UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ingeniería
Departamento de Ing. Civil Informática y
Ciencias de la Computación

Profesor Patrocinante:
John Atkinson Abutridy

Comisión:
Ma. Angélica Pinninghoff Junemann
Javier Vidal Valenzuela

DESARROLLO DE UN PROTOTIPO DE SISTEMA DE RE-ID DE PERSONAS PARA VIDEO-VIGILANCIA

DIEGO A. REYES MOLINA

Informe de Memoria de Título
Para optar al Título de

Ingeniero Civil Informático

Noviembre 2015

Índice general

Índice de cuadros

Capítulo 1

Introducción

Por muchos años, se han utilizado cámaras de video en el área de la seguridad para vigilar zonas de interés, transmitiendo escenas en tiempo real desde distintos lugares a puestos de monitoreo centralizados, que analizan simultáneamente varios flujos de imágenes con el objetivo de detectar situaciones no deseadas. En sus inicios este análisis era llevado a cabo sólo por personas, las que debían dedicar tiempo exclusivo para monitorizar imágenes, lo que lo convertía en una labor costosa y poco eficiente.

En las últimas décadas, los avances en el área de procesamiento de imágenes y visión artificial permitieron desarrollar sistemas de vigilancia inteligente capaces de comprender una escena y detectar condiciones de riesgo sin la ayuda de humanos. En términos de seguridad, la mayoría de estos sistemas se dedican al análisis de personas, debiendo enfrentar problemas como la detección automática de personas en la escena. Luego de la detección, se requiere efectuar un seguimiento cuadro a cuadro de la ubicación de la persona a través de una o varias cámaras. Dependiendo de la ubicación de cada cámara, puede haber campos de visión (Field of View, FOV) superpuestos con otros de cámaras vecinas. En estos casos, se puede utilizar la relación existente entre la información capturada por cada cámara [21,34], pudiendo incluso inferir automáticamente la topología de la red de cámaras [18,37]. De esta forma, la existencia de áreas sin cobertura visual puede ser compensada estimando la trayectoria de una persona, basándose en su velocidad y dirección antes de desaparecer [8]. Sin embargo, en redes de cámaras con regiones ciegas demasiado extensas, es imposible predecir cuándo y dónde reaparecerá el individuo. De ahí que se requiere re-identificar personas sin utilizar características derivadas de su posición (dirección, velocidad, aceleración, etc.).

El principal interés en re-identificar una persona es lograr establecer la trayectoria recorrida y los lugares visitados dentro de toda una red de cámaras de seguridad. En video-vigilancia, la re-identificación (Re-ID) de personas consiste en reconocer si un individuo ha sido observado previamente. Formalmente, esto se puede definir como la tarea de asignar el mismo identificador (o identificadores lo suficientemente parecidos) a todas las instancias de una persona, por medio de aspectos visuales capturados desde imágenes o vídeos [41]. En general, Re-ID busca responder a las preguntas: ¿Dónde se ha visto a esta persona antes? y ¿A dónde fue, luego de ser vista en cierto lugar?. Respondiendo lo anterior, se puede lograr un seguimiento de personas sobre grandes extensiones de terreno resguardado por una red de cámaras [8].

Un subproblema de la Re-ID de personas es la re-identificación dentro de un solo FOV, es decir, establecer cuando un individuo reingresa al lugar monitorizado por una cámara. Una aplicación real de esto [7], es determinar la presencia reiterada de una persona en un lugar sospechoso (paradero

de autobús), con el propósito de detectar traficantes de droga.

En este trabajo se aborda el problema de re-identificar personas dentro de un mismo FOV, con el objetivo de detectar automáticamente robos de vehículos, basándose en la apariencia física global de una persona. Para esto, se desarrolló un modelo de sistema de reidentificación entre personas que abandonan la escena luego de descender de un vehículo, y las que ingresan a la escena para abordar el mismo vehículo. Una alerta de posible robo se provocará cuando el sistema determine que las imágenes de estas personas corresponden a distintos individuos. Además, se desarrolló un prototipo del modelo anterior, y se evaluó su efectividad en un escenario real (estacionamiento comercial).

A diferencia de un sistema de Re-ID clásico que busca coincidencias para una persona entre muchos candidatos [19,20,22,24,36], el modelo propuesto compara cada persona con solo un candidato (la persona que reingresa al vehículo). cambiar

1.1. Objetivo general

Desarrollar un prototipo de sistema Re-ID de personas para vigilancia semi-automática en estacionamientos de automóviles.

1.1.1. Objetivos específicos

- Estudiar técnicas de: detección de personas en videos, segmentación entre primer y segundo plano en videos, extracción de características relevantes en imágenes (cuadros de video), y comparación de descriptores de imágenes.
- Implementar prototipo de sistema de Re-ID de personas.
- Evaluar precisión del prototipo.

1.2. Organización de la Memoria

por definir...

Capítulo 2

Sistemas de Re-ID

Re-ID se define como la tarea de establecer correspondencia entre imágenes de personas tomadas desde cámaras diferentes. Un sistema típico de Re-ID ejecuta este proceso en dos fases [4]: (1) generación de una identificación (descriptor) para cada persona detectada, y (2) comparación entre descriptores, tal como se muestra en la figura 2.1.

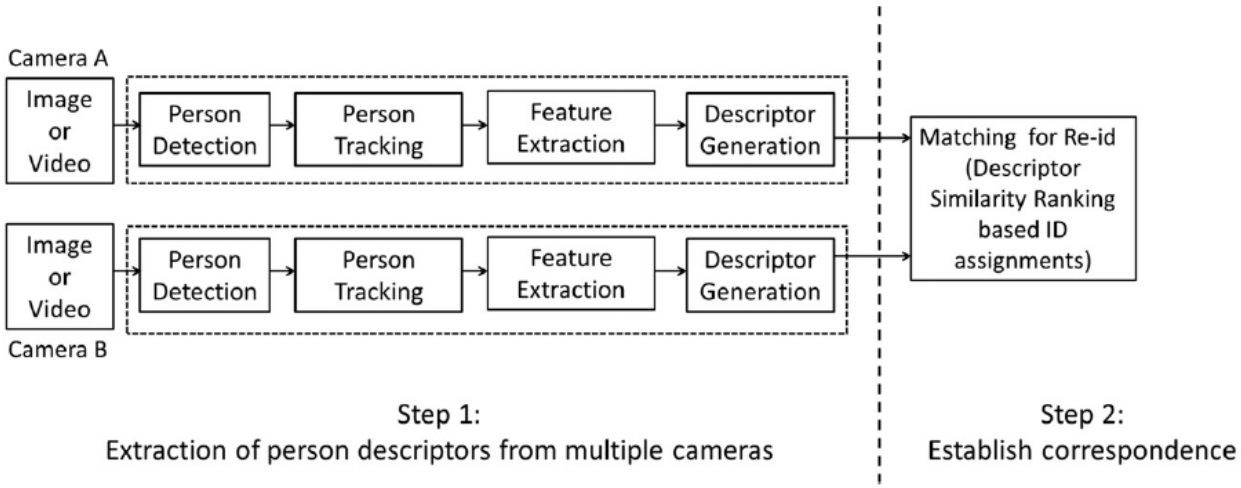

Figura 2.1: Sistema de Re-ID [4]

En una primera fase, se genera el descriptor de una persona detectada en múltiples escenas (capturadas con una o varias cámaras). En una segunda fase, se establece la correspondencia entre descriptores, determinando si estos coinciden con la misma persona o son individuos distintos.

La efectividad de un sistema de Re-ID se evalúa mediante una curva CMC (*Cumulative Matching Characteristic*), que mide la probabilidad de que el sistema entregue correctamente el descriptor que corresponde al descriptor consultado, donde la respuesta está compuesta por k candidatos, y es considerada correcta si el descriptor que hace match con la consulta está entre dichos candidatos [15]. Luego, CMC indica la proporción de respuestas correctas obtenidas para distintos valores de k:

$$\text{Precisión} = \frac{\#\text{ respuestas correctas}}{\#\text{ consultas realizadas}} \times 100$$

aclarar que el denominador corresponde sólo a las pre-

Las tareas típicas de la primera fase incluyen [45]:

- Detección de personas: se establece la presencia de personas en una escena. Para esto, se pue-

Figura 2.2: Ejemplos Data Set VIPeR: cada columna muestra una de las 632 personas.

den utilizar algoritmos de aprendizaje supervisado cuando se cuenta con una base de datos con imágenes etiquetadas [13]. En el caso de un video, existen varias imágenes consecutivas (cuadros o *frames*), lo que permite detectar cuando la persona se mueve o desplaza por la escena, según los cambios producidos en cada píxel. Esto se conoce como segmentación entre primer y segundo plano (que corresponden al movimiento y regiones estáticas, respectivamente).

La detección del segundo plano permite eliminar de la imagen todo lo que no forma parte de la persona, ocultando todo el conjunto de píxeles que permanezcan sin cambios. Para esto, se utiliza un mapa de bits, que se superpone a la imagen original, convirtiendo el segundo plano en un mismo valor (bit cero: color negro), dejando los píxeles del primer plano inalterados (bit uno: color original) [9]. Luego, un algoritmo de detección de bordes [12] encuentra fronteras entre regiones similares, obteniendo la silueta del objeto o persona en movimiento. En algunas ocasiones, cuando existe alto grado de contraste entre una persona y el fondo (como en una toma a la misma altura del individuo) la detección de bordes puede prescindir de la segmentación. Sin embargo, ésta permite obtener bordes con menos posibilidad de error, dado que todo el fondo tiene un mismo valor [9].

Uno de los problemas de utilizar segmentación entre primer y segundo plano para detectar personas, es el ruido producido por cambios de iluminación repentinos y otros elementos en movimiento (vehículos, animales, sombras, etc.), lo que hace almacenar regiones de la imagen que no contiene personas, siendo analizadas erróneamente más adelante. Este problema se puede enfrentar utilizando un clasificador binarios sobre imágenes, seleccionando aquellas que sean clasificadas como personas [33]. Una solución propuesta es filtrar las siluetas detectadas con base en su posición y forma (ver capítulo 3).

- Seguimiento de la persona: en esta tarea, se sigue la trayectoria realizada por una persona dentro de un FOV. De esta forma, la Re-ID se utiliza para lograr seguir de personas a través de varias cámaras. Por otro lado, durante el seguimiento individual de cada cámara, se pueden inferir datos de la persona, como: su dirección de movimiento, ubicación inicial y final. Estos datos pueden ser usados para filtrar los candidatos a re-identificar.

- Extracción de características: en esta tarea, se extraen las características que formarán el descriptor de la persona. Las características pueden ser de distintos tipos, entre ellos [41]:

Escena	personas	ocurrencias	exactitud
Salón	12	80	89.3 %
Corredor	58	38	70.4 %

Cuadro 2.1: Resultados obtenidos en [26]

- Color: una imagen está compuesta por píxeles cuyos valores dependen del espacio de colores empleado. Un espacio de colores está compuesto por varios canales, por ejemplo, el espacio RGB está conformado por los canales rojo (R), verde (G) y azul (B); o el espacio HSV, compuesto por los canales de tonalidad (H), saturación (S) y valor (V). El color como característica de apariencia se ha empleado en forma de histogramas [14, 19, 20, 22, 27, 36, 46]. Se pueden utilizar diferentes canales de colores y sus combinaciones. Por ejemplo, del espacio de colores HSV, se ha empleado sólo el tono [14], tono y saturación [20], o los tres canales del espacio [19]. Por otro lado, histogramas del espacio de colores RGB fueron utilizados en [6, 27, 35]. Otros [22, 36, 46] han adoptado una concatenación de histogramas de los canales de los espacios RGB, YCbCr y HSV (sólo tono y saturación). Un dataset de imágenes (VIPeR) [22] ha sido ampliamente utilizado como *benchmark*, debido a la cantidad de personas (632), las que fueron fotografiadas (48x128 píxeles) en ambientes exteriores, siempre desde dos ángulos distintos y en diferentes posiciones (ver figura 2.2). Las pruebas determinan que los canales más discriminantes (en orden descendente) para Re-ID personas, son: tono, saturación, azul, rojo y verde [32]. Por último, pruebas de Re-ID con el dataset VIPeR utilizando descriptores compuestos únicamente por histogramas de colores y comparándolos con distancia euclidiana, para rangos $k = 1$ y $k = 20$ en la curva CMC, se obtiene un 6 % y 38 % de precisión de reconocimiento, respectivamente [25].
- Forma: se refiere a datos de la silueta detectada (altura, ancho, ejes de simetría, relación entre ancho y altura, etc.). Se han propuesto algoritmos que hacen uso de la simetría de la figura humana [19]: *Symmetry-Driven Accumulation of Local Features*, SDALF. El método segmenta la silueta de la persona en tres partes: cabeza, torso y piernas (ver figura 2.3), comparando cada parte con su homóloga correspondiente. Luego, se buscan ejes verticales de simetría para cada una de las partes mencionadas, con el objetivo de ponderar las features (histogramas del espacio HSV de subregiones de píxeles) en relación a la distancia de éstas con el eje, destacando aquellas features que estén cercanas al eje, dado que tienen menor probabilidad de pertenecer al segundo plano de la imagen. Una evaluación de SDALF con el dataset VIPeR, obtiene para $k = 1$ y $k = 20$ en la curva CMC, un 20 % y 65 % de precisión de reconocimiento, respectivamente [19].
- Posición: cuando los FOV entre cámaras están superpuestos, la posición de una persona puede ser utilizada para Re-ID, dado que el movimiento capturado en cada cámara es equivalente [11, 28].
- Textura: entrega la disposición espacial de los colores de una imagen [38]. Se ha representado por puntos seleccionados de la imagen (puntos de interés o *keypoints*), cuyas propiedades se mantienen invariables a cambios de tamaño [3, 31], permitiendo Re-ID objetos capturados a distintas distancias. Generalmente, estos puntos se encuentran en

keypoint necesarios	Precisión
40	99 %
30	95 %
20	85 %
10	10 %

Cuadro 2.2: Resultados obtenidos en [23]

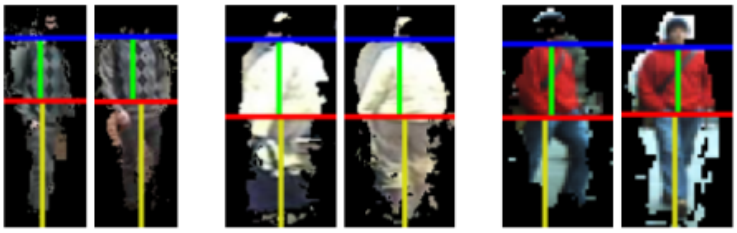

Figura 2.3: Algoritmo SDALF: Segmentación de la imagen usando ejes horizontales (partes asimétricas) y verticales (partes simétricas).

la frontera entre regiones con colores muy distintos. Algunos enfoques buscan Re-ID personas que abandonan y luego reingresan al FOV de una misma cámara, buscando enfrentar de forma robusta, cambios de iluminación, postura y escala [26]. Para formar el descriptor de cada persona, se emplean histogramas de colores y keypoints seleccionados por un algoritmo típico de transformación de características [31] (Scale-invariant feature transform, SIFT). Los keypoints detectados por SIFT corresponden a una región circular de la imagen con una dirección, siendo representada por las coordenadas de su centro (x, y), el tamaño (radio) y dirección (ángulo). El proceso de comparación se realizó con un clasificador binario. Las pruebas se realizaron con videos[1] en dos escenarios distintos, donde aparece un grupo limitado de personas que abandonan e ingresan en reiteradas ocasiones, obteniendo una exactitud desde un 70.4 % hasta 89.3 % (ver cuadro 2.1). Un trabajo comparable a [26] es presentado en [23], donde se emplea el mismo dataset y las características son keypoints detectados con SIFT. La precisión de Re-ID obtenida en este trabajo depende de la cantidad mínima de keypoints coincidentes requeridos para establecer a sus respectivos descriptores como un match (ver cuadro 2.2).

- Generación del descriptores: en esta tarea, las características detectadas son organizadas en estructuras de datos (ej. vector de n dimensiones), las que reciben el nombre de descriptores. Generalmente se calculan estadísticas sobre las características, para generar una representación sucinta de éstas (ej. histogramas).

La segunda fase de un sistema de Re-ID define la forma de asociar descriptores. Para esto, se determina si un par de descriptores corresponden o no a una misma persona.

Para encontrar una coincidencia (*match*) se han propuesto varios algoritmos. Uno típico calcula la distancia Euclidiana entre el descriptor consultado (*query*) con todos los posibles candidatos (vecinos) y luego elije a los k candidatos que se encuentren a menor distancia, técnica conocida como el

[1] Proyecto CAVIAR IST 2001 37540 financiado por la Comunidad Europea, disponible en `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

k-ésimo vecino más cercano (k Nearest Neighbor, k-NN). Una mejora utiliza una métrica de distancia que considera patrones estadísticos obtenidos de ejemplos en forma de tuplas $(A, B, etiqueta)$, donde A y B son descriptores, y la etiqueta indica si éstos son un match o no [16, 24, 25, 44, 46, 47].

Para comparar dos descriptores compuestos por keypoints (ej. SIFT), se calcula la distancia de cada keypoint de un descriptor con todos los keypoints del otro descriptor, estableciendo una correspondencia positiva si existe un par de keypoints a una distancia lo suficientemente pequeña, menor a cierto umbral. Si se cuenta con varias imágenes por persona, se puede utilizar un sistema de votación, el cual asigna un voto al descriptor candidato siempre que éste posea el keypoint más similar (a menor distancia) a un keypoint del descriptor consultado, emitiendo tantos votos como keypoints tenga éste, seleccionando luego al descriptor candidato con más votos [23].

En general, la comparación de descriptores se puede realizar: (a) midiendo su similitud usando métricas de distancia directa, (b) utilizando algoritmos de aprendizaje supervisado o (c) aplicando métodos de optimización [32].

Las métricas de distancia directas estiman diferencias entre descriptores para determinar la correspondencia. La métrica más usada es la distancia Euclidiana, utilizada sobre descriptores basados en color [1, 19] y puntos de interés [20]. Sin embargo, utilizar únicamente métricas de distancia directa no siempre permite establecer una correcta similitud entre descriptores, pues cuando la mayoría de las features de un descriptor coincide con los de otro correspondiente a una persona desigual, los descriptores respectivos estarán cercanos entre sí. Del mismo modo, para una misma persona que cambia de apariencia (cambio drástico en varios features a causa de diferente iluminación, postura de la persona, punto de vista, etc.), sus respectivos descriptores se encontrarán a mayor distancia. Como consecuencia, al emplear sólo una distancia tradicional, se ignora cualquier regularidad estadística, la que podría ser estimada con algoritmos de aprendizaje supervisados [36, 46]. Un sistema de Re-ID de personas que sólo emplea una métrica directa, no es robusto ante cambios de iluminación, razón por la que se requiere una calibración previa en la red de cámaras [21, 27, 34, 37].

Por otro lado, varias técnicas de aprendizaje automático (Machine Learning, ML) se han empleado en sistemas de Re-ID para distintos objetivos. Por ejemplo, un clasificador binario puede distinguir entre descriptores positivos (match) y negativos (mismatch). Un sistema de Re-ID basado en un clasificador binario, puede etiquetar un par de descriptores como match incluso si éstos presentan más diferencias que otro par compuesto por descriptores de personas distintas [36]. El método anterior obtiene, para $k = 1$ y $k = 20$ en la curva CMC, un 14 % y 68 % de precisión en reconocimiento, respectivamente. Un algoritmo clásico (AdaBoost) [22], selecciona las características que determinan una mayor diferencia entre pares de imágenes, con base en pares etiquetados de imágenes, obteniendo un 12 % y 60 % en la curva CMC para rangos de $k = 1$ y $k = 20$, respectivamente. También, se ha empleado ML para aprender funciones de similitud o distancia (Distance Metric Learning o DML). El objetivo es encontrar un espacio geométrico donde descriptores de una misma persona queden a poca distancia, al mismo tiempo que descriptores de personas distintas estén a una distancia mayor [5]. Basándose en este método, se puede establecer distancias de forma relativa a un tercer descriptor, por ejemplo, indicando que A está más cerca de B que de C [47]. Esto obtiene, para $k = 1$ y $k = 20$ en la curva CMC, un 15.7 % y 70.1 % de precisión de reconocimiento, respectivamente. Otros métodos [43] utilizan una función de distancia obtenida con DML, en un clasificador kNN. Los resultados muestran un 18 % y 75 % de precisión de reconocimiento

Referencia	Características			Asociación		
	Color	Textura	Forma	Distancia	Aprendizaje	Optimización
[1, 14, 19, 20]	✓	✓		✓		
[2]		✓			✓	
[6, 42]	✓	✓	✓	✓		
[22, 36, 46]	✓	✓			✓	
[23]		✓		✓		
[24]	✓				✓	
[27, 34]	✓					✓
[29]	✓	✓	✓			✓
[35]	✓			✓		
[40]		✓			✓	

Cuadro 2.3: Clasificación de trabajos en tipos de métodos de Re-ID [32]

para $k = 1$ y $k = 20$ en la curva CMC, respectivamente [25]. Mejoras al enfoque anterior permiten determinar cuándo un par de descriptores no forman un match (detectar una pareja negativa) [16]. Para esto, es necesario establecer una distancia umbral, tal que, el elemento consultado es aceptado sólo si existe un vecino cuya distancia es inferior a dicho umbral. En otro caso, el descriptor consultado es rechazado o detectado como una persona nueva. Los resultados obtenidos para $k = 1$ y $k = 20$, alcanzan aproximadamente un 20 % y 80 % de precisión de reconocimiento, respectivamente[2]. A diferencia de las métricas directas, los métodos que utilizan DML son menos sensibles a las features seleccionadas. Sin embargo, en escenarios reales no siempre se cuenta con un conjunto de datos previamente etiquetados para entrenar el sistema.

Por último, los métodos de optimización, buscan la métrica utilizando una función objetivo que minimiza la distancia entre pares de descriptores que coinciden. Para mantener una distancia mínima entre pares de descriptores no coincidentes, cada caso de mismatch es tratado como una restricción del problema de optimización [27, 29, 34]. Una desventaja de este enfoque es el costo computacional, debido a la cantidad de restricciones y variables del problema.

[2]Resultados obtenidos por la mejor ejecución, a diferencia de otros trabajos donde los autores muestran un promedio de los experimentos realizados

Capítulo 3

Re-ID utilizando puntos de interés

Este capítulo describe un prototipo de sistema de Re-ID de personas, enfocado en el procesamiento de videos de vigilancia reales en ambientes no controlados. La arquitectura en la que se basa el prototipo se puede ver en la figura 2.1

3.1. Detección de Personas

El sistema de Re-ID se desarrolló para trabajar con videos capturados desde cámaras inmóviles, lo que permite distinguir entre las partes de la imagen que corresponden al escenario de fondo (puntos fijos) y primer plano (puntos móviles). De ahí que el sistema se enfoca únicamente en las regiones pertenecientes al primer plano o regiones de interés, por lo que sólo se requiere seleccionar aquellas regiones de interés pertenecientes a personas.

3.1.1. Segmentación de planos

Se utilizó el algoritmo de segmentación entre primer y segundo plano presentado en [48, 49] que identifica los píxeles correspondientes a zonas en movimiento según los cambios producidos entre cada cuadro consecutivo del video, utilizando la distribución de probabilidad de Gauss. Esta detección genera un nuevo mapa de bits de dimensión idéntica al cuadro original y cuyos píxeles pueden ser blanco o negro para zonas en movimiento y estáticas, respectivamente (ver figura 3.1(b)). Sin embargo, esta primera detección es sensible a pequeños movimientos producto de vibraciones, corrientes de aire, cambios de iluminación, etc, lo que altera píxeles aislados que forman regiones conexas que no alcanzan a ser de tamaño considerable. Por esto, se utilizó un método basado en morfología mátematica[1] para eliminar el ruido de píxeles blancos presente en una imagen (figura 3.1(d)).

Las sombras pueden formar parte del primer plano si estás no son detectadas, lo que implica que la región de interés abarque espacio que no corresponde al cuerpo de la persona. Sin embargo, las sombras pueden ser detectadas por el mismo algoritmo de segmentación citado anteriormente. Este algoritmo puede generar un mapa de bits, esta vez con píxeles en escala de grises (ver figura 3.1(c)), donde las sombras son representadas con valores grises. Dado que el segundo plano es representado únicamente por píxeles de color negro, eliminar la sombra del primer plano equivale a transformar el gris a negro. Esta transformación consiste en convertir todo píxel que tenga un

[1] específicamente aplicando el operador de erosión, seguido por la dilatación de la imagen.

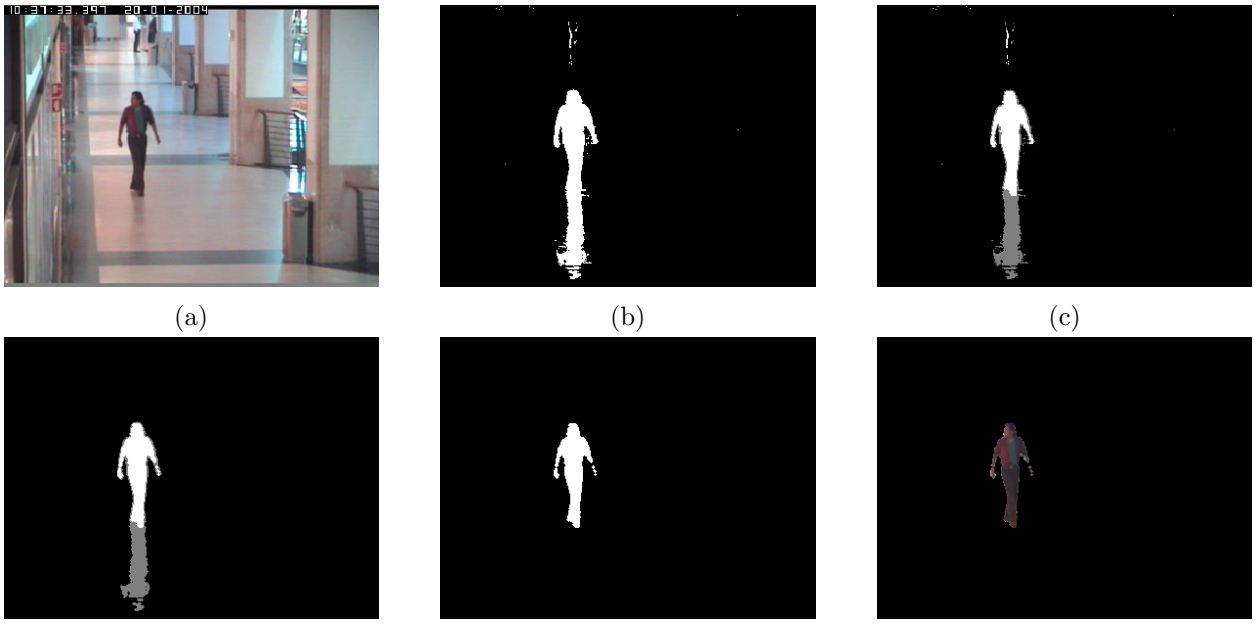

Figura 3.1: Segmentación:(a) Frame original. (b) Sin detección de sombra. (c) Con detección de sombras. (d) Ruido filtrado. (e) Sombra filtrada. (f) Frame enmascarado

valor menor a un umbral en cero (equivalente al negro), y todo valor mayor o igual a dicho umbral en uno (equivalente al blanco). El resultado de la aplicación de dicho umbral se puede apreciar en la figura 3.1(e). Luego, se utilizó este mapa de bits como una *máscara* que se sobrepone al frame, ocultando todo el segundo plano (*background subtraction*). Los píxeles de cada frame (máscara y original) son operados como valores lógicos con el operador AND, considerando sus valores como secuencia de bits (ver figura 3.1(f)).

Una vez eliminado el segundo plano, se identifican todas las formas persistentes en la imagen (regiones de interés). Para esto se considera la existencia de máximo una forma perteneciente a persona dentro de un frame.

3.1.2. Selección de la región de interés

El mapa de bits sin ruido y con sombras filtradas (figura 3.1(e)), se utiliza para calcular los contornos de las formas pertenecientes al primer plano. El cálculo se realiza con un algoritmo de seguimiento de bordes [39], el cual entrega puntos pertenecientes a las esquinas de los contornos detectados. Luego, a partir de estos puntos se obtienen polígonos que se aproximan a las regiones mencionadas (figura 3.2(a)). Esta aproximación se realiza con el algoritmo RDP (Ramer Douglas Peucker) [17], y se concidera como un borde aproximado de la silueta en movimiento.

Para determinar si un polígono corresponde a una persona, se estudió de forma empírica las proporciones de los objetos y personas detectadas. En general, se observa que las personas se diferencian del resto de los objetos detectados (automóviles, ciclistas, animales) por la proporción que hay entre el ancho y alto de la región, siendo la segunda al menos el doble de la primera: ancho < 2(alto). Es por esto que se seleccionan las regiones de interés discriminando según su geometría: ancho y alto. Para esto, se calcula la región rectangular que contiene a cada polígono (figura 3.2(b)). Otra característica utilizada para excluir formas que no pertenecen a personas es el tamaño de la

(a) (b)

Figura 3.2: Región de interés:(a) Polígono que aproxima los bordes. (b) Rectángulo con base en el polígono calculado

región (área en píxeles). Al igual que la proporción, se estableció de forma empírica el rango de área que presenta una persona. Este rango se debe establecer para cada escenario, ya que varía dependiendo de la distancia existente entre la persona y la cámara.

nota: no entiendo

En videos donde conoce por dónde transitarán las personas de interés, se acotó la búsqueda a determinadas regiones del frame, con el fin de procesar sólo lo que se encuentre dentro de estas cotas. Para cada cámara, se estableció una zona rectangular que evita espacios donde no transitan personas (por ejemplo, avenidas de vehículos).

3.2. Almacenamiento y búsqueda de personas

El sistema de reidentificación puede estar en uno de los siguientes estados: *almacenamiento* o *búsqueda*. El primero tiene como objetivo crear una galería con los datos de las personas detectadas, mientras que el segundo reidentifica nuevas personas detectadas con aquellas existentes en la galería. El estado actual del prototipo es determinado por el usuario.

En estado de *almacenamiento*, cada vez que el sistema detecta una nueva persona, se le consulta al usuario si desea agregarla a la galería. Si la elección es afirmativa, se extraen y almacenan las características de la región (sección 3.2.1). En otro caso, la región se descarta y continúa la lectura del video, y así se puede mantener homogénea la cantidad de imágenes por persona.

Por otro lado, cuando el sistema se encuentra en estado de *búsqueda*, se extraen las características de las regiones de interés detectadas, las que se comparan con las de cada persona de la galería con el objetivo de elegir a la persona más parecida (candidato). Aquí se asume que la persona buscada siempre se encuentra almacenada en la galería, por lo que sólo se debe determinar si la imagen seleccionada efectivamente corresponde a dicha persona. Posteriormente, se guarda la imagen buscada junto al candidato elegido, para que más tarde el usuario califique si la reidentificación es correcta (verdadero positivo) o incorrecta (falso positivo). Sin embargo, dado que el cambio entre un estado y otro se establece de forma manual, eventualmente el sistema podría intentar reidentificar a una persona inexistente en una galería. Para estos casos, el usuario debe calificar de forma neutra dichas reidentificaciones, evitando que sean consideradas en el cálculo de rendimiento.

3.2.1. Extracción de características

Antes de extraer las características, la región seleccionada (en espacio RGB) se convierte al espacio HSV, dado que el canal H es invariante a cambios de iluminación [19].

unir

Las características empleadas fueron *keypoints* detectados con SIFT debido a las ventajas que presentan por sobre otras características, tales como propiedades invariantes a cambios de rotación y escala [30].

3.2.2. Comparación de imágenes

La comparación entre imágenes se realiza por medio de sus *keypoints*, cada uno de los cuales se compara con aquellos de la imagen almacenada, con el objetivo de asociarse con el *keypoint* más cercano. La asociación entre un par de imágenes retorna un vector con las distancias entre cada par de *keypoints*. A diferencia de un sistema de votación [23], la re-identificación se realiza sumando la distancia producida entre los pares de *keypoints* más cercanos, eligiendo aquella imagen de la galería que presente la suma más pequeña. La cantidad de distancias a sumar queda como parámetro del sistema.

Capítulo 4

Experimentos y Resultados

Las pruebas presentadas en este capítulo tienen como objetivo determinar los parámetros de configuración del prototipo, comparar los resultados con los obtenidos por un sistema de Re-ID por votación, y evaluar la efectividad del prototipo usando reglas simples (tamaño y relación de aspecto) para discriminar movimientos detectados en videos de vigilancia reales.

4.1. Implementación

La implementación del prototipo utilizó la librería de visión computacional de código libre OpenCV 3.0 [10] utilizando PYTHON 2.7.

El prototipo desarrollado consiste en la implementación de un algortimo de Re-ID basado en la arquitectura de la figura 2.1.

Algoritmo 1: Lectura videos

```
Input: galeria, estado, videos
Output: galeria, reidentificaciones
1 foreach v en videos do
2     stream = abrir(v)
3     while tieneSiguienteCuadro(stream) do
4         frame = obtenerSiguienteCuadro(stream)
5         persona = detectarPersona(frame)
6         if esAlmacenamiento(estado) then
7             galeria = actualizarGaleria(persona)
8         else if esBusqueda(estado) then
9             reidentificaciones = reidentificar(galeria,persona)
```

4.2. Configuración de parámetros

Rendimiento según espacio de colores utilizado.

Algoritmo 2: detectarPersona()

Input: frame, proporcionAltoAncho, areaMinima, areaMaxima, espacioColores
Output: persona

```
1  grises = segmentar(frame)
2  sinRuido = eliminarRuido(grises)
3  sinSombra = eliminarSombra(sinRuido,umbralSombra)
4  puntos = detectarContornos(sinSombra)
5  poligonos = aproximarContornos(puntos)
6  rectangulos = rectanguloContenedor(poligonos)
7  region = seleccionarRegionMaxima(rectangulos,proporcionAltoAncho,areaMinima,
   areaMaxima)
8  detector = SIFT_create()
9  if esRGB(espacioColores) then
10     [keypoints, descriptor] = detector.detectAndCompute(region)
11 else if esHSV(espacioColores) then
12     region = cvtColor(region,COLOR_BGR2HSV)
13     [keypoints, descriptor] = detector.detectAndCompute(region)
14 persona = crearPersona(keypoints, descriptor, region)
```

Algoritmo 3: reidentificar

Input: galeria, persona
Output: reidentificaciones

4.3. Evaluación efectividad de reidentificación

4.3.1. Uso de reglas de discriminación

HSV	RGB
XX %	RGB
XX %	HSV

Cuadro 4.1: Comparación de resultados obtenidos usando RGB y HSV

Precisión	Número de imágenes por persona
XX %	1
XX %	2

Cuadro 4.2: Comparación de resultados obtenidos usando número de imágenes por persona

Precisión	Número de match utilizados
XX %	1
XX %	5
XX %	10
XX %	20

Cuadro 4.3: Comparación de resultados obtenidos usando distinto número de match

Precisión	Uso de filtro
XX %	SI
XX %	NO

Cuadro 4.4: Comparación de resultados obtenidos según uso de filtro

Bibliografía

[1] Bak, S., Corvee, E., Bremond, F., and Thonnat, M. Person re-identification using haar-based and dcd-based signature. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on* (Aug 2010), pp. 1–8.

[2] Bauml, M., Bernardin, K., Fischer, M., Ekenel, H. K., and Stiefelhagen, R. Multi-pose face recognition for person retrieval in camera networks. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on* (2010), IEEE, pp. 441–447.

[3] Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[4] Bedagkar-Gala, A., and Shah, S. K. A survey of approaches and trends in person re-identification. *Image and Vision Computing 32*, 4 (2014), 270–286.

[5] Bellet, A., Habrard, A., and Sebban, M. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709* (2013).

[6] Berdugo, G., Soceanu, O., Moshe, Y., Rudoy, D., and Dvir, I. Object reidentification in real world scenarios across multiple non-overlapping cameras. In *Proc. Euro. Sig. Proc. Conf* (2010), pp. 1806–1810.

[7] Bird, N. D., Masoud, O., Papanikolopoulos, N. P., and Isaacs, A. Detection of loitering individuals in public transportation areas. *Intelligent Transportation Systems, IEEE Transactions on 6*, 2 (2005), 167–177.

[8] Bouma, H., Baan, J., Landsmeer, S., Kruszynski, C., van Antwerpen, G., and Dijk, J. Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall. vol. 8756, pp. 87560A–13.

[9] Bouwmans, T., El Baf, F., and Vachon, B. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science 1*, 3 (2008), 219–237.

[10] Bradski, G. The opencv library. *Doctor Dobbs Journal 25*, 11 (2000), 120–126.

[11] Calderara, S., Prati, A., and Cucchiara, R. Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance. *Computer Vision and Image Understanding 111*, 1 (2008), 21 – 42. Special Issue on Intelligent Visual Surveillance (IEEE).

[12] Canny, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6 (1986), 679–698.

[13] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, IEEE, pp. 886–893.

[14] de Oliveira, I. O., and de Souza Pio, J. L. People reidentification in a camera network. In *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on* (2009), IEEE, pp. 461–466.

[15] DeCann, B., and Ross, A. Can a poor verification system be a good identification system - a preliminary study. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on* (Dec 2012), pp. 31–36.

[16] Dikmen, M., Akbas, E., Huang, T. S., and Ahuja, N. Pedestrian recognition with a learned metric. In *Computer Vision–ACCV 2010*. Springer, 2011, pp. 501–512.

[17] Douglas, D. H., and Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10*, 2 (1973), 112–122.

[18] Ellis, T., Makris, D., and Black, J. Learning a multi-camera topology. In *Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)* (2003), pp. 165–171.

[19] Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (June 2010), pp. 2360–2367.

[20] Gheissari, N., Sebastian, T., and Hartley, R. Person reidentification using spatiotemporal appearance. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, pp. 1528–1535.

[21] Gilbert, A., and Bowden, R. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Computer Vision–ECCV 2006*. Springer, 2006, pp. 125–136.

[22] Gray, D., and Tao, H. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Computer Vision–ECCV 2008*. Springer, 2008, pp. 262–275.

[23] Hamdoun, O., Moutarde, F., Stanciulescu, B., and Steux, B. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on* (2008), IEEE, pp. 1–6.

[24] Hirzer, M., Roth, P., and Bischof, H. Person re-identification by efficient impostor-based metric learning. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on* (Sept 2012), pp. 203–208.

[25] Hirzer, M., Roth, P., Köstinger, M., and Bischof, H. Relaxed pairwise learned metric for person re-identification. In *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7577 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 780–793.

[26] Hu, L., Jiang, S., Huang, Q., and Gao, W. People re-detection using adaboost with sift and color correlogram. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on* (Oct 2008), pp. 1348–1351.

[27] Javed, O., Shafique, K., Rasheed, Z., and Shah, M. Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding 109*, 2 (2008), 146–162.

[28] Khan, S., and Shah, M. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 10 (Oct 2003), 1355–1360.

[29] Kuo, C.-H., Huang, C., and Nevatia, R. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 383–396.

[30] Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[31] Lowe, D. G. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (1999), vol. 2, Ieee, pp. 1150–1157.

[32] Mazzon, R., Tahir, S. F., and Cavallaro, A. Person re-identification in crowd. *Pattern Recognition Letters 33*, 14 (2012), 1828–1837.

[33] Nakajima, C., Pontil, M., Heisele, B., and Poggio, T. Full-body person recognition system. *Pattern recognition 36*, 9 (2003), 1997–2006.

[34] Porikli, F., and Divakaran, A. Multi-camera calibration, object tracking and query generation. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on* (2003), vol. 1, IEEE, pp. I–653.

[35] Prosser, B., Gong, S., and Xiang, T. Multi-camera matching using bi-directional cumulative brightness transfer functions. In *Proceedings of the British Machine Vision Conference* (2008), BMVA Press, pp. 64.1–64.10. doi:10.5244/C.22.64.

[36] Prosser, B., Zheng, W.-S., Gong, S., and Xiang, T. Person re-identification by support vector ranking. In *Proceedings of the British Machine Vision Conference* (2010), BMVA Press, pp. 21.1–21.11. doi:10.5244/C.24.21.

[37] Stein, G. Tracking from multiple view points: Self-calibration of space and time. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* (1999), vol. 1, pp.–527.

[38] Stockman, G., and Shapiro, L. G. *Computer Vision*, 1st ed. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[39] Suzuki, S., et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing 30*, 1 (1985), 32–46.

[40] Teixeira, L. F., and Corte-Real, L. Video object matching across multiple independent views using local descriptors and adaptive learning. *Pattern Recognition Letters 30*, 2 (2009), 157–167.

[41] Vezzani, R., Baltieri, D., and Cucchiara, R. People reidentification in surveillance and forensics: A survey. *ACM Comput. Surv. 46*, 2 (Dec. 2013), 29:1–29:37.

[42] Wang, X., Doretto, G., Sebastian, T., Rittscher, J., and Tu, P. Shape and appearance context modeling. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (Oct 2007), pp. 1–8.

[43] Weinberger, K. Q., and Saul, L. K. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 1160–1167.

[44] Weinberger, K. Q., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res. 10* (June 2009), 207–244.

[45] Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Comput. Surv. 38*, 4 (Dec. 2006).

[46] Zheng, W.-S., Gong, S., and Xiang, T. Person re-identification by probabilistic relative distance comparison. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 649–656.

[47] Zheng, W.-S., Gong, S., and Xiang, T. Reidentification by relative distance comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35*, 3 (2013), 653–668.

[48] Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (2004), vol. 2, IEEE, pp. 28–31.

[49] Zivkovic, Z., and van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters 27*, 7 (2006), 773–780.