital

I. ¿Qué sistema es más rentable?

Hoy en día, la instalación de cámaras analógicas, junto con los DVR sigue siendo el método más rentable para la mayoría de aplicaciones de seguridad. Sin embargo, el coste de las cámaras y componentes IP se está abaratando rápidamente, haciendo que los sistemas IP vayan siendo más asequibles.

Una cámara domo analógicas típica de calidad media se vende al por menor por un precio entre 50€ y 150€. Una cámara IP similar calidad se vende por lo menos el doble de ese precio. Las cámaras analógicas están disponibles con muchas características diferentes: las lentes de distancia focal variable, PTZ, con infrarrojos de largo alcance… Encontrar la combinación adecuada de características en una cámara de red para su aplicación podría ser más difícil y costoso. En ocasiones, puede ser necesario comprar una cámara analógica y añadir un servidor de vídeo independiente para hacer el trabajo. Los servidores de vídeo en red de un solo canal empiezan actualmente a cerca de 300€ por menor.

Los defensores del video IP pueden señalar que las empresas tienen ya inversión hecha en redes IP, por tanto, no se necesita cableado o hardware adicional. Sin embargo, cada cámara necesita un puerto para conectar al switch, lo que normalmente necesita una inversión en switches para poder aprovechar la misma red. Si queremos montar una red PoE (Power over Ethernet) tendremos también que invertir en este tipo de dispositivos adicionales, con lo que la instalación nos puede salir más costosa que el cableado coaxial.

Siguiendo en cuanto a la red, se ha de considerar el ancho de banda en la red de área local (LAN). El video utiliza una gran cantidad de ancho de banda. El ancho de banda utilizado por cada cámara varía según muchos factores, incluyendo la resolución, el método de compresión, e incluso la cantidad de movimiento en el campo de visión.

Si planteamos un sistema de cámaras analógicas transmiten sobre cable coaxial, no la LAN, por lo que su ancho de banda no es un gran problema. El único uso de la LAN en sistemas analógicos es que el DVR emite los datos de vídeo en la red para los usuarios de

escritorio locales o a Internet. Los DVRs tienden a transmitir video de manera muy eficiente y sólo van a utilizar el ancho de banda si la gente está viendo las cámaras.

En los sistemas de cámaras IP, por otro lado, cada cámara IP utiliza la LAN para transmitir su señal a la NVR así el ancho de banda puede ser un gran problema. Como regla general, una cámara IP con total CIF (352 x 288) resolución, 30 fotogramas por segundo (30 fps), y la compresión MPEG4 requerirá cerca de 720K bits por segundo (720kbps). Por lo tanto, si ponemos 100 cámaras IP que se ejecutan CIF en una red, usaríamos sobre 72Mbps de ancho de banda. Este número se duplicará si no se transmite también de audio. Sin embargo, para colmo de ancho de banda para IP - la mayoría de las cámaras IP más nuevas están saliendo con la resolución 'megapíxeles'. Esto es maravilloso, desde el punto de vista de lo mucho que se puede capturar la claridad y el campo de visión, pero tiene un precio muy alto para el ancho de banda. Una sola cámara IP de 2 megapíxeles, corriendo 30 fps con compresión MPEG4 usará un 6.5Mbps friolera de ancho de banda. No debería ser ninguna sorpresa que algunas empresas han ido tan lejos como para crear una red IP completamente separada sólo para ejecutar su sistema de cámaras.

Las cámaras IP de alta resolución también requieren una gran cantidad de espacio en disco duro para guardar el video. La cámara de 2 megapíxeles único descrito anteriormente requeriría aproximadamente 67 gigas de espacio en disco duro para grabar valor de un día de video.

II. ¿Qué sistema tiene mejor calidad?

Hay componentes de mala calidad y componentes de buena calidad, no importa lo que se utiliza el tipo de sistema. Dicho esto, las cámaras de red ofrecen algunos avances tecnológicos en las áreas de instalaciones de calidad de vídeo e inalámbricas. Las cámaras analógicas no pueden proporcionar una resolución por encima de los estándares de televisión, siendo el máximo de alrededor de 0,4 megapíxeles. La resolución de las cámaras IP puede ser muchas veces mayor (actualmente hasta 5 megapíxeles) y se puede capturar una imagen más clara cuando los objetos se mueven. Esto podría hacer la diferencia en aplicaciones de alto riesgo como para los casinos y las fuerzas del orden. La comunicación inalámbrica sobre redes IP tiene menos problemas de interferencia, y la seguridad de cifrado está incorporada en la tecnología.

III. ¿Cuál sistema es más fácil de configurar e instalar?

Si una red IP ya está en marcha en el lugar de la instalación, y se puede manejar la carga adicional de las nuevas cámaras, a continuación, las cámaras IP serán más fáciles de instalar. Si se necesitan conectores RJ-45 adicionales para conectar las cámaras de red, a continuación, el instalador sólo tiene que correr un cable CAT- 5 desde la cámara al conmutador más cercano. Un interruptor de bajo costo puede ser instalado a la derecha en la toma de pared más cercana. Por el contrario, cada cable para las cámaras analógicas se debe ejecutar todo el camino de vuelta a la DVR. Si las actualizaciones deben hacerse a una red IP existente para manejar la carga adicional, obviamente, la instalación sería más difícil.

El poder de las cámaras se puede manejar con bastante facilidad, ya sea con la tecnología. En las redes IP, construido en adaptadores PoE hacer el envío de la energía a través del cable Ethernet existente fácil. Para sistemas analógicos, utilice cable RG59 combinar los cables de vídeo y alimentación en una chaqueta. De cualquier manera, no hay necesidad de cableado adicional para el poder. POE puede correr 328 metros sin repetidor. RG59 puede ejecutar 1.000 metros sin repetidor.

IV. ¿Cuál es mejor en infraestructuras inalámbricas?

Sistemas inalámbricos analógicos no funcionan bien. Esto se debe a que el gobierno regula en qué frecuencias de los dispositivos inalámbricos analógicos pueden funcionar y qué tan fuerte la señal puede ser. La interferencia de otros dispositivos inalámbricos, como los teléfonos celulares puede causar que el vídeo de la cámara a distorsionarse. La interferencia es especialmente problemática en los edificios con iluminación fluorescente.

IP inalámbrica digital es mucho mejor. La transmisión digital no recibir interferencias de otros dispositivos inalámbricos analógicos, y el estándar 802.11x comunicación utilizado tiene cifrado construido adentro En consecuencia, no hay problema con el acceso no autorizado al video.

V. ¿Para qué aplicaciones debería considerar el vídeo IP?

Un sistema de vídeo IP debe ser considerado para instalaciones de gran tamaño que ya tienen una red de banda ancha de alta instalado - especialmente si las cámaras se extienden sobre un área amplia, o si se utilizarán cámaras inalámbricas.

Para grandes instalaciones con muchas cámaras, algunos instaladores prefieren una solución DVR múltiple a una solución IP. El software se incluye con los DVR de alta gama que le permite ver y grabar las cámaras de múltiples DVRs. La solución DVR múltiple también proporciona una mejor protección de conmutación por error. Si la red se cae en un sistema basado en IP, de vídeo se pierde de todas las cámaras. Si la red se cae en un sistema analógico, los DVRs siguen grabando las cámaras.

Una mejor solución puede ser utilizar una solución híbrida que combina el uso de la tecnología analógica e IP. Consulte los DVRs híbridos y cámaras de seguridad híbrido Sistemas FAQ para más información.

Por otro lado, tanto en instalaciones grandes como pequeñas, se están teniendo necesidades de análisis de video, ya sea por temas de seguridad (detección de intrusiones, controles de presencia, controles de acceso…) como con fines meramente estadísticos (controles de aforo, contadores de personas, estadísticas de flujo de personas por zonas determinadas…).

Para el uso de este tipo de aplicaciones de analítica es necesario tener video digital, ya sea de alta calidad digitalizado o IP directamente, pero con estabilidad en la imagen y, según la analítica, con una serie de características en el entorno y configuración de la cámara.

3. Escenarios

VI. Estadísticas

a. Contadores de personas

Para poder medir la eficacia de una campaña publicitaria, no es únicamente indicativa la cantidad de ventas que se hayan realizado en un periodo, ya que es probable que parte de esas personas a las que ha llegado la campaña, no consuman directamente. Para ello, se utilizan analíticas de cómo puede ser el conteo de personas, que nos permite de forma pasiva tener información fidedigna e instantánea sobre el aforo y los puntos en los que hemos tenido mayor afluencia de personas.

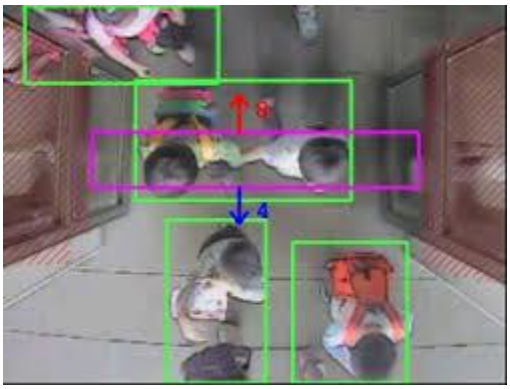


Ilustración 6: Conteo de personas

Para este tipo de analíticas es necesario utilizar una cámara en posición cenital sobre la zona que deseamos realizar el conteo, ya que es la única manera para poder segmentar y discernir las personas sin que una oculte a otra. En muchas ocasiones y con el fin de tener mayor precisión, se utilizan cámaras térmicas, de esta manera quedan directamente excluidos todos los objetos que no produzcan calor, como carros o maletas. Otra forma de optimización es a través de modelos o clasificadores.

b. Mapas de calor

En los centros comerciales, con carácter periódico, se realizan muestreos sobre los recorridos que hacen los clientes, zonas de mayor afluencia y picos de acceso a los mismos. Para ello, hasta ahora, se contrataban a empresas especializadas en ello que ponían a personas en diferentes puntos de los centros con un contador manual o dispositivos físicos que ayudan a estimar este fin. Debido a esta necesidad se han desarrollado diferentes técnicas de captación de esta información aprovechando la infraestructura existente, la red de cámaras de seguridad.

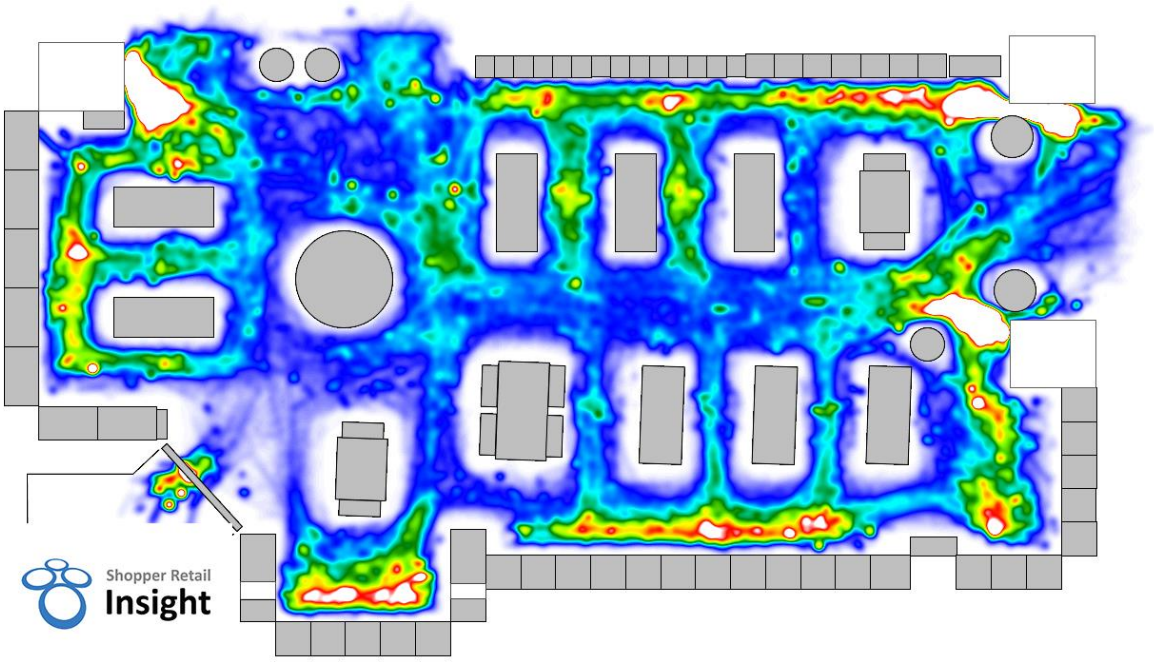


Ilustración 7: Mapa de calor - Centro comercial

El mundo del deporte, y en especial el fútbol, mueve mucho dinero y gracias a la tecnología va evolucionando a marchas forzadas. Esta analítica se lleva utilizando desde hace ya mucho tiempo, siendo inicialmente un sistema supervisado por una operador, que iba marcando las zonas con acciones importantes (http://www.nacsport.com/es/).

c. *Estimación de edad, género y expresiones*

Este tipo de analítica tiene una funcionalidad muy parecida a los contadores de personas, e incluso son complementarias. Gracias a este tipo de analíticas podemos estimar características físicas como el género, la edad, las expresiones, y dependiendo de la implementación, si es un cliente que ha estado anteriormente (usando reconocimiento facial) o cuánto tiempo ha estado mirando a una zona u objeto determinado.

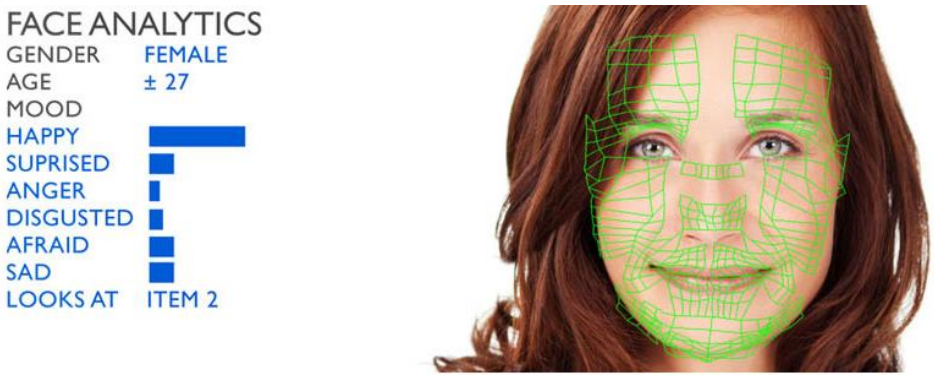


Ilustración 8: Edad, género y expresiones

La aplicación más frecuente en este tipo de analítica es mostrar información relevante para esa persona. Pongamos como ejemplo una tienda de ropa que tiene artículos tanto para hombres como para mujeres y muestra los artículos de la temporada en una pantalla dinámico. Usando este tipo de analítica, se podría mostrar parte de la colección dependiendo de la persona o grupo de personas que se encuentran observando dicha pantalla, haciendo la publicidad más personalizada.

VII. Seguridad

a. *Control de acceso*

La analítica es la más explotada y experimentada del mercado en este ámbito es el reconocimiento o lectura de matrículas. A día de hoy está presente en muchos ámbitos, como puede ser en los accesos a autopistas de peajes, aparcamientos públicos y en sistemas de control de tráfico, como los cinemómetros o radares de control de velocidad. Debido a su reducción de coste y a la nueva necesidad de seguridad que empieza a existir, se empieza a aplicar en otros escenarios como en seguridad ciudadana (búsqueda de

coches robados o investigación de incidentes) e incluso en accesos a edificios privados y centros de estudio.

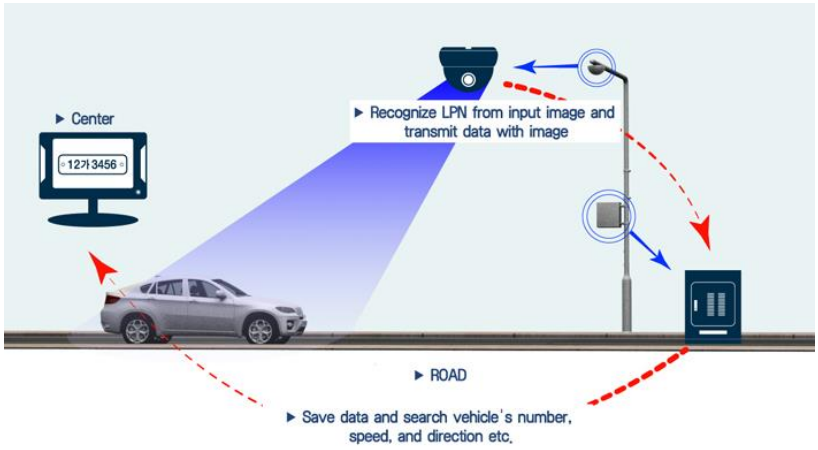


Ilustración 9: Reconocimiento de matrículas

Por otro lado, para el control de acceso, aunque la tecnología más utilizada es la lectura de tarjetas, se está empleando cada día más y más el control de acceso biométrico. Debido a su precio, el más extendido es el reconocimiento de huella dactilar, ya que ofrece una tasa de falso positivo realmente baja a un coste bastante pequeño.

El problema de esta biometría es que requiere contacto con el dispositivo, lo que requiere un mantenimiento continuo para que se consiga mantener las tasas de reconocimiento aceptables y provoca un deterioro más rápido que otro tipo de biometrías que no requieren contacto. Entre las biometrías que no requieren contacto nos encontramos el reconocimiento de iris y reconocimiento facial.

El reconocimiento de iris, antiguamente, requería un uso muy parecido al del reconocimiento por huella dactilar. En los últimos años se ha avanzado en este tipo de tecnología, permitiendo realizar dicho reconocimiento en movimiento a una distancia aproximada de 5 metros de distancia. Estos avances nos hacen pensar en el uso del iris frente a la huella dactilar. La principal razón para dar el paso es la escasa tasa de falsos positivos. Con las huellas dactilares, tenemos una probabilidad de 1/100.000, mientras que con el iris la probabilidad desciende a 1/1.200.000.

En cuanto al reconocimiento facial, no tiene ni por asomo la misma tasa de falso positivo que el reconocimiento de iris, pero su precisión sigue siendo aceptable para la mayoría de los entornos. La principal ventaja de esta biometría es la distancia y la poca intrusividad que tiene. Dependiendo del algoritmo y dispositivo de captura de imágenes que utilicemos, podemos realizar el reconocimiento facial a poca distancia con una cámara de muy bajo coste o a distancias de más de cien metros, dando mucha flexibilidad para su uso.

b. Video vigilancia

Tradicionalmente la video vigilancia requería un uso activo de actores como puede ser los vigilantes de seguridad, que en instalaciones de grandes dimensiones tenían que estar pendientes de muchas más cámaras de las que podían estar pendientes, por lo que la detección de cualquier suceso era más bien suerte, si no se utilizaban dispositivos adicionales como sensores de presión o corte laser.

Para solucionar esta situación, en la video vigilancia se utilizan analíticas de vídeo de detección de sucesos, como puede ser analíticas de detección perimetral, en las que se define una zona restringida, y si aparece algún objeto o persona en dicha zona, se le notifica al sistema y por ende al personal de seguridad encargado de monitorizar el sistema.

En este ámbito, también se utilizan otro tipo de analíticas como el reconocimiento facial, teniendo el escenario más claro infraestructuras críticas como aeropuertos o centros penitenciarios.

4. Tarea a solventar

Para la elaboración de este proyecto, tomamos como caso la seguridad en una sucursal bancaria. Con el fin de evitar accesos no deseados y de poder identificar a todas las personas que accedan a la sucursal, necesitamos un sistema que nos notifique cuando una persona no tiene visible el rostro.

En muchos casos acceden personas a sucursales bancarias con el rostro parcial o totalmente oculto, ya sea por causas no relacionadas, como puede ser un sobrero y gafas de sol en épocas de verano, como bufanda y capucha en invierno. Aun así existen casos en los que sean oclusiones porque no quieren poder ser identificados y estos casos normalmente son los que vamos a intentar evitar, ya que el sistema que se definirá a continuación tiene carácter preventivo.

Estado del arte

1. *Técnicas de detección de movimiento*

Existen diferentes técnicas de detección de movimiento aunque las más usuales son:

- Sustracción del fondo (Background subtraction)
- Imagen diferencia absoluta
- Imagen de diferencias acumuladas
- Ajuste de bloques (block matching)

I. *Detección de cambios en el flujo óptico*

a. *Sustracción de fondo (Background substraction)*

Un algoritmo robusto de sustracción de fondo debe ser capaz de gestionar cambios de iluminación, movimientos repetitivos y cambios de larga duración. Los siguientes análisis utilizan la función *V*(*x*,*y*,*t*) como secuencia de video donde *t* es el tiempo, x e y son la posición de los píxeles, por ejemplo *V(1,2,3)* es la intensidad del pixel en la posición (1,2) en el instante *t = 3* de la secuencia de video.

a. Imagen diferencia absoluta

La imagen diferencia (absoluta) en el instante *t* + 1 es

$$D(t+1) = |V(x,y,t+1) - V(x,y,t)|$$

La imagen en el instante *t* es establecida como fondo. La diferencia en la imagen mostrará únicamente la intensidad en los pixeles que han cambiado entre las dos imágenes. Debido a que hemos eliminado el fondo, esta aproximación solo funcionara

cuando los pixeles de la imagen a comparar estén en movimiento mientras que los del fondo sean estáticos.

El umbral "Th" es utilizado en esta diferencia de imágenes para mejorar la sustracción.

$$|V(x,y,t) - V(x,y,t+1)| > \mathrm{Th}$$

Ilustración 10: Imagen previa aplicación de umbral

Ilustración 11: Imagen con umbral aplicado

(esto significa que la diferencia entre las intensidades de los píxeles de las imágenes están "umbralizadas" o filtradas en base al valor del umbral)

La precisión de esta técnica depende en la velocidad del movimiento en la escena. Cuanto más rápido sea el movimiento, necesitará umbrales mayores.

b. Filtro de media

Con el fin de calcula la imagen contenida solo en el fondo, se hace la media entre un número determinado de imagines que la preceden. Para calcular la imagen que estableceremos cono fondo en el instante t utilizamos la siguiente fórmula,

$$B(x,y) = \frac{1}{N}\sum_{i=1}^{N} V(x,y,t-i)$$

donde *N* es el número de imágenes escogidas para realizar la media de cada pixel. *N* dependerá de la velocidad del vídeo (imágenes por segundo) y la cantidad de movimiento en el vídeo. Tras calcular el fondo *B(x,y)* podemos extraerlo de la imagen *V(x,y)* en el instante *t* = t y aplicarle el umbral. Por lo tanto, la imagen resultante será

$$|V(x,y,t) - B(x,y)| > \mathrm{Th}$$

donde Th es el umbral. En ocasiones, se puede utilizar la mediana en vez de la media para el cálculo de *B(x,y)*.

El uso global y de umbrales dependientes del tiempo (mismo TH para todos los pixeles de la imagen) puede limitar la precisión de las dos técnicas anteriores.

c. Media Gaussiana

Para el uso de esta técnica, Wren y otros, propone usar la función de densidad probabilística Gaussiana (pdf) en las n imágenes más recientes, esto ayuda a evitar el cálculo del pdf desde cero en cada imagen recibida en el instante t, sino la media acumulada.

El pdf de cada pixel se caracteriza por la media y la varianza. Lo dispuesto a continuación es una posible condición inicial (asumiendo que inicialmente cada pixel es parte del fondo)

$$\mu_0 = I_0$$

$$\sigma_0^2 = <\text{valor por defecto}>$$

donde I_t es el valor del pixel en el instante *t*. Con el fin de inicializar la varianza, podemos usar la varianza en x e y como una pequeña ventana entorno a cada pixel.

Hemos de tener en cuenta que el fondo puede cambiar a lo largo del tiempo debido a cambios de iluminación, objetos no estáticos, etc. Para asimilar estos cambios,

en cada imagen en el instante *t*, la media y varianza de cada pixel ha de ser actualizada como se indica a continuación:

$$\mu_t = \rho I_t + (1-\rho)\mu_{t-1}$$

$$\sigma_t^2 = d^2\rho + (1-\rho)\sigma_{t-1}^2$$

$$d = |(I_t - \mu_t)|$$

donde ρ determina el tamaño temporal de la ventana que se usa para cuadrar el pdf (normalmente $\rho = 0.01$) y d es la distancia euclídea entre la media y el valor del pixel.

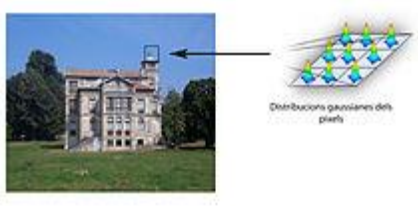


Ilustración 12: Distribuciones Gaussianas

De esta manera podemos clasificar cada pixel como parte del fondo, si su intensidad actual se encuentra dentro de un intervalo determinado de su distribución de la media:

$$\frac{|(I_t - \mu_t)|}{\sigma_t} > k \longrightarrow Foreground$$

$$\frac{|(I_t - \mu_t)|}{\sigma_t} \leq k \longrightarrow Background$$

donde el parámetro k es un umbral libre (normalmente $k = 2.5$). Un valor mayor en k permite un tener un fondo más dinámico, mientras que un valor menor de k incrementa la probabilidad de una transición de pixeles fuera del fondo debido a cambios menos significativos.

En una variante de la técnica, la distribución de un pixel solo es actualizada si entra a formar parte del fondo. Esto evita que los objetos que aparezcan en las nuevas

imágenes entren a formar parte del fondo directamente. La formula resultado actualizada sería la siguiente:

$$\mu_t = M\mu_{t-1} + (1 - M)(I_t\rho + (1 - \rho)\mu_{t-1})$$

donde $M = 1$ cuando I_t es considerado fuera del fondo, en caso contrario $M = 0$. Por lo tanto cuando $M = 1$, el pixel no se considera parte del fondo y la media permanece intacta. Como resultado, una vez establecido el pixel dentro del primer plano, puede únicamente volver a pertenecer al fondo cuando la intensidad del valor se acerque al mismo que era cuando estaba convirtiéndose en parte del primer plano. En cualquier caso, esta técnica tiene una serie de problemáticas: solo funciona si todos los pixeles pertenecen inicialmente al fondo o al primer plano se establecen como fondo. Por otro lado, no puede hacer frente a cambios de fondo graduales: Si un pixel se establece como parte del primer plano durante un periodo prolongado de tiempo, la intensidad del fondo en esa posición puede haber cambiado (por cambio en la iluminación, etc.). Como resultado, una vez que el objeto del primer plano se ha ido, la nueva intensidad de fondo puede no ser reconocido como tal nunca más.

d. Modelos de mezcla de fondo

En el uso de esta técnica, se supone que los valores de intensidad de cada píxel en el video pueden ser asignados usando un modelo de mezcla gaussiana. Una simple heurística determina cuales intensidades son tienen mayor probabilidad de pertenecer al fondo y cuales, por descarte, al primer plano. Los píxeles de primer plano se agrupan utilizando el análisis de componentes conectadas 2D.

En cualquier instante t, un pixel (x_0, y_0) es

$$X_1, \ldots, X_t = \{V(x_0, y_0, i) : 1 \leqslant i \leqslant t\}$$

Este valor esta adquirido a través de una mezcla de K distribuciones Gaussianas:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} N\left(X_t \mid \mu_{i,t}, \Sigma_{i,t}\right)$$

donde

$$N\left(X_t \mid \mu_{it}, \Sigma_{i,t}\right) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_{i,t}|^{1/2}} \exp\left(-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1}\left(X_t - \mu_{i,t}\right)\right)$$

La aproximación K-media se utiliza para actualizar las gaussianas.

b. Imagen Diferencia

Resulta obvia, como método de detección de cambios, la simple comparación entre 2 imágenes, implementada como una resta en valor absoluto. Podemos umbralizar esa resta para obtener así una imagen diferencia binaria:

$$DP_{jk}(x,y)=1 \text{ si } |F(x,y,j)-F(x,y,k)|>\tau$$

$$DP_{jk}(x,y)=0 \text{ en otro caso}$$

Ilustración 13: Imágenes en t y t+1 para realizar Imagen Diferencia

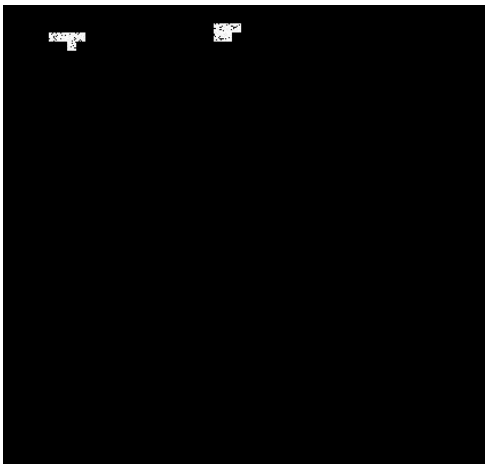


Ilustración 14: Resultado Imagen Diferencia

Problema: movimientos lentos pueden perderse o ruido acentuarse, dependencia del umbral.

Para evitar el ruido en la detección podemos restringir el área de la región detectada imponiendo un umbral de tamaño.

Sin embargo esta técnica puede conllevar pérdidas de información útil de movimientos lentos o de pequeños objetos. Más robusto si comparamos regiones de forma estadística, mediante convoluciones usando máscaras locales y comparar distribuciones de píxeles en diferentes regiones: medidas de similitud.

$$\lambda = \frac{\left[\frac{\sigma_1 + \sigma_2}{2} + \left(\frac{\mu_1 - \mu_2}{2}\right)^2\right]^2}{\sigma_1 \cdot \sigma_2}$$

c. Imagen de diferencias acumuladas

Pero con esta técnica de medidas de similitud podemos seguir teniendo problemas con los objetos pequeños o movimientos lentos por lo que buscamos otro método de detección: Imagen de diferencias acumuladas (ADP).

Si en lugar de analizar 2 imágenes de la secuencia lo hacemos con un mínimo de 3 podremos conseguir información más fiable de la misma, evitando así ruido casual y detectando movimientos más lentos.

La ADP se forma comparando todas las imágenes de la secuencia con el de referencia, incrementando el valor en la ADP si se sobrepasa un umbral para la diferencia de intensidad de los píxeles o alguna medida para el superpíxel.

La ADPk es calculada sobre k imágenes mediante:

$$ADP_0(x,y)=0$$

$$ADP_k(x,y)=ADP_{k-1}(x,y)+DP_{1k}(x,y)$$

Como vemos hace uso de la imagen diferencia y para la imagen 0, inicializamos a 0.

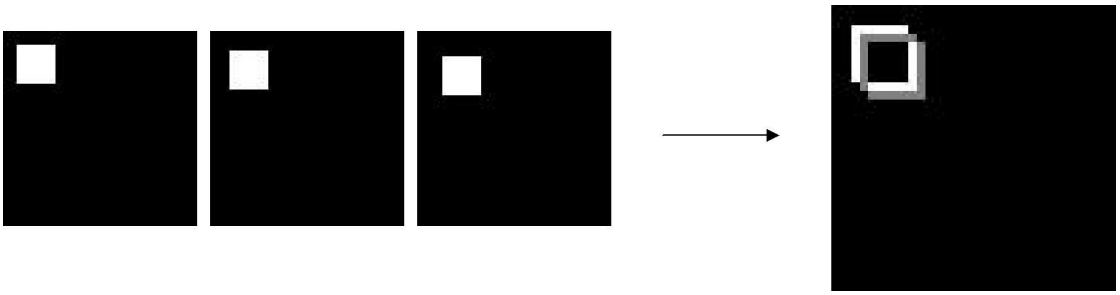


Ilustración 15: Imagen de diferencias acumuladas

Se divide la imagen en bloques y se estima el desplazamiento bajo una medida de error. El desplazamiento que minimice el error se considera el movimiento de la imagen. Las técnicas más usuales de búsqueda inteligente son:

- Método Three-Step
- Método conjugado modificado

a. Método Three-Step

Típicamente en 3 iteraciones se estima un desplazamiento dentro de una distancia máxima (dm) en las 2 direcciones. Se evalúan los puntos (0,0), (0,m), (m,0), (m,m), (-m,0), (0,-m), (-m,-m), (m,-m) y (-m,m) en cada iteración.

Por ejemplo, para una distancia máxima de 6 px, m toma los valores 3, 2 y 1, en cada iteración.

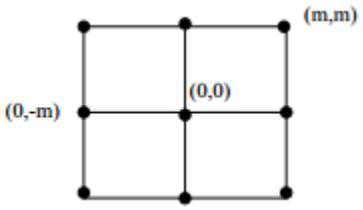


Ilustración 16: Método Three-Step (i)

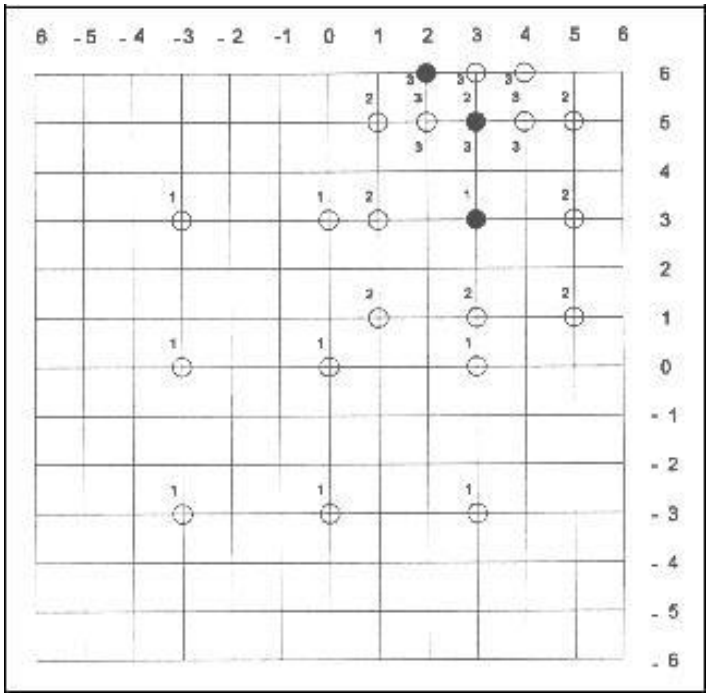


Ilustración 17: Método Three-Step (ii)

b. Método Conjugado Modificado

Se estima un desplazamiento dentro de una distancia máxima (dm) escogiendo una separación (s) entre píxeles a evaluar. Se evalúan los puntos (0,0), (s,0), (-s,0) bajo una medida de error pudiendo suceder 3 casos.

- Mínimo en (0,0): Reducimos s a la mitad y buscamos el mínimo. Logrado, empezamos con igual criterio que con el eje x pero en el eje y ((0,s) y (0,-s))

- Mínimo en (s,0): Continuamos añadiendo puntos a la derecha hasta que el central nos dé mínimo o hasta llegar al máximo desplazamiento, que entonces evaluaremos el punto s/2 a su izquierda, escogiendo el de menor error y conmutando a la dirección y a partir de ese.

- Mínimo en (-s,0): Procedemos de manera simétrica respecto al caso anterior.

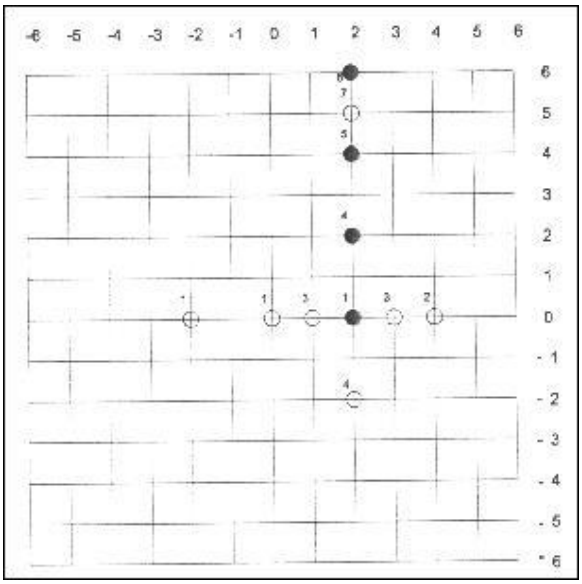


Ilustración 18: Método Conjugado unificado

d. Segmentación por movimiento

La segmentación en general tratará de separar las componentes dinámicas de las estáticas.

Puede ser difícil en casos donde la cámara esté en movimiento puesto que extraer componentes estáticas dependerá del conocimiento del movimiento del sistema de referencia.

Las técnicas más usuales de segmentación en secuencias de imágenes son:

- Detección de bordes
- Detección de regiones

a. Detección de bordes cambiantes en el tiempo

La detección de bordes en escenas estáticas juega un papel muy importante. En escenas dinámicas no es para menos.

La idea es la de combinar gradientes espaciales y temporales usando un operador lógico AND.

$$E_{,}(x,y,t) = \frac{dF(x,y,t)}{dS} \cdot \frac{dF(x,y,t)}{dt} = E(x,y,t) \cdot D(x,y)$$

Aplicando un umbral al producto en lugar de cada uno de los factores detectaremos bordes en movimiento de manera más robusta.

Responderá bien en caso de bordes de poco contraste pero de movimiento rápido y viceversa.

b. Uso de imágenes diferencia en segmentación

Las imágenes diferencia y diferencia acumuladas encuentran áreas en la escena las cuales han cambiado, buenas áreas para segmentar.

Es posible segmentar una escena con poca computación usando las imágenes de diferencias acumuladas.

Puede convenir definir otros tipos de imágenes de diferencias: Imagen diferencia Absoluta:

$$DP_{12}(x,y) = 1 \text{ si } | F(x,y,1) - F(x,y,2)| > \tau$$
$$DP_{12}(x,y) = 0 \text{ en otro caso}$$

- Diferencia Positiva

$$PDP_{12}(x,y) = 1 \text{ si } |F(x,y,1) - F(x,y,2)| > \tau$$
$$PDP_{12}(x,y) = 0 \text{ en otro caso}$$

- Diferencia Negativa

$$NDP_{12}(x,y) = 1 \text{ si } |F(x,y,1) - F(x,y,2)| < \tau$$
$$NDP_{12}(x,y) = 0 \text{ en otro caso}$$

- Imagen diferencias Absoluta Acumuladas

$$\text{AADP}_n(x,y) = AADP_{n-1}(x,y) + DP_{1n}(x,y)$$

- Diferencia Positiva Acumuladas

$$\text{PADP}_n(x,y) = \text{P}ADP_{n-1}(x,y) + \text{P}DP_{1n}(x,y)$$

- Diferencia Negativa Acumuladas

$$\text{NADP}_n(x,y) = \text{N}ADP_{n-1}(x,y) + \text{N}DP_{1n}(x,y)$$

e. Mean Shift

Mean Shift es un procedimiento para la localización de los máximos de una función de densidad dada basada en una muestra de datos discretos. Se utiliza para detectar los modos con esa densidad. Es un método iterativo, en el que comenzamos con un estimado inicial x. Tomamos una función Kernel $K\ (x_i - x)$, con ella determinamos el peso de los puntos vecinos para reestimar la media. Normalmente se utiliza un Kernel Gaussiano para estimar la distancia al estimado actual $K\ (x_i - x) = e^{-c\,||x_i - x||^2}$. La media calculada *K* en la ventana es

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

Donde $N(x)$ es el vecino de x para un conjunto de puntos que cumplen la condición $K(x) \neq 0$. El algoritmo Mean Shift establece $x \leftarrow m(x)$, repitiendo estos pasos hasta que $m(x)$ converja.

Normalmente, este algoritmo se utiliza para seguimiento de objeto. La manera más sencilla de usar este algoritmo es creando un mapa de confianza en la imagen nueva basada en el histograma de color del objeto de la imagen anterior, y usar el Mean Shift para encontrar la confianza máxima del mapa más cercana a la posición anterior. El mapa de confianza es una función de densidad de probabilidad de la imagen nueva, asignando cada pixel de la imagen un valor de probabilidad. Esta probabilidad es la del color del pixel que se repita respecto a la imagen anterior. Existen pocos algoritmos que mantengan utilicen este principio, como puede ser el Cam Shift.

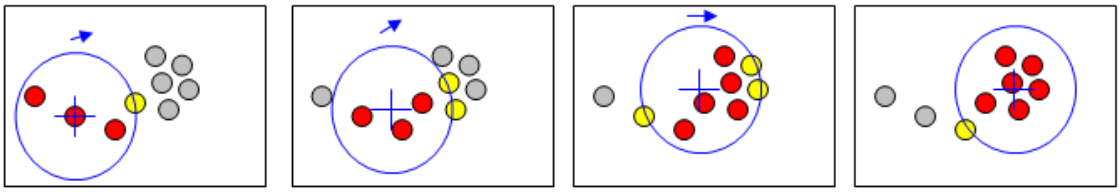


Ilustración 19: Ejemplo de funcionamiento del algoritmo Mean Shift

f. Cam Shift

El algoritmo Cam Shift se puede resumir en los siguientes pasos:

1. Establecer la región de interés (ROI) donde se busca la distribución de probabilidad dentro de la imagen.

2. Seleccionar la ubicación inicial del Mean Shift para cada ventana. La ubicación seleccionada es la distribución objetivo a seguir.

3. Calcular la distribución de probabilidad de color en la región centrada en la ventana de búsqueda generada con Mean Shift.

4. Iterar el algoritmo de Mean Shift para encontrar el centro de la imagen de probabilidad, almacenar el primer momento (área de distribución) y el centro de su ubicación.

5. En la siguiente imagen, centramos la ventana de búsqueda en la ubicación media encontrada en el paso 4 y establecemos el tamaño de la ventana en función del primer momento. Finalmente volvemos al paso 3.

La creación del histograma de color corresponde a los pasos del 1 al 3. El primero paso es definir la región de interés (ROI), la cual es el cuadro delimitador correspondiente al objeto detectado que queremos seguir. Tras ello, necesitamos calcular el histograma de color del propio objeto. Utilizamos el espacio de color HSV, y calculamos un histograma unidimensional correspondiente a la primera componente: Hue (matiz). Se define también una máscara para el cálculo del histograma, la imagen fondo, para calcular el histograma solo de la persona descartando el fondo dentro del cuadro delimitador.

Los resultados obtenidos con este método pueden no ser satisfactorios, porque en el caso en que el fondo tiene un color muy aproximado al objeto, no es posible realizar la detección. Por eso es por lo que se utiliza finalmente un histograma en tres dimensiones, usando las tres componentes del HSV. Con este método somos capaces de encontrar la ubicación del objeto en la imagen, incluyendo los casos en los que tiene un color similar al fondo.

En cualquier caso, los valores del histograma están escalados según un rango discreto de pixeles usando una distribución de probabilidad 2D:

$$\left\{ \hat{p}_u = \min\left(\frac{255}{\max\{\hat{q}\}} \hat{q}_u, 255 \right) \right\}_{u=1\cdots m}$$

Con el histograma del objeto en movimiento, es necesario encontrarlo en todas las imágenes (pasos 4 y 5). Con este fin, calculamos la retroproyección del histograma en la siguiente imagen recibida. Para cada pixel de todas las imágenes recibidas, en la retroproyección, añadimos el valor del histograma de cada pixel. En términos estadísticos, el valor de cada pixel resultado es la probabilidad del pixel observado en el objeto en seguimiento, dada su distribución (histograma). Finalmente, usando la ubicación anterior del objeto, se detecta la nueva posición del objeto en movimiento usándola como ventana inicial de búsqueda de la siguiente imagen. El centro de la ventana de búsqueda se calcula con las siguientes fórmulas:

$$M_{00} = \sum_x \sum_y I(x,y)$$

$$M_{10} = \sum_x \sum_y xI(x,y)$$

$$M_{01} = \sum_x \sum_y yI(x,y)$$

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}}$$

El siguiente punto central de la ventana de seguimiento son x_c y y_c. Después de realizar este cálculo, se vuelve al paso 3 para calcular el nuevo histograma del objeto y actualizar el anterior histograma, usando una actualización lenta para mantener la diferencia entra las posiciones de los objetos si se están solapando.

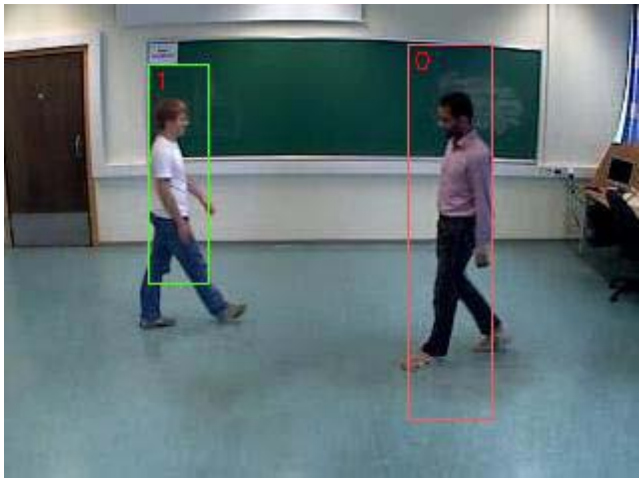


Ilustración 20: Ejemplo de ventana de seguimiento usando Cam Shift

g. Mean Shift vs. Cam Shift

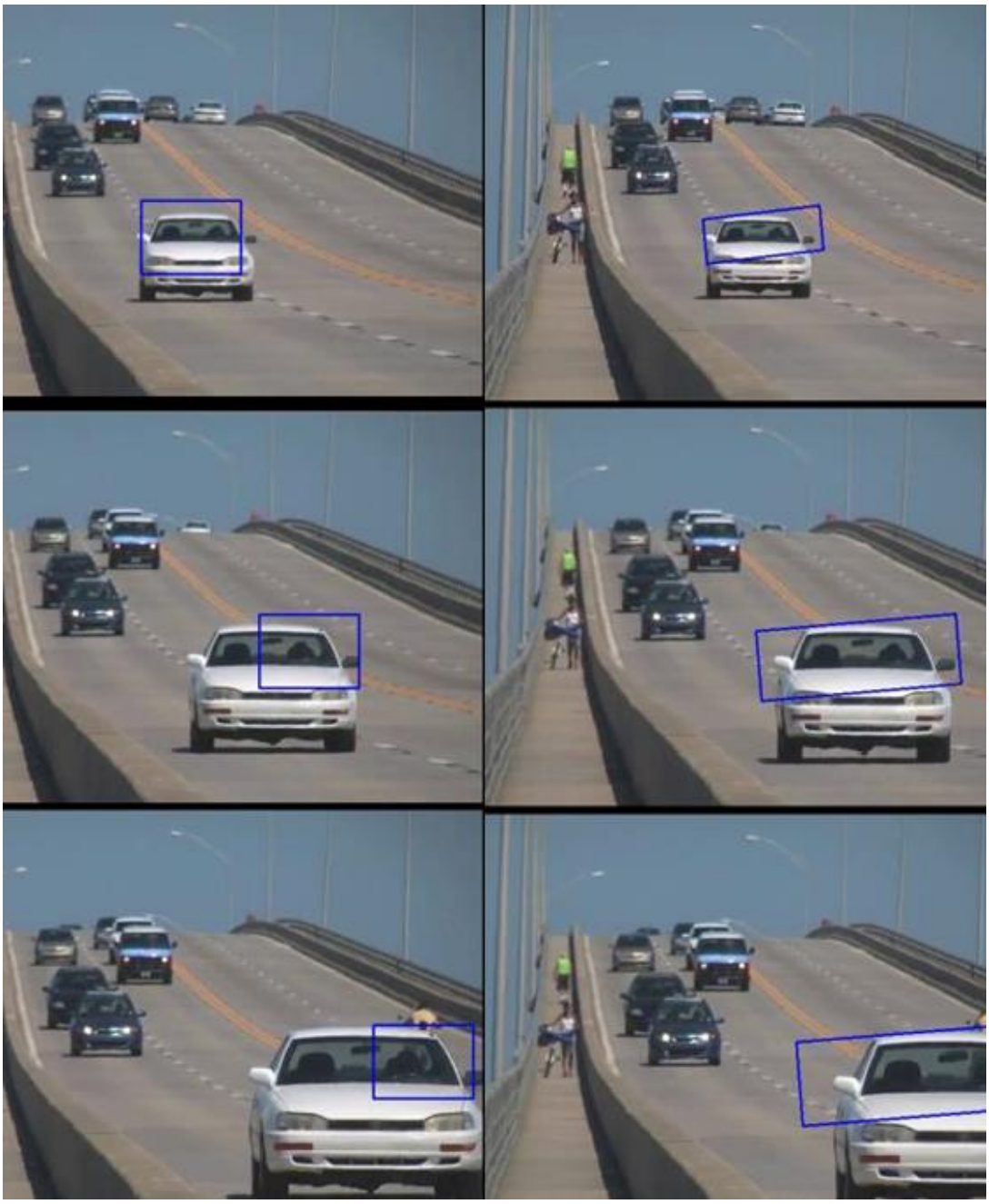


Ilustración 21: Mean Shift vs. Cam Shift

2. Detección de características

I. Detección rápida de objeto usando cascadas optimizadas de características simples.

Este procedimiento de detección de objetos clasifica las imágenes basándose en valores de características simples.

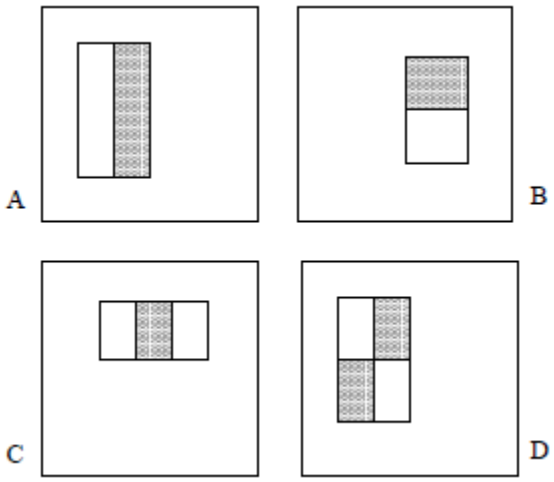


Ilustración 22: Ejemplo de características rectangulares en una ventana de detección predefinida.

La suma de los pixeles que se encuentra dentro de los rectángulos es extraída de la suma de los pixeles en los rectángulos grises como parte interna de la característica definida. Las figuras A y B muestran características basada en dos rectángulos, mientras que la figura C muestra la característica basada en tres rectángulos y la D por cuatro rectángulos.

Existen muchas razones para el uso del análisis por características en vez del análisis de píxeles de forma aislada. La razón más común es que a través de las características podemos definir un entorno de conocimiento personalizado que es más difícil de construir a partir de un numero finito de aprendidos. Otra importante razón para

su uso es que los sistemas basados en características son perceptiblemente más rápidos que los sistemas basados en píxeles. Las características simples que se usan son reminiscencias de funciones de base de Haar que han sido utilizados por Papageorgiou y otros

De forma más específica, se utilizan tres tipos de características. El valor de la característica basada en dos rectángulos es la diferencia entre la suma de los pixeles definidos en dos regiones rectangulares. Las regiones que tienen el mismo tamaño y forma y son adyacentes horizontalmente o verticalmente. La característica basada en tres rectángulos se calcula a través de la suma de los dos rectángulos exteriores extraídos de la suma del rectángulo central. Por último, la característica basada en cuatro rectángulos se calcula a través de la diferencia entre los pares de rectángulos que se encuentran diagonalmente entre sí.

Usando una resolución base del detector de 24x24, el conjunto exhaustivo de características rectangulares es bastante grande, mayor a 180.000. Debemos tener en cuenta que a diferencia de la base Haar, el conjunto de características rectangulares está sobredimensionado.

a. Imagen Integral

Las características rectangulares se pueden calcular muy rápidamente utilizando una imagen intermedia a la que llamamos imagen integral. Una imagen integral ubicada en x,y contiene la suma de los pixeles superiores y situados a la izquierda de x,y, con el punto inicial incluido:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

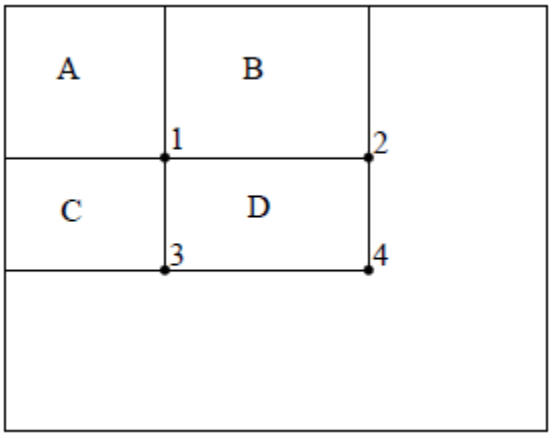


Ilustración 23: La suma de los pixeles en el rango D puede ser computado con un array de 4 referencias.

El valor de la imagen integral en 1 es la suma de los pixeles en el rectángulo A, valor en 2 es A+B, en la posición 3 es A+C, y en la posición 4 es A+B+C+D. La suma en D se puede calcular como 4-1-(2+3).

Siendo ii(x, y) la imagen integral y i(x, y) la imagen original. Usamos el siguiente par de recurrencias:

$$s(x,y) = s(x,y-1) + i(x,y) \qquad (1)$$
$$ii(x,y) = ii(x-1,y) + s(x,y) \qquad (2)$$

(donde s(x, y) es la suma acumulada de las filas, s(x-1) = 0, y ii(-1, y) = 0) la imagen integral se puede calcular en un paso sobre la imagen original.

Usado la imagen integral, se puede calcular cualquier suma rectangular en un array de cuatro referencias. Por lo tanto, la diferencia entre dos sumas rectangulares se puede calcular en ocho referencias. Teniendo en cuenta que las características basadas en dos rectángulos definidas anteriormente requieren sumas de rectángulos adyacentes, se pueden calcular en arrays de seis referencias ocho en el caso de características basadas en tres rectángulos y nueve en el caso de características basadas en cuatro rectángulos.

En resumen, las características rectangulares son algo arcaicas si las comparamos con alternativas como los filtros orientables. Los filtros orientables y similares, son perfectos para análisis detallados de límites, compresión de imagen y análisis de texturas. Por contrario las características rectangulares, aunque son más sensibles a la presencia de bordes, barras y otras estructuras simples, son bastante robustas. Mientras que los filtros orientables solo permiten orientaciones verticales, horizontales y diagonales, el conjunto de características rectangulares permiten una representación más rica de la imagen que aporta un aprendizaje más efectivo. En conjunto con la imagen integral, la eficiencia de las características rectangulares provee una amplia compensación con su limitada flexibilidad.

b. Funciones de clasificación para el aprendizaje

Dado un conjunto de características y un conjunto de entrenamientos de imagines positivas y negativas, cualquier enfoque de aprendizaje automático puede ser utilizado como función de clasificación para el aprendizaje. Usando el ejemplo de AdaBoost, podemos utilizar una variante para seleccionar un pequeño conjunto de características y enseñar al clasificador. En su formato original, el algoritmo de aprendizaje de AdaBoost se utiliza para aumentar rendimiento de la clasificación de un algoritmo de aprendizaje simple. Freund and Schapire pudieron probar que un error en el entrenamiento de una clasificador robusto se aproxima a cero exponencialmente en diferentes ensayos, siendo más importante el número de resultados que posteriormente probaría la generalización del rendimiento. La clave es que la generalización del rendimiento se relaciona con el margen de los ejemplos, y que con AdaBoost logra grandes márgenes de forma rápida.

Recordemos que hay más de 180.000 características basadas en rectángulos asociadas a cada imagen de una sub-ventana, un número mucho mayor que el número de píxeles. Incluso sabiendo que cada función se puede calcular muy eficientemente, la realización del conjunto completo es prohibitivamente pesado. La hipótesis, es que un número muy pequeño de estas características se pueden combinar para formar un clasificador eficaz.

Para poder conseguir este reto, se ha diseñado el algoritmo débil de aprendizaje, que selecciona una característica rectangular de forma más eficaz los ejemplos positivos de los negativos.

Para cada característica, el algoritmo de aprendizaje débil determina el umbral óptimo de la función de clasificación, que será el mínimo número de ejemplos que se desecharan de la clasificación. Un clasificador débil *hj(x)* consiste en una característica *fj*, un umbral *Θj* y una paridad *pj* indicando la dirección de la desigualdad en el signo:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

En la práctica, no hay ninguna característica que pueda realizar la tarea de clasificación con una tasa de error baja. Las características que se seleccionan en las primeras fases de la optimización del proceso han de tener tasas de error de entre un 0.1 y un 0.3. Las características seleccionadas en las últimas rondas, debido a que la tarea se vuelve más complicada, tendrán una tasa de error de entre un 0.4 y un 0.5

Para un caso similar de reconocimiento, Papgeorgiou y otros propusieron un esquema para la selección de características basado en la varianza de las mismas.

Roth y otros propusieron un proceso de selección de características basado en la regla de aprendizaje del perceptrón exponencial Winnow. El proceso de aprendizaje de Winnow converge en una solución donde muchos de los pesos son cero, sin embargo un gran número de características se conservan.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Ilustración 24: Algoritmo de clasificación de AdaBoost

Desafortunadamente, la técnica más directa para mejorar el rendimiento en la detección es añadir características al clasificador, lo que incrementa el tiempo de computación. Para la detección facial, los rectángulos iniciales de características seleccionados por AdaBoost son significativos y fácil de interpretar.

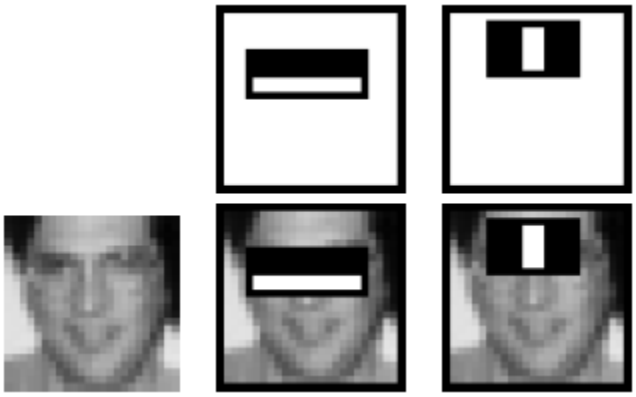


Ilustración 25: Primera y segunda característica seleccionada con AdaBoost

La primera característica mide la diferencia en la intensidad entre la región de los ojos y la zona superior de los pómulos. Esta característica es relativamente grande comparada con la detección de la sub-ventana, y no debe ser sensible a tamaños y localización de la cara. La segunda característica compara las intensidades en la zona de cada ojo con la intensidad del puente de la nariz.

Ambas características se basan en la premisa de que normalmente la zona de los ojos es más oscura que la zona superior de los pómulos

c. Cascada de atención

Esta sección describe un algoritmo para la construcción de una cascada de clasificadores el cual consigue mayor rendimiento durante la detección, al mismo tiempo que reduce radicalmente el cálculo computacional. La idea es que clasificadores optimizados más pequeños, y por tanto más eficientes, se pueden construir para rechazar muchas de las sub-ventanas negativas mientras que se detectan la mayoría de las instancias positivas (ej. El umbral de un clasificador optimizado se puede ajustar para que su tasa de falso negativo sea lo más cercano a cero). Los clasificadores más sencillos se utilizan para rechazar la mayoría de las sub-ventanas antes de usar clasificadores más complejos, que son los encargados de reducir una tasa baja de falsos positivos.

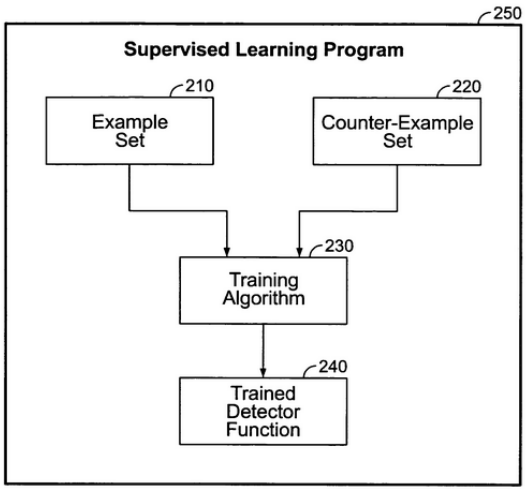


Ilustración 26: Entrenamiento de una cascada de atención

La forma general del proceso de detección es a través de un árbol de decisión alterado, la cual denominamos "cascada". Un resultado positivo del primer clasificador genera la evaluación por un segundo clasificador, el cual también está ajustado para conseguir ratios de detección altos. Un resultado positivo del segundo clasificador genera el uso de un tercer clasificador, y así en adelante. Por el contrario, un resultado negativo en cualquier momento conlleva a rechazar inmediatamente la sub-ventana. Las diferentes etapas del proceso en el cascada se construyen mediante la generación de clasificadores usando AdaBoost y ajustando el umbral para minimizar los falsos negativo. El umbral por defecto de AdaBoost está diseñado para producir una tasa de error baja durante el proceso de entrenamiento. En general, un umbral bajo durante el entrenamiento permite tener tasas altas de detección y por lo tanto tasas altas de falsos positivos.

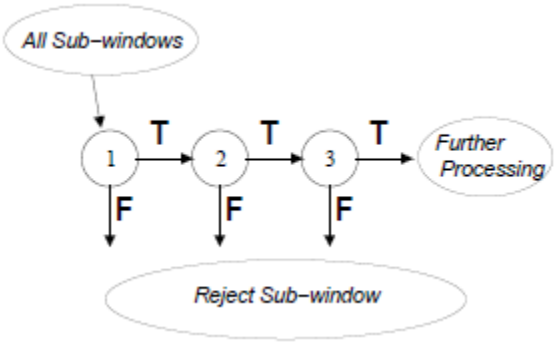


Ilustración 27: Esquema de una cascada de detección

Una serie de clasificadores se aplican a cada sub-ventana. El clasificador inicial se encarga de descartar un gran número de resultados negativos utilizando muy poco procesamiento. Las capas posteriores eliminan aquellos resultados negativos que no han aparecido en la primera fase debido a la morfología del clasificador. Tras varias etapas de procesamiento de la cantidad de sub-ventanas aceptadas se han reducido radicalmente con respecto a las sub-ventanas analizadas. El tratamiento de los resultados puede realizarse de cualquier forma, tal como etapas adicionales de la misma cascada (u otro utilizado en serie) o un sistema de detección alternativo.

Por ejemplo, un clasificador de primera fase excelente se puede construir a partir de un clasificador robusto de dos características reduciendo el umbral para minimizar los falsos negativos. A través de la validación del conjunto entrenado, el umbral se puede ajustar para detector el 100% con una tasa de falso positive de un 40%.

El procesamiento necesario para un clasificador de dos características puede llegar como mucho al 60% de las instrucciones del microprocesador. Parece difícil de imaginar que un filtro tan sencillo pueda llegar a niveles altos de rechazo. A modo de comparación, el escaneo de una plantilla de imagen simple, o de una sola capa, requeriría al menos 20 veces el número de operaciones por cada sub-ventana.

La estructura de la cascada refleja el hecho de que en cualquier imagen simple, la gran mayoría de las sub-ventanas son negativas. Tanto es así, que la cascada intenta rechazar el mayor número de sub-ventanas en la etapa más temprana posible. Por el

contrario, una sub-ventana positiva conlleva la evaluación de cada clasificador en la cascada, siendo un caso muy excepcional.

Tendiendo a ser parecido a un árbol de decisión, los siguientes clasificadores son entrenados usando aquellos ejemplos que han resultado positivos en las etapas anteriores. Como resultado, el segundo clasificador se encuentra con una tarea más compleja que el primero. Cuantos más complejos son los ejemplos a analizar por los clasificadores de las etapas más internas, el receptor operativo de características (ROC) se curva hacia abajo. En un ratio de detección dado, los clasificadores más internos tienen tasas de falsos positivos más altas.

d. Entrenando cascadas de clasificadores

El entrenamiento de las cascadas tiene dos tipos de compensaciones. En la mayoría de los casos, los clasificadores con más características conseguirán tasas de detección más altas y tasas de falsos positivos más bajas.

Al mismo tiempo, los clasificadores con más características requieren mayor tiempo de computación. A priori, se pueden definir un marco de optimización dependiendo de los siguientes puntos:

- Número de etapas de clasificación
- Número de características en cada etapa
- Umbral utilizado en cada etapa

A través de estos puntos, tratamos de realizar una compensación para minimizar el número de características evaluadas. Desgraciadamente es realmente difícil encontrar esta optimización en la mayoría de los casos.

En la práctica, para construir un clasificador efectivo, lo más eficiente es utilizar un marco muy sencillo. Cada etapa en la cascada reduce la tasa de falsos positivos y la tasa de detección. El entrenamiento e cada una de ellas se realiza añadiendo características hasta que se llega al objetivo de detección y la tasa establecida de falsos positivos (estos valores

se establecen a través de testeos con el detector usando un conjunto de validación). A priori no se establece un máximo de etapas, se van añadiendo etapas hasta conseguir llegar al objetivo de establecido de detección y de tasa de falsos positivos.

e. Procesamiento de la imagen

Todas las sub-ventanas utilizadas durante el entrenamiento se normalizan con la varianza para minimizar el efecto de las diferentes condiciones lumínicas y por lo tanto, esta normalización es necesaria también durante el proceso de detección. La varianza de la sub-ventana de una imagen se puede calcular rápidamente utilizando un par de imágenes integrales.

Recordemos la siguiente función:

$$\sigma^2 = m^2 - \frac{1}{N}\sum x^2$$

Teniendo en cuenta que σ es la desviación estándar, m es la media, y x es el valor del pixel dentro de la sub-ventana. La media de la sub-ventana de puede calcular usando la integral de la imagen. La suma de los pixeles cuadrados se calcula usando una imagen integral de la imagen cuadrada (ej. Dos imágenes integrales se usan en el proceso de escaneado). Durante el escaneado, el efecto de la normalización de la imagen se puede conseguir a través de la post-multiplicación de los valores de las características en vez de pre-multiplicándolos.

f. Uso del detector

Utilizamos el detector en diferentes escalas y ubicaciones a través de la imagen. El escalado se consigue no a través de la modificación de la imagen, sino escalando el propio detector. Esto hace que el proceso permita la detección de las características a cualquier escala con el mismo coste de procesamiento. Los resultados positivos se obtienen usando un conjunto de escalas con un factor de 1,25 de diferencia.

Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

Ilustración 28: Tasas de detección en diferentes umbrales de falsos positivos usando los conjuntos de prueba MIT + CMU simultáneamente (130 imágenes y 507 caras)

El escaneo se realiza también a lo largo de las diferentes localizaciones, que se obtienen deslizando la ventana un cierto número de pixeles por cada iteración. El deslizamiento está afectado por la escala del detector: si la escala actual es de *s*, la ventana se desliza [*sΔ*], donde [] es una operación de redondeo. La elección de Δ afecta a tanto la velocidad del detector como a su precisión. Los resultados que se presentan son con Δ=1.0, mientras que podríamos conseguir resultados más rápidos usando Δ=1.5, aunque esto influiría reduciendo levemente la precisión.

g. Integración de múltiples detecciones

Debido a que el último detector es insensible a pequeños cambios en traslación y escala, por norma se realizaran múltiples detecciones en cada objeto de la imagen escaneada. Lo mismo suele ocurrir con algunos tipos de falsos positivos. En la práctica suele devolver una detección fina por cada objeto. Con este fin se recomienda realizar un post-procesado de las sub-ventanas detectadas para así combinar las detecciones solapadas en una misma detección acotando el número final de sub-ventanas detectadas.

En los experimentos, las detecciones se combinan de una manera muy sencilla. Primero, se divide el conjunto de detecciones en subconjuntos totalmente diferentes, estableciendo como premisa que *dos detecciones se encuentran en un mismo subconjunto si sus regiones se solapan*. Cada partición produce una única detección final. Las esquinas de la región final de cada subconjunto es la media de las esquinas de todas las detecciones del subconjunto.

3. Seguimiento de objetos

I. Mezcla adaptativa de fondo para seguimiento en tiempo real

En vez de un modelado específico de los valores de cada pixel como un tipo de distribución, se modelan los valores de un pixel en particular como una mezcla de Gaussianas. Basándonos en la persistencia y en la varianza de la Gaussiana de cada mezcla, se determina que Gaussiana puede corresponder a los colores del fondo. Lo valores de los pixeles que no corresponden a las distribuciones del fondo son considerados parte del primer plano, hasta que aparezca una Gaussiana que los incluya con pruebas consistentes que lo apoyen.

El sistema se adapta para gestionar de forma robusta los cambios de iluminación, movimientos repetitivos en la escena, seguimiento en regiones desordenadas, objetos con movimientos lentos e introduciendo o eliminando objetos de la escena.

Los objetos con movimientos lentos tardarán más en incorporarse al fondo puesto que su color tiene una varianza mayor que el fondo. Por otro lado, las variaciones se van aprendiendo y el modelo para la distribución del fondo normalmente se mantiene igual si se reemplaza temporalmente por alguna otra distribución que nos lleve a una recuperación más rápida cuando se eliminan objetos de la escena.

El método de creación del fondo contiene dos parámetros principales:

- α: La constante de aprendizaje.
- T: La proporción de información que se han de tener en cuenta por el fondo.

Sin necesidad de más parámetros, este sistema puede utilizar en entornos de interior, aplicaciones de comunicación hombre-máquina y en entornos de exterior.

a. El método

Por cada resultado de un pixel en una superficie afectada por una iluminación específica, será suficiente una única Gaussiana para modelar el valor del pixel mientras se represente el ruido de la adquisición

Ilustración 29: Imagen original (a)

Ilustración 30: Imagen compuesta de la media de las Gaussianas en el modelo del fondo (b)

Ilustración 31: Fondo resultante de la imagen original (c)

Ilustración 32: Imagen resultado con la información del seguimiento (d)

Nota: mientras que las sombras se quedan en el primer plano en este caso, la superficie está cubierta de ellas durante un periodo considerable de tiempo. La Gaussiana que representa esos pixeles pude ser suficientemente significante para ser considerada parte del fondo

Si solo cambia la iluminación, será suficiente una única Gaussiana adaptativa por cada pixel. En la práctica suelen aparecer múltiples superficies en la vista raíz de un pixel en particular dependientes de las condiciones de iluminación. Por lo tanto, serán

necesarias varias Gaussianas adaptativas, usando una mezcla de las Gaussianas para crear una aproximación fiable.

Cada vez que se actualizan los parámetros de las Gaussianas, se evalúan utilizando una heurística simple con el fin de crear una hipótesis de adhesión al "proceso de generación de fondo". Los pixeles que no concuerden con los pixeles de la Gaussiana del fondo, se agrupan usando componentes conectadas. Finalmente se realiza el seguimiento de estas componentes entre imágenes utilizando un tracker de múltiples hipótesis, como se ilustra en los pasos (a), (b), (c), (d).

b. Modelo de mezclado en línea

Consideramos los valores que tiene de un pixel en particular durante un periodo de tiempo como "proceso de pixel", los cuales podríamos almacenar en un vector. Para un instante *t* lo que conocemos de un pixel en particular $\{x_0 , y_0\}$ es su evolución, en relación a la siguiente fórmula:

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

Dónde *I* es la secuencia de la imagen. En ocasiones, los "procesos de pixel" se pueden mostrar en una gráfica (R,G) corroborando la necesidad de utilizar sistemas adaptativos con umbrales automáticos debido a algunas dificultades que se pueden encontrar en un entorno real, como se muestra a continuación

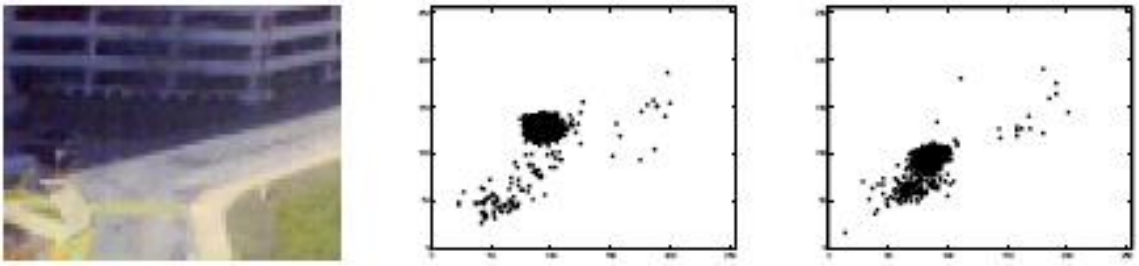


Ilustración 33: Diferencia de los pixeles en un periodo de 2 minutos

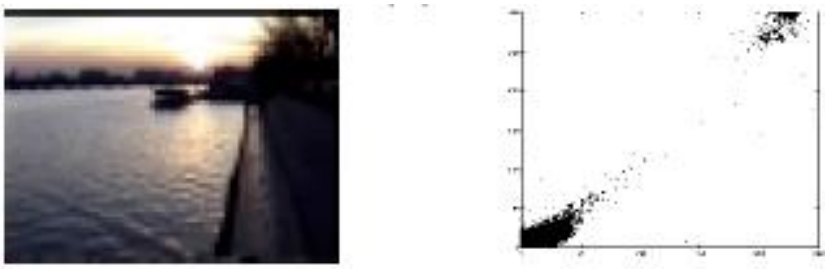


Ilustración 34: Distribución bi-modelo de los valores de los píxeles resultantes de los reflejos sobre la superficie del agua

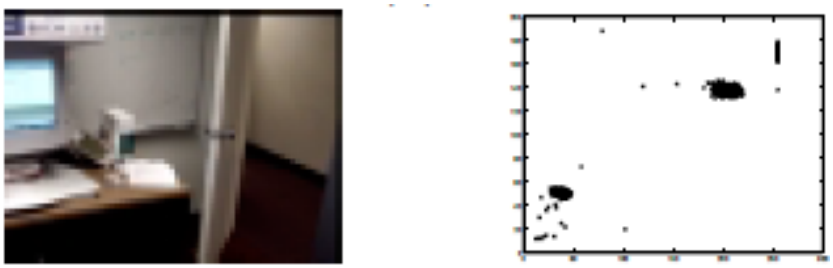


Ilustración 35: Distribución bi-modelo resultado del parpadeo del monitor

El valor de cada pixel representa la medida de la radiación y la dirección del sensor de primer objeto que entra en intersección con el pixel del rayo óptico. Con fondo e iluminación estáticos, el valor será relativamente constante. Si asumimos esta premisa, el ruido Gaussiano que incurre en el proceso de muestreo, su densidad se podría describir con una distribución Gaussiana simple centrada en el valor medio de pixel. Desafortunadamente, loa mayoría de las secuencias de video tienen cambios de iluminación, escena y objetos en movimiento. Si ocurren cambios de iluminación en la escena, es necesario que la Gaussiana tenga en cuenta dichos cambios. Si un objeto estático se incorpora a la escena y no se ha incorporado en el fondo hasta un periodo de tiempo determinado, los pixeles correspondientes seguirán formando parte del primer plano durante ese periodo de tiempo. Este último caso, puede llevar a errores en la estimación del primer plano, impidiendo la realización del seguimiento de forma correcta. Debido a este tipo de situaciones, se recomienda realizar un estudio del entorno actual para estimar los parámetros de la Gaussiana.

Otro aspecto de la variación sucede cuando hay objetos en movimiento en la escena. Incluso un objeto con color homogéneo en una imagen produce una varianza mayor que un objeto estático. Por norma, se debe obtener información adicional para dar soporte a las distribuciones del fondo ya que los pixeles de diferentes objetos normalmente no son del mismo color.

Existen algunos factores que nos pueden guiar a la hora de elegir el modelo y actualizar el procedimiento. Los valores adoptados por los pixeles más recientemente, $\{X_1, \dots, X_t\}$, se modelan a través de una mezcla de K distribuciones Gaussianas. La probabilidad de observar el valor actual del pixel se calcula a través de la siguiente formula:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

Donde K es el número de distribuciones, $\omega_{i,t}$ es un estimado del peso la i-ésima Gaussiana de la mezcla en el instante *t*, $\mu_{i,t}$ es el valor medio de la i-ésima Gaussiana de la mezcla en el instante *t*, $\Sigma_{i,t}$ es la matriz de covarianza la de i-ésima Gaussiana de la mezcla en el instante *t*, y donde η es la función de la probabilidad Gaussiana de probabilidad:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

K está definida por la capacidad de memoria y de computación. Con el fin de optimizar la computación, se calcula de la matriz de covarianza se realiza de la siguiente manera:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I}$$

Para este cálculo, asumimos que los valores RGB son independientes y que tienen la primera varianza. Esto nos permite evitar matrices de inversión que requieran mucho procesamiento penalizando en parte a la precisión.

Teniendo en cuenta lo anterior, la distribución de los valores de cada pixel recientemente observados en la escena se caracteriza por una mezcla de Gaussianas. Un nuevo valor de píxel será, en general, representado por uno de los componentes principales del modelo de mezcla y se utiliza para actualizar el modelo.

Considerando el proceso de pixel como un proceso estacionario, podemos utilizar la maximización de la expectativa, un método estándar con el fin de maximizar la probabilidad de los datos observados. Por desgracia, cada proceso pixel varía con el tiempo, así que usamos un método aproximado que esencialmente trata cada observación como un conjunto de muestras de tamaño 1 y utiliza reglas de aprendizaje estándar para integrar los datos nuevos.

Debido a la mezcla del modelo por cada pixel de la imagen, la implementación de un algoritmo exacto EM en una ventana con datos nuevos sería muy costosa. Sin embargo, se implementa una aproximación K-media en línea. Por cada pixel, X_t, es comparada con las K distribuciones Gaussianas existentes hasta que se encuentra coincidencia. Una coincidencia se define como un pixel dentro del valor 2.5 de la desviación típica de una distribución. Este umbral puede ser modificado teniendo poco efecto en el rendimiento. Esto es efectivo en un umbral de distribución pixel a pixel, siendo muy útil cuando regiones diferentes tienen iluminación distinta, ya que los objetos que aparecen en zonas sombreadas no suelen generar tanto ruido como los que se encuentran en las zonas iluminadas. Un umbral uniforme puede generar que los objetos dejen de ser detectados al entrar en zonas en sombra.

Si ninguna de las K distribuciones coinciden con el valor actual del pixel, la distribución menos probable se reemplaza por la distribución actual como su valor medio. Los pesos anteriores de las distribuciones K en el instante t, $\omega_{k,t}$, se ajustan de la siguiente manera:

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

Donde α es el valor de aprendizaje y $M_{k,t}$ es 1 para el modelo coincidente y 0 para el resto de los modelos. Después de esta aproximación, los pesos son normalizados. 1/α define el tiempo constante que determina la velocidad a la que cambian los parámetros de la distribución. $\omega_{k,t}$ es la media filtrada en paso bajo de la probabilidad posterior (umbralizada) en la que los pixeles tienen un modelo K coincidente en el intervalo de tiempo 1 a t. Esto es equivalente al valor esperado con una ventana exponencial de los valores anteriores.

Los parámetros μ y σ de las distribuciones no coincidentes permanecen intactos. Los parámetros de la distribución que coinciden en la nueva observación se actualizan de la siguiente manera:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t$$
$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t)$$

Donde

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$$

El cuál es el mismo tipo de filtro de paso bajo mencionado anteriormente, exceptuando que solo los datos que coinciden en el modelo son incluidos en la estimación.

Una de las principales ventajas de este método es que cuando se identifica algo que vaya a pertenecer al fondo, no destruye el modelo actual del fondo. El color del fondo original se mantiene en la mezcla hasta que forma parte de la K-ésima más probable. Por esta razón, si un objeto es estático suficiente tiempo como para tomar parte del fondo y posteriormente se mueve, seguirá existiendo en el fondo anterior con el mismo μ and σ^2, aunque con un ω menor y se podrá reincorporar rápidamente al fondo.

c. Modelo de estimación de fondo

Como parte de los parámetros del modelo de la mezcla de los cambios de cada pixel, debemos determinar cuáles Gaussianas de la mezcla tienen mayor probabilidad de ser generadas en el proceso de creación de fondo. Heurísticamente, estamos interesados en que las distribuciones Gaussianas que tienen el mayor número de evidencias de apoyo y la mínima varianza.

Para poder entender esta elección, consideramos la acumulación de evidencias de apoyo y la relativa baja varianza como la distribución de fondo cuando es estático, es decir, el objeto estático es visible. Por el contrario, cuando un objeto nuevo oculta el objeto de fondo, no coincidirá con ninguna de las distribuciones existentes, lo que no creará una nueva distribución ni incrementara la varianza de la distribución actual. Se espera que la varianza del objeto en movimiento se mantenga mayor al fondo hasta que el objeto en movimiento se pare. Para añadir al modelo esto, necesitamos un método para decidir que porción del modelo de la mezcla representa mejor el proceso de fondo. Primero, se ordenarán las Gaussianas según su valor ω/σ, Este valor incrementan tanto cuando a la distribución consigue más evidencias como cuando su varianza decrece. Des pues de reestimar los parámetros de la mezcla, con ordenar las distribuciones en relación a la que tenga mayor probabilidad de pertenecer al fondo., ya que solo los valores relativos de los modelos coincidentes habrán cambiado.

As the parameters of the mixture model of each pixel change, we would like to determine which of the Gaussians of the mixture are most likely produced by background processes. Heuristically, we are interested in the Gaussian distributions which have the most supporting evidence and the least variance. En este orden de los modelos, los que sean más cercanos a las distribuciones de fondo, se mantendrán a principio, mientras que los menos probables estarán más al fondo, y en muchos casos reemplazados por nuevas distribuciones.

Entonces, se escogen las primeras distribuciones B como modelo de fondo donde:

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right)$$

T es la medida de la porción mínima de datos que se deben tener en cuenta para el fondo. Esta fórmula selecciona las mejores distribuciones hasta que una porción (T) de datos recientes se tiene en cuenta. Si escogemos un valor bajo para Y, le modelo de fondo es normalmente unimodal. En este caso, usando únicamente la distribución más probable ahorraremos en procesamiento.

Si T es un valor alto, una distribución multimodal originada por movimiento repetitivo en el fondo (ej. Un árbol, una bandera con viento…) puede tener como resultado la inclusión de un color en el modelo de fondo. Este resultado en un tiene un efecto transparente que permite al fondo aceptar dos o más colores separados.

d. Componentes conectadas

El método descrito anteriormente nos permite identificar los pixeles del primer plano por cada nueva imagen, actualizando la descripción en cada proceso de pixel. Estos pixeles de primer plano identificados se pueden segmentar en regiones usando un algoritmo de componentes conectadas de dos pasos.

Debido a que este procedimiento es efectivo para determinar el movimiento de un objeto competo, las regiones en movimiento se pueden caracterizar no solo por su posición, sino también por su tamaño, momentos y cualquier información que podamos obtener de la forma. Podemos aprovechar esta información no solo para procesamiento posterior y clasificación, sino también puede ayudar al proceso de seguimiento.

e. Seguimiento con multiples hipótesis

Teniendo en cuenta que esta sección no es esencial para el método de sustracción de donde, nos permite comprender y evaluar mejor los resultados de las siguientes secciones. Estableciendo una correspondencia de componentes conectadas entre las

imágenes conseguimos usar algoritmo predictivo de seguimiento basado en múltiples hipótesis, que incorpora tanto la posición como el tamaño. Hemos implementado un método en línea para poder rellenar y mantener conjuntos de filtros Kalman. En cada imagen, tenemos un grupo de modelos de calman y un nuevo grupo de componentes conectadas que los complementan. Primero, estos modelos son probabilísticamente coincidentes a las regiones conectadas que los complementan. Segundo, las regiones conectadas que no están suficientemente complementadas, son comprobadas con el fin de encontrar nuevos modelos de Kalman. Finalmente, aquellos modelos que no entren (determinados por la inversa de la varianza de su error de predicción) dentro del umbral, son eliminadas.

La comprobación de los modelos con las componentes conectadas requiere comprobar cada modelo existente con el grupo de componentes conectadas existentes que son mayores a uno y dos pixeles. Todas las coincidencias se utilizan para actualizar el modelo correspondiente. Si el modelo actualizado tiene suficiente información, se utilizará en la siguiente imagen que se reciba. Si no se encuentran coincidencias, podemos tomar como hipótesis un valor nulo que se propagara por el modelo reduciendo la información en un valor constante. Los modelos no coincidentes de la imagen actual y de las dos anteriores, se utilizaran para crear hipótesis de los modelos nuevos al igual que pares de componentes conectadas no coincidentes en las mismas imágenes. Si la imagen actual contiene una coincidencia con suficiente información, el modelo actualizado se añade a los modelos existentes. Para evitar posibles explosiones de combinaciones en situaciones de ruido, es deseable limitar el número máximo de modelos existentes recudiendo aquellos modelos menos probables cuando existan demasiados modelos. En situaciones ruidosas (ej. Cámaras CCD en condiciones de baja iluminación…), normalmente es útil eliminar toda información se seguimientos cortos.

II. Seguimiento de objetos a través de histograma de color usando un algoritmo tipo EM

a. Mean-Shift como un algoritmo tipo EM

Si para cada punto usamos un factor de peso ω_i, utilizaremos la siguiente fórmula

$$f(\vec{\theta}, V) = \sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}, V).$$

Buscaremos resolver los parámetros $\vec{\theta}$ y V para obtener el mayor resultado posible de la anterior función. Esto se puede conseguir de forma iterativa usando iteraciones tipo EM. Para la desigualdad de Jensen obtenemos:

$$\log f(\vec{\theta},V) \geq G(\vec{\theta},V,q_1,\ldots,q_N) = \sum_{i=1}^{N} \log\left(\frac{\omega_i \mathcal{N}(\vec{x}_i;\vec{\theta},V)}{q_i}\right)^{q_i}$$

Donde $q_i - s$ son constantes arbitrarias que cumplen con el siguiente requisito:

$$\sum_{i=1}^{N} q_i = 1 \text{ and } q_i \geq 0.$$

Asumiremos que los valores estimados actuales para los parámetros están marcados por $\vec{\theta}^{(k)}$ y por $V^{(k)}$. Los pasos E y M descritos posteriormente se repiten hasta que conseguimos la convergencia:

1. Paso E: encontrar $q_i - s$ para maximizar G mientras mantenemos fijos $\vec{\theta}^{(k)}$ y $V^{(k)}$. Para acotar este paso, podemos encontrar el valor máximo a través de la siguiente fórmula:

$$q_i = \frac{\omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}.$$

2. Paso M: Maximizar *G* con respecto a $\vec{\theta}$ y a V mientras que mantenemos $q_i - s$ constante. Los valores de $q_i - s$ se fijan para minimizar una parte de *G* que depende de los siguientes parámetros:

$$g(\vec{\theta}, V) = \sum_{i=1}^{N} q_i \log \mathcal{N}(\vec{x}_i; \vec{\theta}, V).$$

A partir de $\frac{\partial}{\partial \vec{\theta}} g(\vec{\theta}, V) = 0$ obtenemos:

$$\vec{\theta}^{(k+1)} = \sum_{i=1}^{N} q_i \vec{x}_i = \frac{\sum_{i=1}^{N} \vec{x}_i \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}$$

Hemos de tener en cuenta que esta función de actualización para la posición estimada es equivalente a la función de actualización "Mean Shift" de los kernels Gaussianos. Para otro tipo de kernels, esta función puede variar. Este nuevo punto de vista tipo EM del problema puede llevar a funciones de actualización para *V* como se describen a continuación.

b. Selección de la escala

Siendo $p^*(\vec{x})$ es una distribución de datos reales, el resultado esperado de la Gaussiana maximizada es:

$$E\left[f(\vec{\theta}, V)\right] = \int_{\vec{x}} p^*(\vec{x}) \mathcal{N}(\vec{x}; \vec{\theta}, V).$$

Esto se puede ver como una versión suavizada del p^* original y el maximo respecto a *V* cuando no tiene alguna condición deseada. Por ejemplo, si p^* es una Gaussiana local $N(\vec{x};\vec{\theta};V^*)$, el valor esperado será $N(\vec{x};\vec{\theta};V^*+V)$. El máximo esperado para $\vec{\theta}=\vec{\theta}^*$, aunque desafortunadamente, el valor tribial para *V = 0* teniendo en cuenta que el valor en el modelo local decrece cuando *V* es mayor. Normalizamos el resultado a base de multiplicar la densidad estimada por $|V|^{\gamma/2}$, el cual lo obtenemos cuando llamamos a la función "γ-normalizada":

$$f_\gamma(\vec{\theta},V)=|V|^{\gamma/2}f(\vec{\theta},V).$$

Bajo la misma suposición en la que el modo local es próximo a la Gaussiana, el valor del modo será proporcional a $|V|^{\gamma/2}/|V^*+V|^{1/2}$. El máximo respecto a *V* se encuentra en:

$$\frac{\partial}{\partial V}\frac{|V|^{\gamma/2}}{|V^*+V|^{1/2}}=0$$

Siendo $\frac{\partial}{\partial V}|V|=|V|\,[2V^{-1}-diag(V^{-1})]$ obtenemos:

$$\gamma|V|^{\gamma}\left[2V^{-1}-\mathrm{diag}(V^{-1})\right]|V^*+V|$$
$$-|V|^{\gamma}|V^*+V|\left[2(V^*+V)^{-1}-\mathrm{diag}((V^*+V)^{-1})\right]=0.$$

A través de esto sacamos que $\gamma V^{-1}=(V^*+V)^{-1}$ y $V=\frac{\gamma}{1-\gamma}V^*$. Obviamente solo cuando $\gamma\epsilon(0,1)$ obtenemos un valor positivo. Cuando $\gamma=1/2$ podemos esperar que $V=V^*$. La solución de usar la función "γ-normalizada" no está sesgada siendo esto una propiedad deseada en la estimación de un algoritmo.

El algoritmo iterativo tipo EM se puede aplicar a la función de normalización. La única diferencia es en el paso M, en el que en tenemos la siguiente función:

$$g(\vec{\theta}, V) = \sum_{i=1}^{N} q_i \log |V|^{\gamma/2} \mathcal{N}(\vec{x}; \vec{\theta}, V).$$

La función de actualización de la posición se mantiene intacta. Desde $\frac{\partial}{\partial \vec{\theta}} g(\vec{\theta}, V) = 0$ es fácil mostrar que la función de actualización para *V* en el paso M está determinada por:

$$\vec{V}^{k+1} = \beta \sum_{i=1}^{N} q_i (\vec{x}_i - \vec{\theta}^{(k)})(\vec{x}_i - \vec{\theta}^{(k)})^T,$$

Donde $\beta = 1/(1-\gamma)$.

En las siguientes ilustraciones podemos ver un ejemplo del rendimiento del nuevo algoritmo. La simulación consiste en 600 muestras generadas usando una mezcla de tres distribuciones Gaussianas. Estos tres modelos son claramente visibles en las ilustraciones. Las iteraciones (contornos 2-sigma de la Gaussiana estimada) en el procedimiento Mean Shift lo podemos observar en la primera ilustración mientras que en la segunda ilustración podemos ver el nuevo algoritmo tipo EM con $\gamma = 1/2$ $(\beta = 2)$. Podemos también observar como el nuevo algoritmo, de forma simultánea, estima ambas posiciones tanto en el modo local como en la matriz de covarianza que describe la forma del modo. Hemos de tener en cuenta que $(\beta = 2)$ es apropiado si la distribución subyacente es Gaussiana. Si cualquier otra distribución se aproxima a la Gaussiana, se necesitara asignar otro valor a β para evitar una solución sesgada.

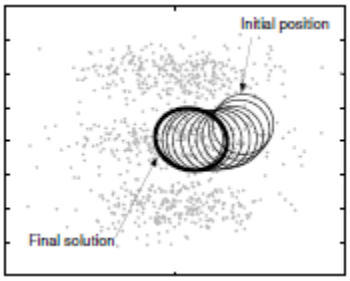


Ilustración 36: Rendimiento usando Mean Shift en una simulación 2D

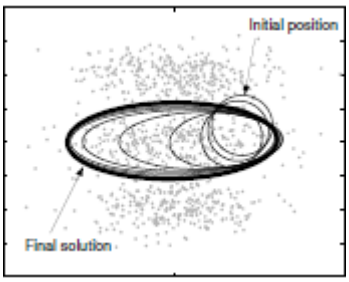


Ilustración 37: Rendimiento usando EM Shift en una simulación 2D

c. Seguimiento por histograma de color

Se asume que la forma de un objeto no rígido se puede aproximar a una región elíptica de la imagen. Inicialmente el objeto se selecciona de forma manual o usando cualquier otro algoritmo, como puede ser el de sustracción de fondo. Siendo $\overrightarrow{x_i}$ la posición del pixel y $\overrightarrow{\theta_0}$ la posición inicial del centro del objeto en la imagen. El segundo momento se puede utilizar para aproximar la forma del objeto:

$$V_0 = \sum_{\text{all the pixels that belong to the object}} (\vec{x}_i - \vec{\theta}_0)(\vec{x}_i - \vec{\theta}_0)^T$$

Además, el histograma de color se utiliza para modelar la apariencia del objeto. Teniendo el histograma M contenedores y siendo la función $b(\overrightarrow{x_i})$: $R^2 \rightarrow 1, \dots, M$ la que asigne el valor del color del pixel en $(\overrightarrow{x_i})$ de su contenedor. El modelo de color del histograma del objeto consiste entonces en los valores de M en los contenedores del

histograma $\vec{o} = [o_1, \dots, o_m]^T$. El valor del m-ésimo contenedor se calcula a través de la siguiente función:

$$o_m = \sum_{i=1}^{N_{V_0}} \mathcal{N}(\vec{x}_i; \vec{\theta}_0, V_0)\delta\left[b(\vec{x}_i) - m\right],$$

Donde δ es la función delta de Kronecker. Usamos el Kernel Gaussiano *N* para darle más importancia a los pixeles que se encuentran en medio del objeto y darle menos peso de confianza a los valores que se encuentran en los bordes de los objetos. Usamos solo los pixeles N_{V_0} vecinos del kernel y los pixeles que estén desagregados por más de 2.5 sigma.

d. Medición de la similitud

Asumimos que tenemos una imagen nueva en una secuencia de imágenes y que el objeto que estamos siguiendo está presente en la imagen. El objetivo del algoritmo de seguimiento es encontrar el objeto en la nueva imagen. Supongamos una región elíptica definida en la imagen nueva con posición θ y conforma descrita por la matriz de covarianza *V*. El histograma de color que describe su aspecto en la región es $\vec{r}(\vec{\theta}, V)$ y el valor del m-ésimo contenedor se calcula a través de la siguiente formula:

$$r_m(\vec{\theta}, V) = \sum_{i=1}^{NV} \mathcal{N}(\vec{x}_i; \vec{\theta}, V)\delta\left[b(\vec{x}_i) - m\right]$$

La similitud en la región del objeto se define por la similitud entre sus histogramas, como se hace con el coeficiente de Bhattacharyya :

$$\rho\left[\vec{r}(\vec{\theta}, V), \vec{o}\right] = \sum_{m=1}^{M} \sqrt{r_m(\vec{\theta}, V)}\sqrt{o_m}.$$

La aproximación del primer orden de Taylor alrededor del estimado $\vec{r}(\overrightarrow{\theta^{(k)}}, V^{(k)})$ usando

$$\rho\left[\vec{r}(\vec{\theta},V),\vec{o}\right] \approx c_1 + c_2 \sum_{i=1}^{N_V} \omega_i \mathcal{N}(\vec{x}_i;\vec{\theta},V)$$

Donde c1 y c2 son factores constantes y

$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{r_m(\vec{\theta}^{(k)},V^{(k)})}}\delta\left[b(\vec{x}_i)-m\right]$$

De forma práctica y tomando el modelo de objeto $\vec{o}$, su inicio *(k=0)* posición $\overrightarrow{\theta^{(k)}}$ y su forma definida por $V^{(k)}$, el algoritmo funciona de la siguiente manera:

1. Calcular los valores del histograma de color en la región actual definida por $\overrightarrow{\theta^{(k)}}$ y por $V^{(k)}$en la imagen actual.
2. Calcular los pesos
3. Calcular $q_i - s$
4. Calcular la nueva posición estimada $\overrightarrow{\theta^{(k+1)}}$
5. Calcular la nueva varianza $V^{(k+1)}$
6. En el caso en que no se incluyan nuevos pixeles en la región elíptica definida por las nuevas estimaciones $\overrightarrow{\theta^{(k+1)}}$ y $V^{(k+1)}$, se termina el proceso, en caso contrario se establece *k = k + 1* y se vuelve al paso inicial.

Este proceso se repite por cada imagen. En su forma más simplificada, la posición y la forma de la región elíptica de la imagen anterior se utilizan como los valores iniciales de la nueva imagen.

Propuesta

Teniendo en cuenta los apartados anteriores, en los que se describen los algoritmos existentes y propuestos, a continuación se describe brevemente cómo se trata la información extraída en cada paso.

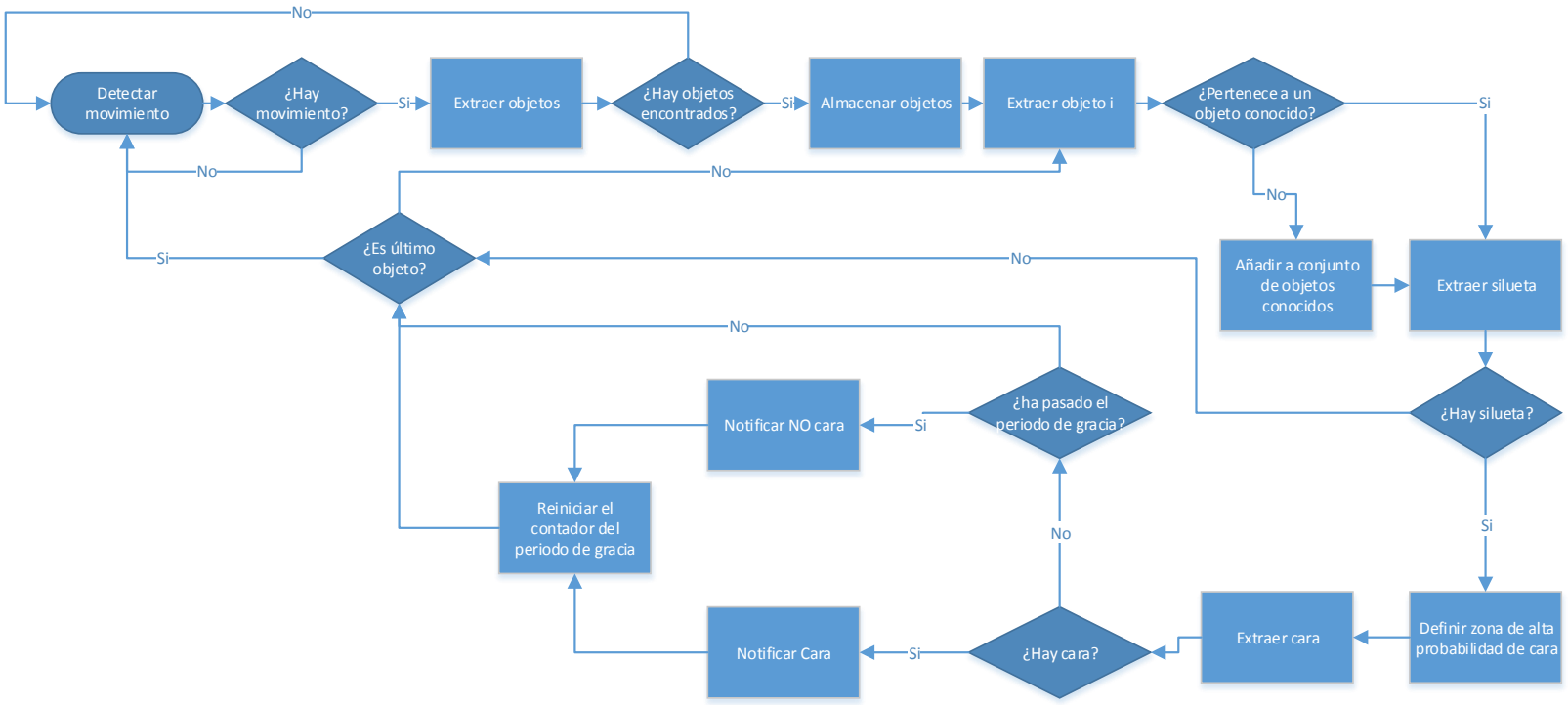


Diagrama 1: Proceso completo

1. Detección de movimiento y extracción de histograma

El proceso se inicia con la detección de movimiento, que es quien genera el desencadenante del proceso. Si se detecta suficiente movimiento (cumpliendo parámetros como el umbral para definir la cantidad de movimiento), procedemos a la extracción de los objetos. Durante la extracción de los objetos, se detectan los diferentes segmentos en la imagen recibida para ver si forman entre sí objetos cerrados, con el fin de poder realizar un primer filtro, en caso negativo, volveremos al estado inicial, descartando los segmentos detectados, por contrario, en caso afirmativo, procederemos al almacenamiento de dichos objetos.

Una vez pre-filtrados los segmentos y extraídos los objetos, procedemos a evaluarlos si son objetos conocidos o no comparando su histograma (seguimiento a partir de histograma de color). En caso de no ser conocidos, lo añadiremos como un nuevo objeto en nuestro conjunto de objetos conocidos. Tras haber sido identificados o añadidos dentro de nuestro conjunto de objetos conocidos, ya podemos proceder al paso final de esta fase, la detección de la silueta.

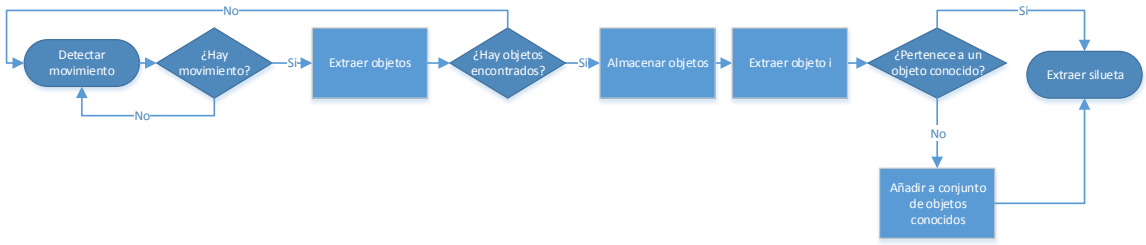


Diagrama 2: Detección de movimiento y extracción de histograma

2. Detección de personas basado en entrenamiento

En esta segunda fase, la labor principal es poder descartar todos aquellos objetos que no cumplen con las características de ser la silueta de una persona. Como se ha explicado anteriormente, realizamos una serie de cálculos para extraer únicamente los bordes del objeto dentro del marco de referencia. Una vez encontrada la silueta inicial, buscamos crear el esqueleto, a través de las distancias máximas de los bordes al centro del objeto. Si cumple con las características definidas, pasaremos al estado final de esta fase, la definición de la zona de alta probabilidad de cara. En caso contrario tendremos que descartar el objeto.

Si el objeto que estábamos analizando es el último objeto del conjunto de objetos de la imagen, volveremos al primer estado de la primera fase del proceso, la detección de movimiento. Si por el contrario no es el último objeto pasaremos analizar el siguiente objeto en búsqueda de una silueta.

Diagrama 3: Detección de personas basado en entrenamientos

3. Detección facial en zona de alta probabilidad

En esta tercera y última fase comenzamos con la definición de la zona de alta probabilidad de cara, para ello nos basamos en las proporciones del Hombre de Vitrubio. Según el Hombre de Vitrubio, la cabeza corresponde a 1/8 parte del cuerpo, un 12,5%. Para otorgarnos un margen de error en proporciones, se toma el 20% superior de la silueta que recibimos de la fase anterior como zona de alta probabilidad.

Una vez definida esta zona, procedemos a procesar la cascada de características en búsqueda de la cara. En caso de encontrar la cara, notificaremos de su localización y su marca de tiempo para que se de validación de acceso y se registre en el sistema de seguridad, además de reiniciar el contador de periodo de gracia. En caso contrario, revisaremos la marca de tiempo con la última marca de tiempo registrada, para comprobar si se ha excedido el tiempo máximo de detección de la cara. En caso de exceder el tiempo, se notificara la "no cara" o ausencia de cara encontrada con la región de alta probabilidad de cara y la marca de tiempo al sistema de seguridad. Posteriormente se reinicia el contador del periodo de gracia. Para terminar se comprueba si es el último objeto de conjunto detectado en esa imagen para proceder al siguiente o volver al estado inicial de la primera fase.

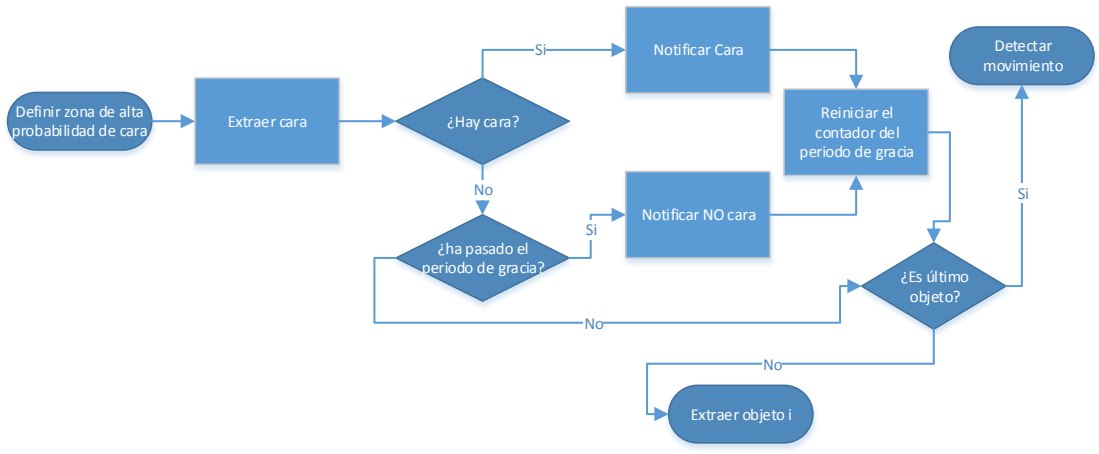


Diagrama 4: Detección facial en zona de alta probabilidad

4. Propuesta técnica

I. Detección de movimiento y extracción de histograma para seguimiento

El procedimiento iterativo Mean Shift es un método simple y robusto para encontrar la posición del máximo local de un estimado basado en Kernel de una función de densidad. El seguimiento de objetos basado a través de histograma de color basado usando un algoritmo tipo EM es un procedimiento evolucionado del Mean Shift. Este algoritmo estima simultáneamente la posición y la matriz de covarianza que del máximo local. Aplicando este algoritmo para desarrollar un histograma de color con 5 grados de libertad basado en un el algoritmo de seguimiento de objetos no rígidos.

El histograma de color es una representación muy robusta de la apariencia de un objeto. La forma de un objeto no rígido se representa con una elipse. La función de similitud está definida entre el histograma de color y el histograma de color de una región elíptica de un candidato elipsoidal de una nueva imagen correspondiente a una secuencia. El procedimiento Mean Shift se utiliza para encontrar la región en la nueva imagen que tiene el objeto más similar al comparado.

El problema de la adaptación de la elipse que se aproxima a la forma del objeto, es que cuando la forma y el tamaño del objeto cambian, no se consigue resolver. Algunos descriptores de forma locales se utilizan en cada fase de seguimiento adaptando la elipse a un ±10% de su tamaño y seleccionando la mejor. En una búsqueda extensa, se procesa la búsqueda con escala variable, pudiendo ser un máximo del 0,1% de la imagen al 100% de la imagen. A continuación se muestran ejemplos de los resultados que se obtienen al usar los diferentes tipos de escalas:

Ilustración 38: Escala fija

Ilustración 39: Escala ±10%

Ilustración 40: Escala variable

En este caso, en vez de solo estimar la posición del máximo local, es estima simultáneamente la matriz de covarianza que describe su forma.

Esto lo podemos observar en la ilustración “Rendimiento usando EM Shift en una simulación 2D“. Se propone el uso de un histograma de color de 5 grados de libertad con el fin de estimar además de la posición del objeto, la elipse que aproxima la forma del mismo.

El problema de adaptación de la elipse se soluciona como podemos observar en las secciones a y b del algoritmo para posteriormente aplicar el histograma de color basado en el seguimiento en la sección c.

Se ha hecho pruebas del algoritmo con diferentes entornos para poder ver su sensibilidad tanto a escalas como a rotaciones.

En el siguiente ejemplo, podemos observar el rendimiento del algoritmo en situaciones de rotación. Se seleccionó un jugador, indicado con la elipse en la ilustración de “Región seleccionada”. Para una presentación mejor, se incrementó el brillo en las imágenes. Con el fin de probar el seguimiento, se escaló a 1.5 veces y se rotó 45º, como se puede ver en las ilustraciones resultado. Se realizaron pruebas tanto con el algoritmo

Mean Shift, como con el descrito anteriormente tipo EM. Cada elipse que aparece dibujada, es la que indica la nueva posición del mismo contenido seleccionado en la región inicial. Como podemos observar, el nuevo algoritmo consigue una mejor adaptación a las rotaciones.

Ilustración 41: Región seleccionada

Ilustración 42: Búsqueda usando Mean Shift

Ilustración 43: Búsqueda usando EM

En el segundo ejemplo, se pretende demostrar el rendimiento del algoritmo en un video de baja calidad con diferencias de escalado. Se ha utilizado solo las componentes H y S de HSV para conseguir ser más robustos frente a cambios de luz (contraluz de fondo).

Los objetos se representaron usando un histograma de 8x8 en el espacio HS. Debido a que los objetos eran personas, no se tuvieron en cuenta elipses con orientación diferente a la vertical. Por ello, se limitó V a la diagonal.

Las dos imágenes que representan la situación típica son las que se muestran en las ilustraciones a continuación. Debido a que el objeto se mueve en dirección a la cámara, el tamaño del objeto cambia considerablemente por lo que este seguimiento sería imposible de realizar con el algoritmo estándar Mean Shift, puesto que es muy sensible a cambios de tamaño. Se utilizó una secuencia similar en *Mean Shift Blob Tracking through Scale Space* aunque con resultados mucho más lentos.

Ilustración 44: Instante inicial de seguimiento en pasillo

Ilustración 45: Instante final de seguimiento en pasillo

Por último, se realizaron pruebas con una secuencia de movimientos rápidos. En esta secuencia se realizó el seguimiento de una mano para demostrar el seguimiento basado en un histograma de color de 5 dominios de libertad. Con el fin de ser robustos también con los cambios de iluminación, se utilizó un histograma de 8x8 en el espacio de color HS.

Se hizo el seguimiento de la mano durante 250 imágenes con cambios de forma y posición rápida. En la siguiente ilustración podemos ver como el algoritmo fue capaz de adaptarse a estos cambios y realizar el seguimiento. Esta prueba se realizó también con el algoritmo Pfinder (*Pfinder: Real-Time Tracking of the Human Body*), aunque el uso de ese algoritmo no resultó demasiado robusto ya que sólo permite el seguimiento de un único objeto de color.

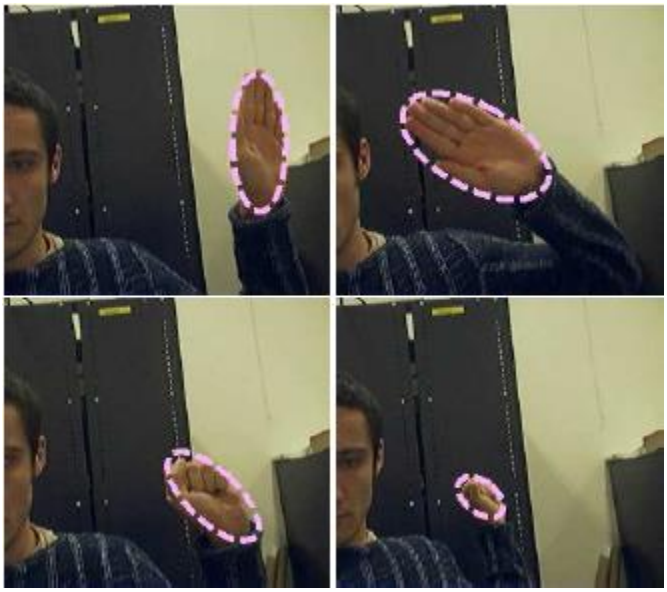


Ilustración 46: Secuencia de seguimiento de la mano. Instantes: 0, 100, 200, 250

En la siguiente gráfica, podemos observar el número de iteraciones del algoritmo en la secuencia de la mano. La media de iteraciones por imagen es 6, superando en 2 iteraciones a los resultados obtenidos utilizando Mean Shift *(Real-time tracking of non-rigid objects using mean shift)*. La complejidad de computación de este algoritmo es sensiblemente mayor que Mean Shift, llegando a ser 2 veces más lento, aunque suficientemente rápido para su uso en tiempo real. Las pruebas del algoritmo fueron realizadas en un PC con un procesador de 1GHz.

II. *Detección de personas basado en entrenamiento*

Como hemos podido ver con anterioridad, para la detección de objeto nos basamos en la idea del primer plano y fondo. La idea principal una detección de regiones en un primer plano se fundamenta en la sustracción adaptativa del fondo manteniendo los valores estadísticos del fondo, explicado en la sección del estado del arte. Los objetos del primer plano se extraen del fondo en cada imagen de la secuencia de video siguiendo un proceso de cuatro fases: umbralización, limpieza de ruido, filtros morfológicos y detección de objetos.

Las personas tenemos diferente forma, apariencia y patrones de movimiento comparado con otros objetos como coches, animales… Se puede usar un análisis estadístico de forma, como el porcentaje de aspecto, área, tamaño del perímetro o análisis de movimiento, como velocidad o periodicidad del movimiento, con el fin de distinguirnos de otro tipo de objetos. Todas las claves en el análisis estadístico de formas se pueden obtener fácilmente de una serie de datos binarios extraídos de la umbralización y de los parámetros de un marco de referencia, como se observa en la siguiente ilustración.

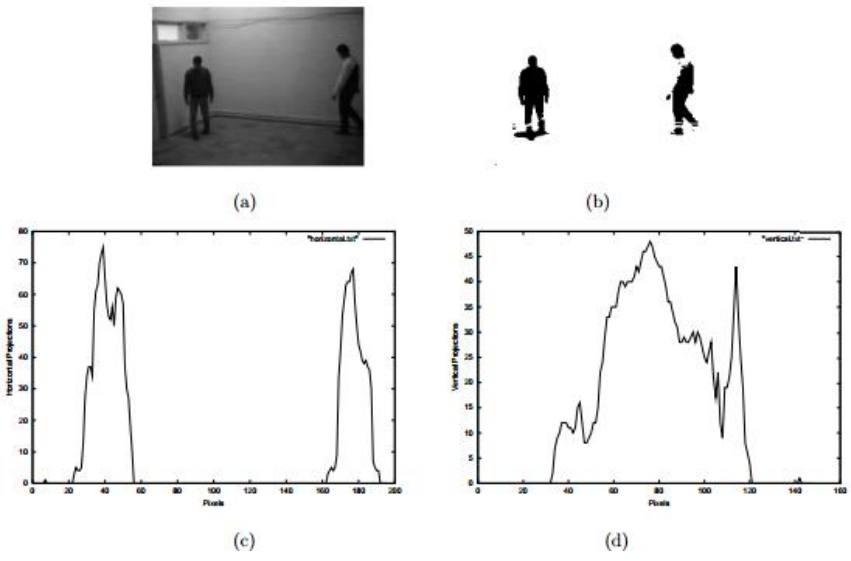


Ilustración 47: a) imagen inicial, b) regiones detectadas, c) proyecciones horizontales, d) proyecciones verticales

Pero las características pueden ser producidas por el análisis dinámico de movimiento, el cual contiene información más fiable del clasificador, como puede ser el movimiento siguiendo un clasificador de silueta o esqueleto.

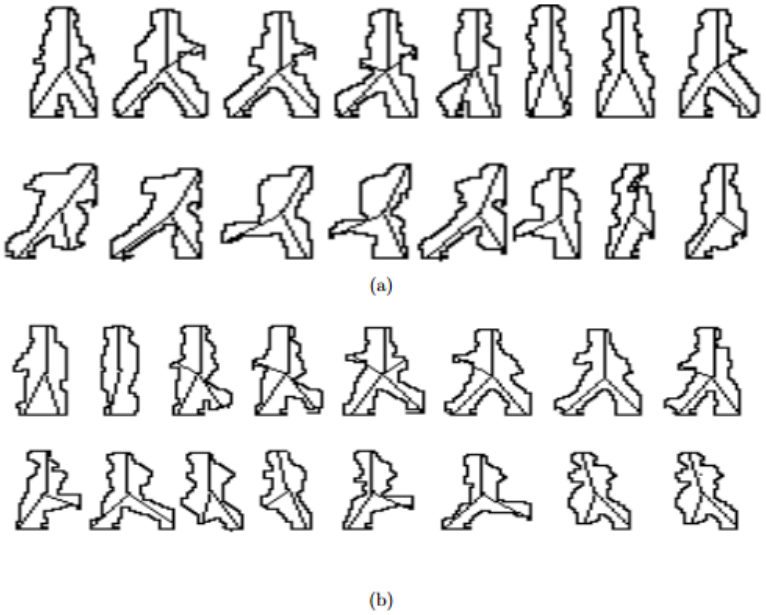


Ilustración 48: Secuencias de silueta y esqueleto andando y corriendo respectivamente: a) hacia la derecha, b) hacia la izquierda

Durante el proceso de extracción de los objetos del primer plano, pueden existir puntos, agujeros o efectos del entrelazado del vídeo que producen distorsión en la imagen y que dificultan por tanto el correcto procesamiento. Por esta razón, el primer paso del pre-procesado para sacar el esqueleto es limpiar las anomalías encontradas en las regiones detectadas. Esto se implementa con un filtro morfológico (dilatación seguida de erosión). Se realiza una doble dilatación seguida de una erosión simple, esto consigue eliminar todos los pequeños agujeros que se puedan encontrar en la silueta y suaviza sus contornos. Tras este proceso, se realiza la extracción de la silueta usando un algoritmo de extracción de bordes, como se muestra a continuación.

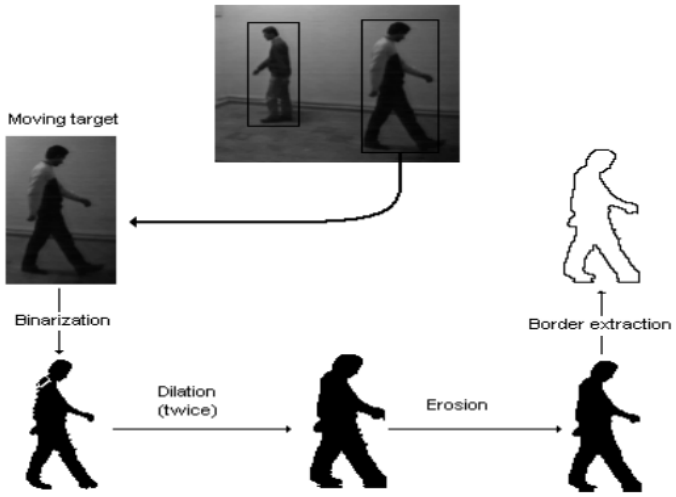
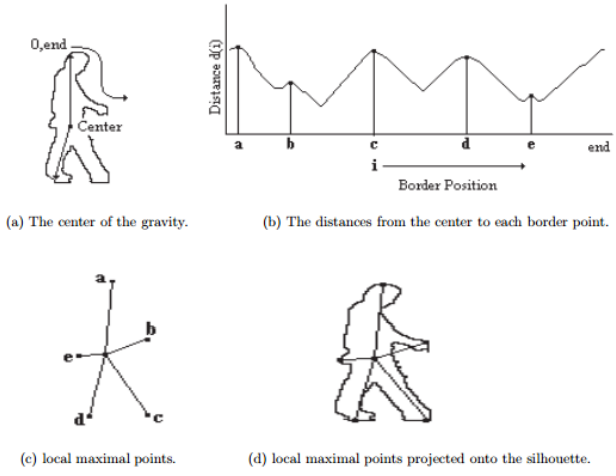


Ilustración 49: Proceso de extracción de silueta

Tras la extracción de la silueta, se construye el esqueleto conectando el centro de la región a las diferentes extremidades de la misma. Este procedimiento se basa en cuatro pasos básicos: extracción del centro de gravedad, cálculo las distancias del centro con el borde, localización de los puntos de inflexión en el contorno, proyección de los puntos de inflexión sobre la silueta.



(a) The center of the gravity.

(b) The distances from the center to each border point.

(c) local maximal points.

(d) local maximal points projected onto the silhouette.

Ilustración 50: Proceso de creación del esqueleto

Las características en una forma estática se miden directamente usando la silueta y su marco de referencia, siendo estas: ratio de aspecto en el marco de referencia, los ejes en el segundo momento de la silueta y la dispersión en el marco de referencia (perimetro^2 / área). En las características de una forma dinámica también se producen con el esqueleto de una silueta detectada en una región. Tanto la estructura del esqueleto como la su repetitivo cambio a lo largo del tiempo, nos dan importantes características para el análisis de diferentes objetos. En las tres siguientes ilustraciones podemos observar el esqueleto en diferentes tipos de objetos (personas, grupo de personas y coche). Si nos fijamos en los cambios a lo largo del tiempo de los esqueletos, podemos acabar determinando que alguna de esas siluetas contiene las características registradas en nuestro clasificador, como es el caso de la primera ilustración.

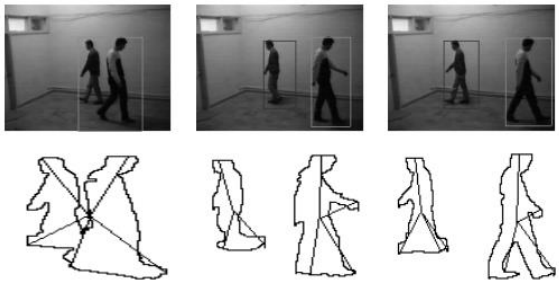


Ilustración 51: Generación de esqueleto - Personas

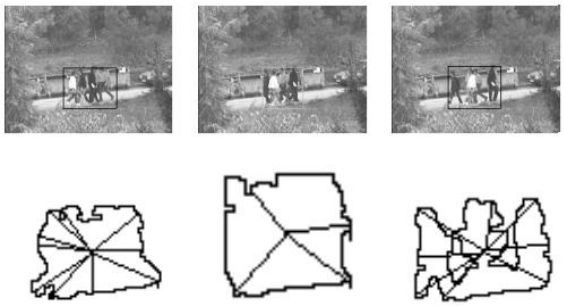


Ilustración 52: Generación de esqueleto - Grupo de personas

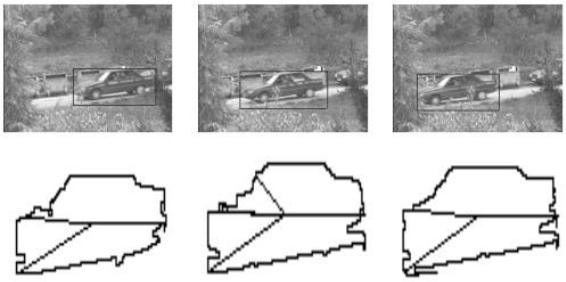


Ilustración 53: Generación de esqueleto - Coche

Las distancias máximas que hemos encontrado en el proceso de generación del esqueleto, nos pueden ayudar a determinar si la silueta encontrada pertenece al modelo de una persona o no, ayudándonos así a descartar otros objetos. Para una clasificación más precisa, se pueden considerar otras características del movimiento normal de las personas. Un movimiento determinado puede ser determinado por la auto-similitud de las características de las siluetas a lo largo del tiempo usando sus esqueletos. Como resultado, podemos obtener características de forma estática usando un análisis dinámico de periodicidad que combinados distingan una persona de objetos, como puede ser un grupo de personas, que no cumplen un patrón de esqueleto único, pero si un patrón de comportamiento característico que permite diferenciarlo del comportamiento de otros objetos, como puede ser un coche.

La identificación de las personas no se basa únicamente en el esqueleto de la silueta, sino también en las variaciones del mismo en un corto periodo de tiempo, como hemos podido observar en las tres últimas ilustraciones. Estas muestran la variación de los esqueletos producida en la silueta a lo largo del tiempo en una persona, grupo de personas y en un coche. En un instante en particular, y debido a los patrones de movimiento, es posible que tengamos una estructura similar en la persona y en el grupo de personas. Pero analizando las variaciones de la estructura del esqueleto y otros parámetros, como el contorno, podemos obtener resultados más robustos sobre la identificación de personas.

Para el seguimiento de objetos (persona y grupo de personas), después de detectar un objeto, el algoritmo de seguimiento calcula el marco de referencia, el centro y la correspondencia de dicho objeto sobre el resto de las imágenes. En una situación

óptima, el algoritmo de seguimiento consigue distinguir entre los diferentes objetos tras suceder oclusiones entre los mismos. Cuando aparece un gripo de personas, se detectan como un mismo objeto, pero en cuanto una persona se separa del grupo se empieza a hacer un seguimiento individual.

Para el análisis del movimiento, una aproximación similar de hace en *Automatic Spatio-Temporal Video Sequence Segmentation* aunque más fiable ya que se añaden características más importantes como el paso normal o corriendo a las acciones de la escena vigilada, con el fin de diferenciar entre cada acción. Para analizar estas dos acciones se implementan básicamente cuatro parámetros: θ, α, β, variaciones de aceleración en el centro de movimiento del blob que son representados en el siguiente diagrama:

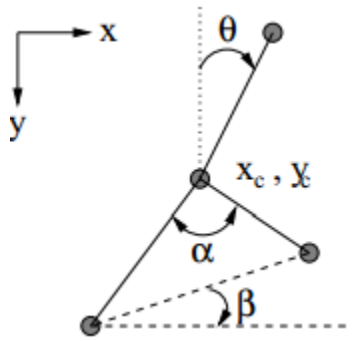
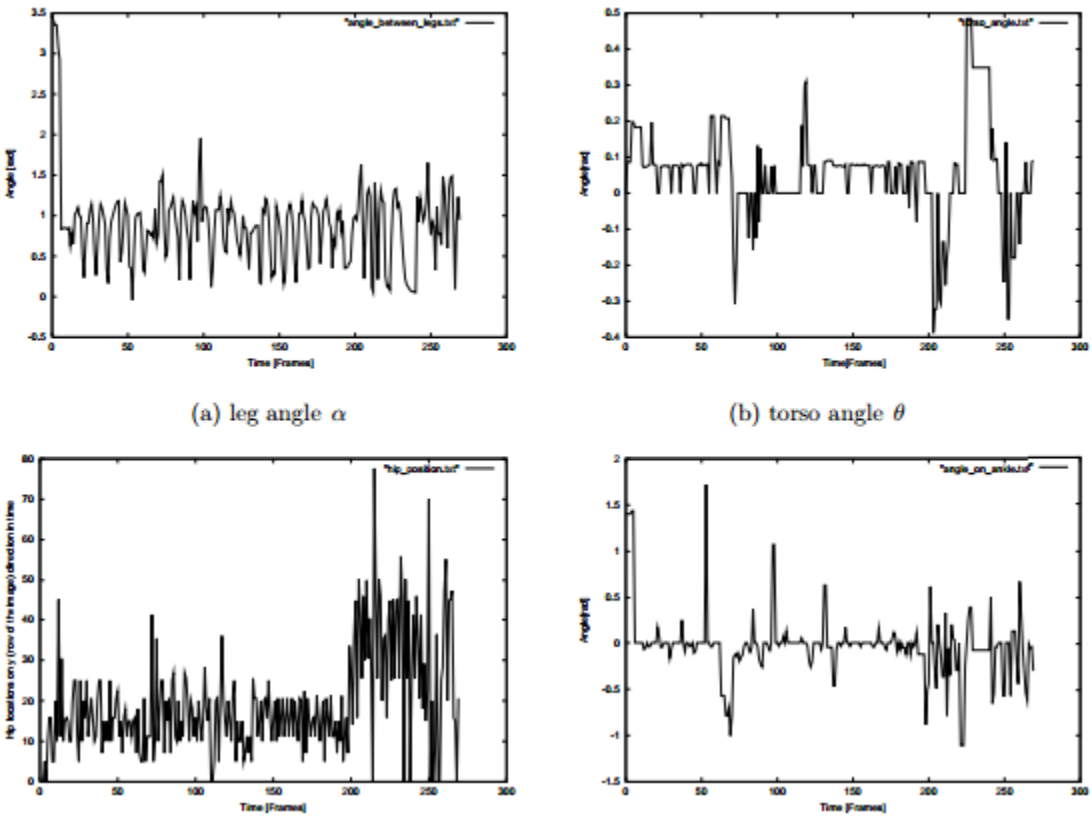


Ilustración 54: Características de la postura del esqueleto

De forma adicional, la velocidad de movimiento del marco de referencia sobre la silueta podría ser considerado en el análisis, pero únicamente se presenta una aproximación básica para que pueda ser extrapolada en estudios futuros en los que se incluya diferentes acciones como saltos, sentarse o tumbarse. Por esta razón, la idea básica del análisis del movimiento de las personas se ha centrado únicamente en la silueta, y para el escenario en el que necesitamos emplearlo, es el único paso que necesitamos conseguir, siendo correr o andar un paso adicional.

Los resultados mostrados a continuación nos alientan a implementar este tipo de parámetros en aplicaciones de vigilancia que utilizan análisis de movimiento de personas en tiempo real, aunque tenemos que tener en cuenta que las sombras en la escena

pueden perturbar los resultados, ya que podrían entrar a formar parte de la silueta y distorsionar la forma real del esqueleto. Los resultados mostrados a continuación son producidos por el análisis del movimiento de personas en una zona de vigilancia sin sombras.



(a) leg angle α (b) torso angle θ

Ilustración 55: Gráfica de resultado de detección de siluetas

El movimiento periódico de los ángulos θ, α, β y la aceleración en el centro del blob de movimiento nos dan las claves de la actividad de la persona. La actividad de caminar esta en las imágenes anteriores a 200, mientras que la actividad de correr se encuentra posterior a esa imagen en la misma persona.

III. Detección facial en zona de alta probabilidad

Tomamos como zona de alta probabilidad, el tercio superior de la silueta detectada en el apartado anterior, de esta manera centramos el procesamiento en una región específica y con un tamaño relativamente regulado aprovechando al máximo los recursos existentes. Por otro lado, al utilizar una cascada de detectores como al definida en el estado del arte, podremos descartar rápidamente los tamaños de cara definidos que no cuadren con el perteneciente a la silueta. A continuación describiremos un poco más en profundidad el algoritmo de detección facial en escala.

La cascada de detección facial complete tiene 38 fases con más de 6000 características, sin embargo, los resultados dan una media de detección rápida. En un conjunto de datos difícil, con 507 caras y 75 millones de sub- ventanas, las caras son detectadas con una media de 10 características validadas por sub-ventana. Como comparativa, este sistema es aproximadamente 15 veces más rápido que la implementación del sistema de detección construido por Rowley *(Neural network-based face detection)*, en el que se usa una implementación similar con dos redes de detección, una primera red que hace una detección más rápida pero menos precisa para preseleccionar la imagen y una segunda más lenta pero más precisa, que se encarga de encontrar las regiones de alta probabilidad.

Aunque es difícil de determinar exactamente, aparentemente el sistema de detección facial de dos redes es el más rápido actualmente. La estructura del proceso de detección en cascada es esencialmente lo que degenera en un árbol de decisión, como podemos ver en el trabajo realizado por Amit y German *(Joint induction of shape features and tree classifiers)*. A diferencia de otras técnicas que usan detectores fijos, Amit y German propusieron un punto de vista alternativo, donde ocurrencias inusuales de características simples en la imagen se utilizaban para iniciar la evaluación de un proceso más complejo de detección. De esta manera, no es necesario realizar el proceso de detección completo en todas las zonas y escalas potenciales de la imagen. Aunque suena muy sencillo el uso de esta técnica, nos obliga a utilizar una primera búsqueda de características en todas las zonas. Estas características se agrupan para encontrar ocurrencias inusuales. En la práctica, debido a la forma del detector y de las

características, su uso es extremadamente eficiente y el coste de detección inicial en todas las escalas y zonas está más que amortizado.

Recientemente Fleuret y German han presentado una técnica de detección facial que se basa en una "cadena" de tests para indicar la presencia de una cara en una localización y zona específica *(Coarse-to-fine face detection)*. Las propiedades de la imagen medidas por Fleuret y German, disyunciones de bordes de escala fina, son un tanto diferentes a las características rectangulares simples y poco interpretables. Las dos vías también difieren radicalmente en su filosofía de aprendizaje. La motivación del proceso de aprendizaje de Fleuret y German es la estimación y discriminación de la densidad, mientras que el detector propuesto es meramente discriminante. Por último, la tasa da falsos positivos de la propuesta Fleuret y Geman parece ser mayor que otras propuestas anteriores como la de Rowley y otros y que esta. Las imágenes de ejemplo tienen cada una entre 2 y 10 falsos positivos

Con el fin de entrenar las imagines frontales se ha utilizado el clasificador de 38 capas. Para entrenar el detector, se ha utilizado un conjunto de imágenes con caras y sin cara. El conjunto consiste en 4916 caras recortadas y alineadas en base de 24 x 24pixeles. Las caras se extrajeron de imágenes de multitudes aleatorias de internet. A continuación se muestran algunos ejemplos de caras típicas.

Ilustración 56: Conjunto de caras aleatorias para entrenamiento

Las sub-ventanas sin cara vienen de 9544 imágenes en las que manualmente se ha inspeccionara que no tuviesen cara. Hay en torno a 350 millones de sub-ventanas dentro de estas imágenes sin cara.

El número de características en las primeras 5 capas del detector es 1, 10, 25, 25 y 50 respectivamente. El resto de las capas tienen más características de forma incremental, llegando a un número total de 6061 en todas las capas.

Cada clasificador en cascada ha sido entrenado con las 4916 caras (además de sus imagines espejo, lo que hace un total de 9832 caras) y con 10000 sub-ventanas sin cara (también de tamaño 24 x 24) usando el procedimiento de entrenamiento Adaboost. Para el clasificador de una característica inicial, se eligieron las sub-ventanas de ejemplos sin cara de entrenamiento de forma aleatoria de un conjunto de 9544 imágenes que no contenían caras. Los ejemplos sin cara utilizados para entrenar las siguientes capas se obtuvieron con la cascada parcial a través de las imágenes sin cara y recolectando los falsos positivos, con un límite de 10000 en cada capa.

La velocidad del detector en cascada está directamente relacionada con el número de características evaluadas en cada sub-ventana escaneada.

En la evaluación del conjunto de pruebas MIT+CMU *(Neural network-based face detection)*, una media de 10 características por sub-ventana eran evaluadas de las 6061. Es posible que la mayoría de las ventanas fueran rechazadas en la primera o segunda capa de la cascada. En un procesador Pentium III a 700 MHz, el detector facial puede procesar una imagen de 384 x 288 en aproximadamente 0,067 (usando una escala inicial de 1.25 y tamaño de paso de 1.5). Esto es 15 veces más rápido que el detector propuesto por Rowley-Baluja-Kanade *(Neural network-based face detection)* y aproximadamente 600 veces más rápido que el de Schneiderman-Kanade *(A statistical method for 3D object detection applied to faces and cars)*.

El conjunto de prueba MIT+CMU frontal consiste en 130 imágenes con 507 caras frontales. A continuación podemos observar la curva ROC que muestra el rendimiento del detector. Para crear la curva ROC, el umbral de la última capa del clasificador se ha ajustado de $-\infty$ a $+\infty$. Al ajustar el umbral a $+\infty$, producirá una tasa de detección de 0,0

y una tasa de falso positivo de 0,0. Al ajustar el umbral a $-\infty$, incrementa ambas tasas aunque hasta un cierto punto. Ninguna de las tasas puede ser mayor que la tasa de detección menos la capa final. En realidad, el umbral es equivalente a eliminar la última capa. Para incrementar la tasa de detección y la de falsos positivos, es necesario reducir el umbral del clasificador en la siguiente cascada, por ello, para construir una curva ROC completa, se quitan capas del clasificador. Usamos el número de falsos positivos comparado con la tasa de falsos positivos en otros sistemas. Para calcular esta tasa, se dividen los falsos positivos entre el total de las sub-ventanas escaneadas. En este test, el número de sub-ventanas escaneadas es de 75081800.

Desafortunadamente, la mayoría de los resultados publicados anteriormente en detección facial, solo incluyen un único régimen operativo. Para poder realizar la comparación de una forma más sencilla, se ha listado la tasa de detección para la tasa de falsos positivos reportados por otros sistemas. Teniendo en cuenta los resultados de Rowley-Baluja-Kanade *(Neural network-based face detection)*, se han tomado los valores los valores de diferentes detectores que han usado produciendo diferentes resultados, los cuales aparecen bajo la misma cabecera. En el detector de Roth-Yang-Ahuja *(A snowbased face detector)*, reportaron sus resultado del test MIT+CMU menos 5 imágenes en las que mostraron las caras sin recuadro como podemos ver en la imagen a continuación.

For the Rowley-Baluja-Kanade results [12], a number of different versions of their detector were tested yielding a number of different results they are all listed in under the same heading. For the Roth-Yang-Ahuja detector [11], they reported their result on the MIT+CMU test set minus 5 images containing line drawn faces removed. Figure 7 shows the output of our face detector on some test images from the MIT+CMU test set.

False detections / Detector	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

Ilustración 57: Comparación de algoritmos con el conjunto MIT+CMU

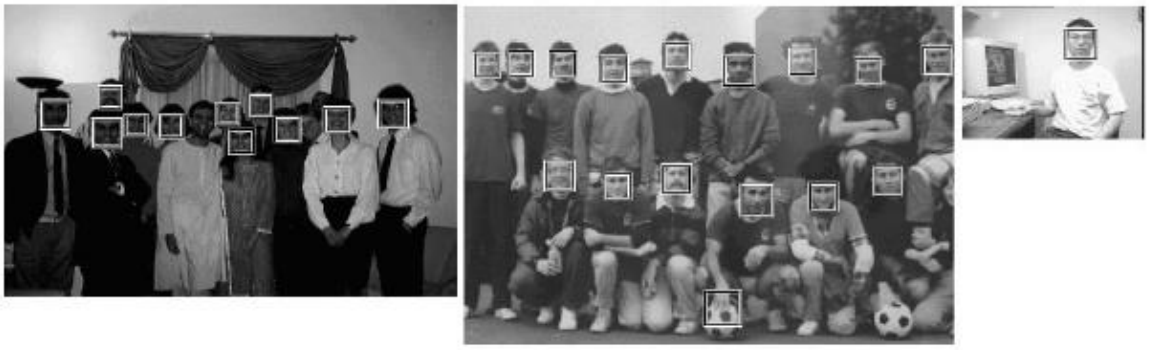


Ilustración 58: Resultado de detección sobre imagen

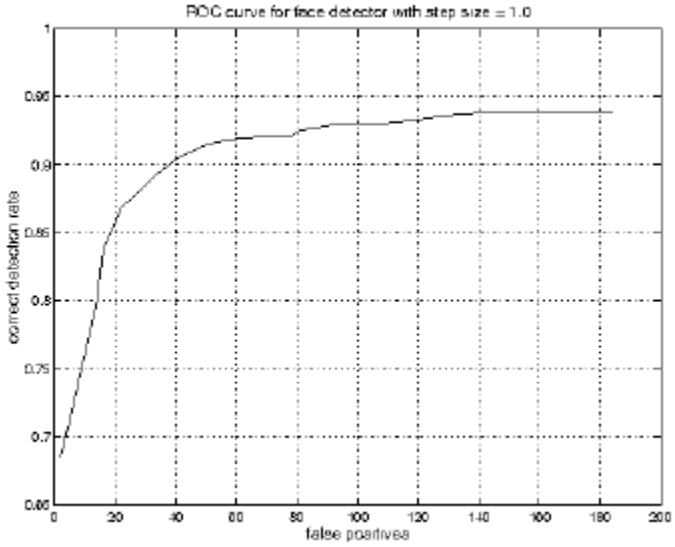


Ilustración 59: Curva ROC de la detección facial en el conjunto MIT+CMU

La tabla anterior muestra los resultados de tres detectores (el de 38 capas descrito anteriormente y dos detectores de similitudes entrenados) teniendo como salida la votación mayor de los tres detectores. Esto incrementa la tasa de detección además de eliminar más falsos positivos. Esta mejora podría ser mayor si los detectores fuesen más independientes, es decir, que tengan una correlación de los falsos positivos mayor.

Validación

1. Restricciones

Dentro de todas las técnicas que se realizan en la metodología propuesta, la que tiene mayor probabilidad de fallo es la parte de detección de rostros. A continuación enumeramos una serie de debilidades que hacen que el control de las condiciones del entorno sea una parte fundamental:

I. Angulo de incidencia

Este es el principal aspecto que hace que un algoritmo sea puntero o se encuentre en una zona medio baja, dentro del punto de vista de usabilidad. Se tiende a intentar usar las cámaras ya existentes para realizar este tipo de análisis, y por lo general su posición no es la más adecuada, ya que se suele buscar cubrir el mayor campo de visión, lo cual se consigue poniendo las cámaras en zonas altas y con un ángulo vertical considerable.

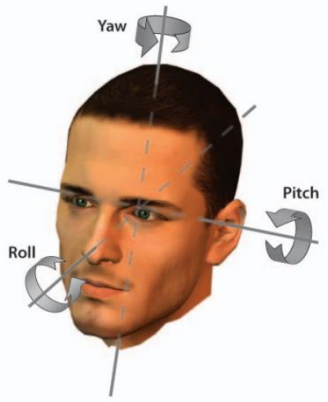


Ilustración 60: Pitch, yaw y roll

En el ángulo de incidencia se tienen en cuenta 3 vectores: pitch, yaw y roll. La rotación más conflictiva, se suele encontrar en el pitch, teniendo como norma general una posibilidad de detección de rostros cuando estos tienen un ángulo de incidencia menor a 15º con respecto al plano horizontal. Dependiendo del algoritmo utilizado, en el yaw podemos obtener detecciones positivas incluso con caras laterales, usando un límite general común entre los algoritmos de 30º, mientras que el roll permite una flexibilidad absoluta puesto que la única dificultad que produce es la ejecución de la detección más veces.

II. *Resolución*

Para poder extraer correctamente las características faciales, es necesario poder tener una cantidad de píxeles mínima que dibujen dichas características. Con una resolución excesivamente pequeña, nos sería imposible calcular algunas características como la distancia entre ojos, mientras que en el caso de una resolución demasiado grande, tendríamos problemas de rendimiento, al tener que minimizar el cálculo a la forma más básica.

III. *Iluminación*

Este aspecto es muy dependiente de la tecnología utilizada en la captación de imágenes. Por norma se estima que la iluminación ha de ser homogénea sin tener contraluces, puesto que esto hace que la zona del rostro refleje menos luz y el sensor acabe por oscureciendo dicha zona, evitando así la posibilidad de distinguir las características faciales. Existen tecnologías embebidas en las cámaras que ayudan a solucionar este tipo de problemas, como podemos ver a continuación en las cámaras Panasonic usando su tecnología *Super Dynamic*.

Ilustración 61: Tecnología Super Dynamic de Panasonic

IV. *Efectividad*

Los factores anteriormente mencionados afectan directamente a la efectividad, y no solo a la tasa de falsos negativos, sino también a la tasa de falsos positivos. El caso más claro es el de los gemelos idénticos. Este caso de falso positivo es muy complicado de solventar, ya que la constitución de su rostro puede llegar a ser idéntica e imperceptible

por el ojo humano y no medible por la mayoría de clasificadores. Otros casos de falsos positivos se pueden dar debido a sombras o brillos creados por las restricciones anteriores, los cuales pueden llegar a crear una serie de formas que hagan extraer características erróneas, y dependiendo de la restricción del umbral seleccionado, pueden llegar a convertirse en falsos positivos. Para nuestro caso, en la efectividad debemos darle la misma importancia a la tasa de falsos negativos, ya que estamos realizando únicamente la búsqueda de la existencia de rostro.

V. Privacidad

Debido a las leyes existentes, este tipo de técnicas tienen un uso muy restringido y suelen causar controversia, a pesar de la probada aportación de facilidades y seguridad que tienen. La legislación actual permite en casos contados poder grabar imágenes a empresas privadas del entorno público circundante a sus dependencias siempre y cuando no se haga ningún tipo de manipulación de las imágenes. Nuestro caso, depende de la interpretación que se le dé al sistema, podría llegar a considerarse manipulación de la información.

2. Requisitos del escenario

Teniendo en cuenta las restricciones anteriormente comentadas, se proponen las siguientes condiciones.

I. Ángulos de incidencia:

a. Pitch: Se debe ubicar la cámara lo más horizontal posible a una altura aproximada de 1,7m, y a una distancia que permita la visión de 1,2m a 2,3m.

b. Yaw: Se debe ubicar la cámara lo más frontalmente al acceso y con algún cartel que ayude a que el usuario lo observe, haciendo que este ángulo sea lo más cercano a 0º.

c. Roll: se debe poder observar al sujeto andando en forma erguida en la perpendicular al suelo.

II. *Resolución:*

a. Con el fin de asegurar una correcta detección, como densidad de pixeles estándar, se recomienda tener 300 px/m.

b. El enfoque debe estar centrado en la zona donde se estime que el sujeto va a realizar una parada antes de acceder a la sucursal.

III. *Iluminación:*

a. La zona de alta probabilidad estimada debe estar homogéneamente iluminada.

b. No debe existir iluminación de fondo

c. No se debe existir ningún tipo de iluminación orientada directamente hacia el sensor.

IV. *Escenario*

a. Debe haber una persona al tiempo en la zona de detección.

b. El fondo de la imagen debe ser estático.

Demostración práctica

A continuación se ilustran ejemplos de los diferentes casos explicados en los apartados anteriores con los pasos del algoritmo.

1. Pasos

I. Detección de movimiento y extracción de histograma

Para poder ilustrar de forma correcta la detección de movimiento, recuadramos la zona resultante con un rectángulo azul.

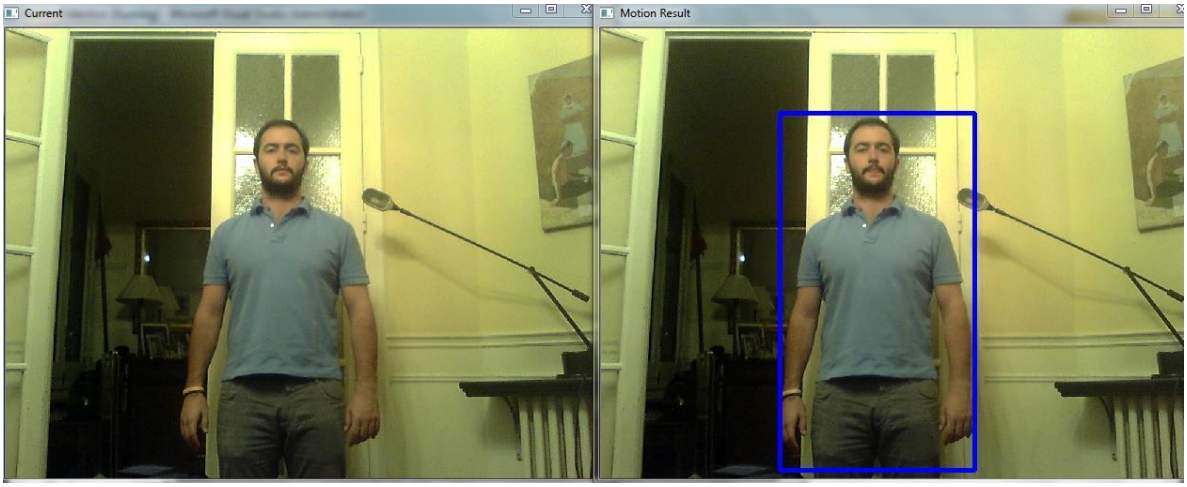


Ilustración 62: Detección de movimiento

II. Detección de personas basado en entrenamiento

En esta implementación, se ha utilizado un clasificador de detección de la zona superior del cuerpo. Una vez detectado se muestra en una ventana individual con la zona de interés recortada. Esta zona de interés es de la que sacamos la zona de alta probabilidad para realizar la detección facial.

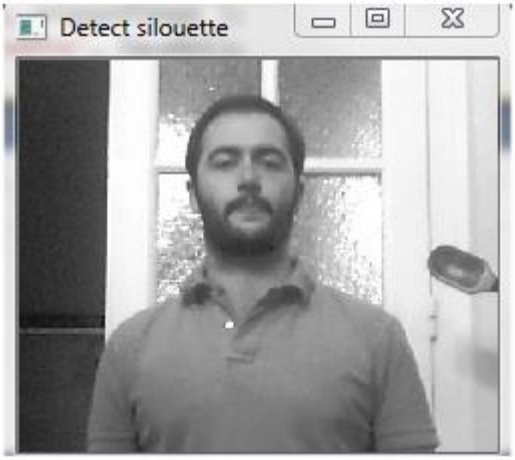


Ilustración 63: Extracción de silueta

III. Detección facial en zona de alta probabilidad

Por último, en una ventana adicional, mostramos la cara encontrada en la zona de alta probabilidad definida tras el paso anterior.

Ilustración 64: Detección facial

2. Escenarios probados

I. Cara visible

Ilustración 65: Cara visible (ejemplo I)

Ilustración 66: Cara visible (ejemplo II)

II. *Cara oculta*

Ilustración 67: Cara oculta

III. *Condiciones restrictivas*

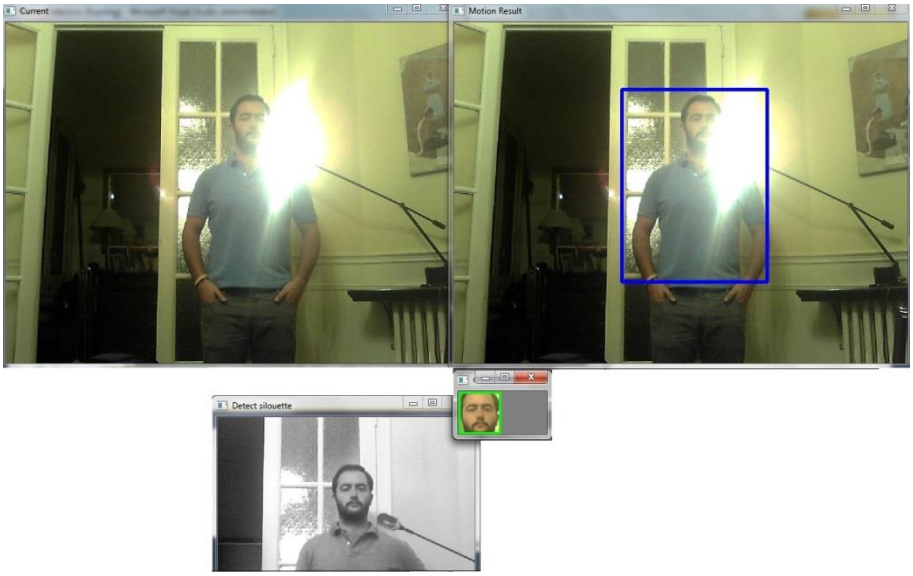


Ilustración 68: Condiciones restrictivas

Conclusión

A través de la investigación de las diferentes técnicas existentes, se ha podido demostrar de forma documental la posibilidad de implantación e integración un sistema que solvente los problemas de falta de identificación en los accesos a sucursales, siempre y cuando se cumplan los requisitos del escenario. Para el sistema se ha propuesto una solución combinada de diferentes tipos de análisis: detección de movimiento, búsqueda de siluetas y detección facial en zona de alta probabilidad.

Se ha podido ver que a nivel de rendimiento, podemos basarnos en clasificadores básicos de cascada a través de los cuales podemos hacer detecciones de patrones con un alto porcentaje de probabilidad de detección.

Como trabajo futuro, se proponen las siguientes líneas de investigación:

- Estudio de escenarios no controlados que dificultan la extracción de las características.
- Utilización de otras técnicas de detección de movimiento para poder trabajar con fondos no fijos.
- Estudio de algoritmos de extracción de características no basados en cascadas.
- Ampliación de las funcionalidades del sistema, así como de los algoritmos utilizados, para poder realizar identificación de usuarios con el fin de evitar el acceso a posibles amenazas.

Referencias

A Robust Skin Color Based Face Detection Algorithm [Online] / auth. Singh Sanjay Kr. [et al.]. - http://www.csee.wvu.edu/~richas/papers/tkjse.pdf.

Adaptive background mixture models for real-time tracking [Online] / auth. Grimson Chris Stauffer W.E.L. - http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf.

An Adaptive Clustering Algorithm for Image Segmentation [Online] / auth. Pappas Thrasyvoulos N.. - http://www.cs.gsu.edu/~wkim/index_files/papers/adaptiveclustering.pdf.

An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for purpose on visual tracking [Online] / auth. Cox Ingemar J. and Hingorani Sunita L.. - http://www.csd.uoc.gr/~hy576/bibliography/i0138.pdf.

An EM-like algorithm for color-histogram-based object tracking [Online] / auth. Zivkovic Zoran and Krose Ben. - http://staff.science.uva.nl/~zivkovic/Publications/zivkovic2004CVPR.pdf?origin=publication_detail.

An image preprocessing algorithm for illumination invariant face recognition [Online] / auth. Gross Ralph and Brajovic Vladimir. - https://kuliahku.googlecode.com/files/Paper%20A4.pdf.

Analíticas de vídeo IP [Online] / auth. Addati Mg. Ing. Gastón A.. - http://www.ucema.edu.ar/publicaciones/download/documentos/529.pdf.

Analog Vs. Megapixel IP Cameras [Online] / auth. Jansen Electronics. - http://www.jansenelectronics.net/shoppingcart/pages/Analog-Vs.-Megapixel-IP-Cameras.html.

Automatic Spatio-Temporal Video Sequence Segmentation [Online] / auth. Vass Jozsef, Palaniappan Kanappan and Zhuang Xinhua. - http://meru.cs.missouri.edu/people/vass/seg_icip_pap.pdf.

Computer Vision Face Tracking For Use in a Perceptual User [Online] / auth. Bradski Gary R.. - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.7673&rep=rep1&type=pdf.

Digital video vs Analog [Online] / auth. DSC. - http://www.discount-security-cameras.net/analog-vs-ip-technology.aspx.

Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection [Online] / auth. Belhumeur Peter N., Hespanha Joao P. and Kriegman and David J.. - http://ftp.idiap.ch/pub/courses/EE-700/material/17-10-2012/fisherface-pami97.pdf.

Face Description with Local Binary Patterns: Application to Face Recognition [Online] / auth. Ahonen Timo, Hadid Abdenour and Pietikäinen Matti. - http://robotics.csie.ncku.edu.tw/HCI_Project_2010/P76994220_%E6%B1%9F%E5%A7%BF%E6%85%A7/Face%20Description%20with%20Local%20Binary%20Patterns_TPami_2006_2.pdf.

Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis [Online] / auth. Tziritas Christophe Garcia and Georgios. - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.9044&rep=rep1&type=pdf.

Face detection: A survey [Online]. - http://sdpy.googlecode.com/svn/tags/temp/unsorted3/to_delete/tmp/unsorted/to_remove/research/my_papers/phdthesis/review/survey/FaceDetection(survey).pdf.

Fast Multiple Object Tracking via a Hierarchical Particle Filter [Online] / auth. Yang Changjiang, Duraiswami Ramani and Davis Larry. - http://www.umiacs.umd.edu/~ramani/pubs/Yang+ICCV2005.pdf.

Fast Object Tracking Using Adaptive Block Matching [Online] / auth. Hariharakrishnan Karthik and Schonfeld Dan. - http://140.133.9.112:8080/cgit/PaperDL/CMS_071116105936.pdf.

Fast Occluded Object Tracking by a Robust Appearance Filter [Online] / auth. Nguyen Hieu T. and Smeulders Arnold W.M.. - http://dare.uva.nl/document/34284.

FloatBoost Learning and Statistical Face Detection< [Online] / auth. Li Stan Z. and Zhang ZhenQiu. - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.6563&rep=rep1&type=pdf.

Human Skin Colour Clustering for Face Detection [Online] / auth. Kovac Jure, Peer Peter and Solina Franc. - http://eprints.fri.uni-lj.si/2113/1/Human_Skin_Colour_Clustering_for_Face_Detection.pdf.

Integrating Color and Shape-Texture Features for Adaptive Real-time Object Tracking [Online] / auth. WANG Junqiu and YAGI Yasushi. - http://www.am.sanken.osaka-u.ac.jp/~wang/pdf/TIPSmallII2008.pdf.

Intro Mean Shift [Online]. - http://www.cse.psu.edu/~rcollins/CSE598G/introMeanShift_6pp.pdf.

Kernel-Based Object Tracking [Online] / auth. Comaniciu Dorin, Ramesh Visvanathan and Meer Peter. - https://www.cs.drexel.edu/~kon/advcompvis/papers/Comaniciu_TPAMI03.pdf.

La evolución de los sistemas de vigilancia por vídeo [Online] / auth. Axis Communications. - http://www.axis.com/es/products/video/about_networkvideo/evolution.htm.

Meanshift [Online]. - http://xphilipp.developpez.com/articles/meanshift/.

Meanshift and Camshift [Online] / auth. Open CV. - http://docs.opencv.org/trunk/doc/py_tutorials/py_video/py_meanshift/py_meanshift.html.

Mean-shift Blob Tracking through Scale Space [Online] / auth. Collins Robert T.. - http://www.cse.psu.edu/~rcollins/Papers/cvpr2003.pdf.

Mean-shift Blob Tracking through Scale Space [Online] / auth. Collins Robert T.. - http://www.cse.psu.edu/~rcollins/Papers/cvpr2003.pdf.

Neural Network-Based Face Detection [Online] / auth. Rowley Henry A.. - http://atgpsts.xeds.eu/other/H_A_Rowley_Neural_Network-Based_Face_Detection.pdf.

Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces [Online] / auth. Allen John G., Xu Richard Y. D. and Jin Jesse S.. - http://crpit.com/confpapers/CRPITV36Allen.pdf?origin=publication_detail.

Object Tracking Using the Gabor Wavelet Transform and the Golden Section Algorithm [Online] / auth. He Chao, Zhen Yuan F. and Ahalt Stanley C.. - http://www2.ece.ohio-state.edu/~zheng/publications/Gabor-object.pdf.

Object tracking: a survey [Online]. - http://cvpr.uni-muenster.de/teaching/ss10/BildverarbeitungundComputerVisionSS10/script/BVCV-12-Tracking.pdf.

People Tracking via a Modified CAMSHIFT Algorithm [Online] / auth. Guraya Fahad Fazal Elahi, Bayle Pierre-Yves and Cheikh Faouzi Alaya.

Performance Evaluation of Object Tracking Algorithms [Online] / auth. Yin Fei, Makris Dimitrios and Velastin Sergio. -

http://www.researchgate.net/publication/228873288_Performance_evaluation_of_object_tracking_algorithms/file/79e41508ee5814ccfd.pdf.

Pfinder: Real-Time Tracking of the Human Body [Online] / auth. Wren Christopher [et al.]. - http://pdf.aminer.org/001/033/397/pfinder_real_time_tracking_of_the_human_body.pdf.

Rapid Object Detection using a Boosted Cascade of Simple Features [Online] / auth. Viola P. and Jones M.. - https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf.

Real-Time Tracking of Non-Rigid Ob jects using Mean Shift [Online] / auth. Comaniciu Dorin, Ramesh Visvanathan and Meer Peter. - http://www.cs.ucf.edu/courses/cap6412/2001/mean_shift.pdf.

Robust Object Tracking with Online Multiple Instance Learning [Online] / auth. Babenko Boris, Yang Ming-Hsuan and Belongie Serge. - http://faculty.ucmerced.edu/mhyang/papers/pami11b.pdf.

Robust Real-Time Face Detection [Online] / auth. Viola P. and Jones M.. - http://www.vision.rwth-aachen.de/teaching/cvws08/additional/viola-facedetection-ijcv04.pdf.

Rotation invariant neural network-based face detection [Online]. - http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA341629.

Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance [Online] / auth. EKINCI Murat and GEDIKLI Eyüp. - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.1771&rep=rep1&type=pdf.

Training an attentional cascade [Online] / auth. Brandt Jonathan. - http://www.google.com/patents/US7440930.

Training Support Vector Machines: an Application to Face Detection [Online] / auth. Isuna Edgar, Freund Robert and Grosi Federico. - http://mmm.csd.uwo.ca/faculty/olga/Courses/Fall2008/9840/Chosen/SVMFaceOsunaCVPR97.pdf.

Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition [Online] / auth. Yang Jian [et al.]. - http://repository.lib.polyu.edu.hk/jspui/bitstream/10397/190/1/137.pdf.