ISSN 1870-4069

Recopilación de bases de datos de estacionamientos para aplicaciones en visión computacional

Nisim Hurst Tarrab, Leonardo Chang, Miguel Gonzaalez-Mendoza

Instituto Tecnológico y de Estudios Superiores de Monterrey, México

langheran@gmail.com, {lchang, mgonza}@itesm.mx

Resumen. Un estacionamiento es un ambiente muy bien estructurado en donde usualmente los sistemas de vigilancia se han enfocado. Sin embargo, el conocimiento previo de la estructura del estacionamiento es muchas veces ignorado por los investigadores que hacen uso de las bases de datos tradicionales para entrenar sus algoritmos. Inclusive que estos algoritmos sean correctos y completos, los modelos con los que han sido entrenados o comparados usando este tipo de datos tienden a quedar muy atrás o presentar una naturaleza engañosa. En este artículo proponemos un enfoque basado en tareas, en el que cuidadosamente desglosamos la compleja tarea de detectar comportamientos en estacionamientos entre partes mucho más tratables. Luego, por cada parte proponemos una serie de bases de datos actualmente disponibles en la literatura que pueden ayudar a dominar el problema, cada una desde una perspectiva diferente. Una de las mayores referencias de este artículo ha sido el trabajo de [5] en el que un enfoque mucho más amplio sobre conducción autómata fue tomado.

Palabras clave: visión computacional en estacionamientos, detección de objetos al aire libre, seguimiento de objetos al aire libre, seguimiento de vehículos, bases de datos para estacionamientos, estacionamientos.

Survey on Computer Vision Algorithms and Datasets for Parking Lot Applications

Abstract. Parking lots are well structured environments in which many surveillance systems focus. However, previous knowledge of the parking lot structure is frequently obviated by researchers who make use of traditional datasets for training their algorithms without regard of the inherent data structures stemming from those environments. Even though those algorithms can be correct and complete, models trained or compared by such data tend to fall behind or be misleading in nature. In this paper, a task-based approach is taken in which we carefully breakdown the complex task of detecting behavior in parking lots into

smaller tractable pieces. Each piece falls into a serial processing pipeline. Then, for each of those pieces in the pipeline we propose a set of datasets already available in the literature that can help to tackle the problem, each from a different perspective. A mayor reference for this paper was the work of [5] in which a broader focus on autonomous driving was taken.

Keywords: parking lot, parking lot dataset, outdoors object detection, outdoors object tracking, computer vision classification performance metrics.

1. Introducción

Los vehículos son capital usado en casi todos los aspectos de la vida moderna. Comúnmente interacciones entre vehículos y humanos implican algún evento de importancia para la persona involucrada o inclusive para el resto de las personas alrededor del vehículo. Esta es la razón por la cual tanto esfuerzo está siendo invertido en vigilar estacionamientos.

Hay muchas dificultades en detectar estas interacciones. Cada vehículo ocupa un área relativamente extensa que es frecuentemente imposible de cubrir sin la ayuda de sistemas autónomos. Oclusiones entre vehículos, cambios de perspectiva y condiciones climatológicas variantes dificultan la vigilancia mucho más. Aun así, se requiere un alto tiempo de respuesta en dicho apresurado pero crítico ambiente.

La interacción entre agentes en un estacionamiento puede ser concebida como el comportamiento detectado que a su vez contiene algún significado intrínseco explotable. Este significado depende en qué parte de los agentes están interactuando, dónde y cómo. Al reconocer lo que pasa en un estacionamiento a través de este comportamiento detectado, es fácil tomar una acción preventiva o correctiva. Sin embargo, para llegar a reconocer un comportamiento, una máquina debe descomponer cada paso que los humanos damos por sentado. La serie de pasos generalmente es [15]:

1. Detectar o inferir un área en el estacionamiento
2. Detectar al vehículo o transeúnte
3. Identificar la parte del vehículo involucrada (segmentación semántica)
4. Seguir al vehículo o al transeúnte
5. Detectar el comportamiento entre Vehículo-Estacionamiento, Humano/Vehículo y Vehículo/Vehículo

En este artículo compendiamos brevemente una serie no exhaustiva de artículos que explican su base de datos adaptada a alguno de estos pasos según sus necesidades. Algunas de estas bases de datos vienen anotadas con valores de eficacia comparativos al usar diferentes algoritmos. Algunas de estas comparaciones son retos abiertos dentro de la comunidad, para contribuir o publicar cada quién sus resultados. Así, el investigador interesado en mejorar alguna tarea visual

específica puede primero identificar de manera general la tarea más fácilmente explotable en su serie de tareas y solo usar aquellas bases de datos relacionadas, en vez de desgastarse en explorar todas las bases hoy en día disponibles.

2. Detectar o inferir un área en el estacionamiento

Detectar un área en el estacionamiento está relacionado con una previa segmentación del ambiente. Un estacionamiento puede ser descompuesto según la siguiente taxonomía:

1. Espacio para estacionar
2. Vía para transitar
3. Entradas y Salidas
4. Complemento que no forma parte de los espacios de estacionamiento ni de la vía

Al identificar cada una de estas estructuras se pueden aplicar como evidencia previa en una red bayesiana para clasificar el evento detectado. Se ofrecen bases de datos de ejemplo para cada una de estas clasificaciones exceptuando los puntos de entrada y salida que son comúnmente marcados manualmente.

2.1. Espacio para estacionar

Los espacios para estacionar están usualmente pintados y pueden ser fácilmente identificados por color. Por otro lado, no todos los estacionamientos están en buenas condiciones o necesariamente pintados. Para tartar con estos casos, hay quienes han identificado los lugares a partir de imágenes áreas usando Campos Aleatorios de Markov y Eigenspots [14]. Otras aproximaciones incluyen inferir los lugares a partir de analizar el campo de movimiento [19].

PNNL ParkingLot: La base de datos de PNNL (Pacific Northwest National Laboratory) ParkingLot fue publicada por la Universidad Central de Florida. Esta base cuenta con 3 sequencias de video, una de 1000 frames a 29 frames por segundo a una resolución de 1920x1080, otra de 1500 frames a 30 fps a 1920x1080 y finalmente otra de 4000x3000 a 6 frames por segundo. Cada cámara está posicionada a una diferente altura.

Este dataset contiene tanto a transeuntes como a vehiculos en un espacio concurrido del estacionameinto. Sin embargo, los demarcamientos incluyen solo transeuntes sin incluir vehículos. La primera secuencia tiene 14 transeuntes y la secuencia 2 tiene 13.

Esta base de datos es interesante en el contexto de la vigilancia de estacionamientos dado que enfatizan sus resultados en un comparativo contra otros 9 algoritmos que se enfocan en seguir multiples objetos en ambientes con muchas oclusiones.

Entre la métricas usadas están las ubicuas medidas de seguimiento *CLEAR* [17], es decir la Multiple Object Tracking Accuracy (MOTA) y la Multiple Object Tracking Precision (MOTP). Estas métricas proveen una forma de medir y comparar la eficacia en reconocer y marcar consistentemente a los mismos objetos a través del tiempo. Otras métricas usadas son la MT (mostly tracked), la ML (mostly lost), la IOU (intersection over union) y también la IDS (id switches) propuesta por [7].

PKLot: La base de datos PKLot fue ensamblada por [1] y publicada bajo el auspicio de la Universidad Federal de Paraná de Brazil. Contiene 12417 imágenes a una resolución de 1280x720 en condición soleada, nublada y lluviosa registradas en intervalos de 5 minutos. Su objetivo principal es capturar las diferentes condiciones ambientales. En total cuenta con 695,900 imágenes de espacios de estacionamiento, 43.48 % de ellos ocupados y 56.42 % vacíos.

La base de datos brinda un archivo XML con las cajas delimitadoras y si es que están ocupadas por un vehículo o están libres. Esta base de datos es interesante en el contexto de la vigilancia en estacionamientos porque brinda: 3 diferentes condiciones climáticas en la misma escena, 3 diferentes tomas del estacionamiento, cámaras a diferentes alturas, presencia de sombra de los árboles, sobre exposición a la luz, etc.

Recomendación al detectar espacios de estacionamiento: El reto más importante que enfrentar al detectar espacios de estacionamiento es tratar las oclusiones y mapear correctamente el estacionamiento usando algún tipo de heurística. Calibrar previamente la cámara puede ayudar en sobremanera a tratar las oclusiones al considerar el hecho de que los espacios para estacionar forman un rectángulo de tamaño estándar indivisible. Para reconocer automáticamente estos espacios es posible usar técnicas como Campos Aleatorios de Markov o análisis de movimiento tanto a partir de imágenes aéreas como de perspectivas locales. Se recomienda también probar con otras bases de datos en donde las cámaras están posicionadas a una altura similar. Bases de datos con diferentes condiciones climatológicas pueden ayudar si estamos tratando de probar algoritmos basados en la apariencia de puntos de interés sobre el escenario.

2.2. Vías para circular

Las vías para circular y caminos delimitan una región en dónde varios eventos de interés pueden suceder de forma natural. Características como el color de la vía y su textura pueden ser usadas. Los caminos son un poco más complicados porque carecen de demarcaciones de tránsito. Sin embargo, la elevación, color y movimiento de los vehículos de todos modos pueden ser usados.

Las vías para circular pueden ser fácilmente demarcadas manualmente cuando se cuenta con mapas aéreos por ejemplo de OpenStreetMaps [10] o de Google Maps [9]. Estas fuentes pueden ser consideradas bases de datos por mismas.

Cubren una vasta área geográfica y son actualizadas frecuentemente pero no cuentan con anotaciones a nivel vehículo. Otra opción es inferir las vías para circular y estructuras de movimiento, tal como fue previamente mencionado.

Base de datos KIT AIS: La base de datos de KIT AIS incluye 239 png 895x1036 imágenes áreas en secuencia así como trayectorias de referencia y código fuente para los comparativos de seguimiento. Es útil para desarrollar algoritmos que pretenden inferir estructura a partir del movimiento [13]. Esta base de datos es interesante ya que incluye anotaciones tanto de las trayectorias de vehículos como de caminos.

Base de datos Cityscapes: La base de datos CityScapes brinda anotaciones semánticas en escenarios urbanos para más de 30 clases segmentadas a nivel pixel. Consiste en 5000 imágenes de alta calidad en formato stereo además de 20000 imágenes levemente marcadas. El conjunto de imágenes anotadas corresponde a la veinteava imagen de un video de 30 frames por segundo. También brinda un servidor para comparar que se enfoca en la eficacia en demarcaciones a nivel pixel y a nivel instancia.

Una ventaja de este dataset es su diversidad. Ha sido grabado en alrededor de 50 ciudades en diferentes temporadas del año y con condiciones climáticas que van desde muy buenas hasta medianas.

Esta base de datos es interesante porque también brinda información precisa de coordenadas GPS, movimiento ego, vistas estéreo y temperatura externa. De esta forma puede ayudar a calibrar de forma automática la cámara de un estacionamiento para clasificar secciones de caminos sobre múltiples condiciones atmosféricas. Las anotaciones semanticas son medidas usando la métrica de intersección sobre unión, partiendo desde los pixeles hasta las instancias.

También provee un comparativo con más de 66 resultados a nivel pixel para demarcamiento semántico (usando intersección sobre unión IoU e intersección sobre unión a nivel instancia iIoU) y 14 resultados para la demarcamiento semántico a nivel instancia (usando precisión media). Finalmente, en su sitio web los investigadores pueden enviar sus propios resultados sobre un algoritmo personalizado.

Recomendación al detector vías para circular: Detectar los límites de una vía para circular puede ser visto como un problema más general que detectar los límites de un estacionamiento. La perspectiva de cámara debe ser considerada al momento de elegir el dataset que incluirá los comparativos. Por ejemplo, las bases de datos con vista área son útiles para probar algoritmos basados en movimiento. Sin embargo, carecen de ejemplos útiles para probar algoritmos que traten con oclusiones. Así, se debe escoger las bases de datos basados en el reto principal que nuestro algoritmo se propone a resolver. Los estacionamientos en general pueden beneficiarse tanto de imágenes aéreas como de perspectivas locales, siempre que añadimos un paso previo de calibración que haga que los

movimientos sean ortogonales a la cámara. Sin embargo, esto solo se mantiene en estacionamientos al aire libre para los que se cuenta con imágenes aéreas.

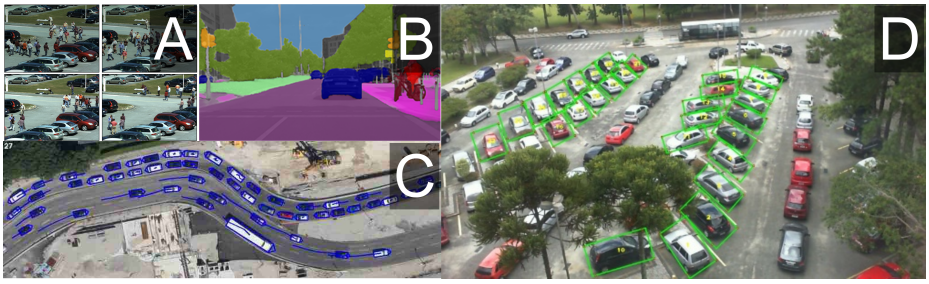

Fig. 1. Bases de datos para segmentar entre vías y espacios de estacionamiento. A) Base de datos PNNL ParkingLot, B) BD. CityScapes, C) BD. KIT AIS, D) BD. PKLot.

La Figura 1 muestra 4 imágenes. La imagen A muestra la secuencia de video "Parking Lot Pizza" en PNNL. La imagen B muestra una de las secuencias para segmentación de vías, leftImg8bit-demoVideo.zip (6.6GB), parte de la base de datos Cityscapes. La imagen C muestra una imagen aérea cortesía del German Aerospace Center dentro de la base de datos KIT AIS. La imagen D pertenece a PKLot y muestra espacios de estacionamiento marcados. Nótese que todos los espacios de estacionamiento son marcados inclusive vacíos.

3. Detectar al vehículo o transeúnte

Para detector objetos, [6] propone una serie de pasos clásicos en la visión computacional, también conocido como *pipeline*. Comprende las siguientes etapas: preprocesar, extraer la región de interés, clasificar al objeto y obtener retroalimentación. En este estudio se dejan de lado el paso de calibrar los sensores, preprocesar la imagen y obtener retroalimentación. En un estacionamiento nos enfocamos en identificar precisamente la región de interes que puede contener un vehículo o transeúnte y posteriormente la detección del tipo de objeto e inclusive modelo del vehículo (en caso de ser un vehículo).

Características útilies en esta caso son el tamaño del área, tipo de vehículo y ubicación relativa del transeúnte. El usar características HAAR en un clasificador Viola-Jones, detectar esquinas usando Harris o usar enfoques basados en partes (quizás al usar la transformada Hough para detectar las luces de frenado) puede ayudar a extraer tanto regiones como puntos de interés y descriptores.

3.1. Evaluación de detección de objetos KITTI 2012

El comparativo para visión computacional KITTI fue construido gracias a un vehículo equipado con una plataforma de filmación por [4]. Esta base de datos permite comparar vehículos, ciclistas y transeúntes en los mismos escenarios.

La base de datos para detección de objetos está dividida en 3 secciones: una para detección de objetos 2D, una para detección de objetos 3D y otra con vista de pájaro. La base de datos de detección de objetos 2D consiste en 7481 imágenes de training y 7518 en imágenes para testing en formato PNG. Están anotadas con una caja limítrofe alrededor de cada objeto y también pueden ser descargadas individualmente.

Las aplicaciones actuales suman más de 131 métodos para detección de coches, 109 métodos para detección de transeúntes y 74 métodos para detección de ciclistas. Tiempos de ejecución también son publicados.

Se brindan 3 subsets de la base de datos en los que cada algoritmo es probado: Fácil (la altura de la caja limítrofe es al menos de 40px y no hay oclusiones), Moderado (la caja limítrofe tiene entre 25px-40px y puede tener oclusiones parciales) y Difícil (de 0 a 25px en la caja limítrofe y puede tener oclusiones difíciles de ver).

Los resultados comparados también comprenden detección de orientación, específicamente 65 algoritmos de detección de orientación de coches, 46 en orientación de transeúntes y 40 en orientación de ciclistas.

La base de datos es útil en el contexto de vigilancia en estacionamientos por el comparativo de algoritmos que presenta en detección de coches y su orientación. La orientación en coches puede ser una característica dominante para detectar un comportamiento extraño de un vehículo, por ejemplo un coche violando el sentido de la vía establecido.

3.2. Base de datos de tráfico en el MIT

La base de datos de tráfico en el MIT fue ensamblada con el único objetivo de entrenar un detector de transeúntes genérico [20]. La perspectiva en vista de pájaro es asumida dentro de todas las imágenes. El movimiento de transeúntes y vehículos está regulado por la estructura inherente de las calles y siguen cierto patrón de movimiento.

Comprende 20 clips de video. Cada video es de 90 minutos y contiene vehículos y transeúntes. Fueron filmadas de día por una cámara fija al aire libre de gran altura.

La base de datos viene anotada con las cajas limítrofes de transeúntes, tamaño de la imagen y rango de frames en los que aparecen. Los datos están divididos en 2 archivos, uno para entrenamiento y el otro para pruebas, cada uno apuntando a 10 clips de 20. Ambos archivos están en formato MATLAB.

A pesar de que solo los datos reales de transeúntes son brindados, la base de datos es interesante porque en su documentación proponen un nuevo algoritmo para adaptar un detector genérico de transeúntes a una escena específica a partir

de otras pistas como la estructura de la calle. Así, en escenas de estacionamientos, donde dichas estructuras son comúnmente brindadas por humanos, este algoritmo de re-entrenamiento puede mejorar la precisión de detección considerablemente.

En el caso de KITTI, vimos que siguen el formato de PASCAL VOC para medir la intersección entre las áreas predichas y los datos reales. Sin embargo, en este caso solo ROC (curva de recepción de características operativas) fue propuesto para medir la precisión en la caja limítrofe.

3.3. Base de datos de Statlog (siluetas vehiculares)

La base de datos Statlog fue compilada en el Turing Institute de Glasgow en 1986 por JP Siebert y publicada como parte de [12]. Pretende capturar la silueta actual de un vehículo como región de interés binaria sin importar la textura o segmentación semántica a un grado mayor. Incluye cuatro modelos de vehículo: el bus double deacker, la van Chevrolet, el Saab 900 y el Opel Manta 400.

Statlog es una base de datos interesante porque su único objetivo es poder detectar la silueta de un vehículo a partir de su binarización y las imágenes están escaladas para caber en una matriz de 128x128. Un algoritmo de extracción de background puede usar un método intermedio de aprendizaje máquina entrenado sobre esta base de datos para simplemente filtras aquellas regiones de interés que no son vehículos y luego proceder con algoritmos más complicados sobre esas regiones.

3.4. Recomendación para detector vehículos

Una aproximación a partir de características locales puede ayudarnos a detectar vehículos muy rápido. La base de datos KITTI puede ayudar en comparer algoritmos relacionados con extraer orientación y estimar en 3D. Es recommendable que usemos esta base de datos si nuestros algoritmos están esperando imágenes bien formadas y escaladas. La base de datos del MIT fue grabada a partir de condiciones ideales donde la vista de pájaro es asumida. Así, no recomendamos usar esta base de datos para entrenar algoritmos basados en instancias. Sin embargo, si nuestro algoritmo toma en cuenta las posiciones en las cuales el objeto es detectado, entonces esta base de datos es altamente recomendable. La base de datos Silhouette debe ser usada solo en casos en los que nos convenga filtrar la región de interés que va a ser procesada por algoritmos mucho más complejos. Por otra parte, la base de datos Silhouette no ayuda en extraer otras características basadas en apariencia exceptuando el contorno. Exploraremos un enfoque más complicado que usa un detector de partes en la siguiente sección.

4. Detectar parte en el vehículo

Una forma de clasificar es primero partir objetos complejos en objetos más simples que sean más sencillos de entrenar con menos datos. Los modelos de

Partes Deformables y el de *Formas Implícitas* ambos extraen características, uno a partir de características HOG y el otro a partir de entradas en un codebook. Comúnmente un modelo de contexto es usado por separado para aprender el contexto indispensable a la hora de tratar con oclusiones [5]. Dicho esto, conviene evaluar dentro del paso de clasificar la acción cuál es la parte del objeto que está dirigiendo la acción [22].

4.1. Base de datos de partes en PASCAL VOC

La base de datos PASCAL VOC (Visual Object Classes) consiste en diferentes imágenes de objetos que están semánticamente partidos en partes. Fue ensamblada alrededor del año 2005 a partir de un reto auspiciado por Flickr periódicamente. Sin embargo, el programa terminó en el año 2012 y ya no se siguieron agregando objetos. Tiene alrededor de 11,530 imágenes con 27,450 secciones de interes marcadas para 20 clases distintas, *incluyendo vehículos*. Así, las anotaciones sobre datos reales consisten en sus clases y cajas limítrofes. También incluye más de 11 etiquetas para denotar la acción siendo realizada.

El reto consiste en 4 tareas principales: Clasificación, Detección, Segmentación y Reconocimiento de la Acción.

Para el reto de segmentación se introduce el uso de *recall* en detección de objetos, donde la caja limítrofe 2D es correcta si se empalma al menos con 50 % de la caja limítrofe real. Bootstraped averaged precisión (AP) y en rango son usados para medir los resultados en detección, clasificación de entidades y clasificación de acciones.

En 2012 los resultados finales publicados en [3] incluyeron más de 12 métodos que por lo menos participaron en una de estas cuatro tareas. El hecho de que exista fertilización entre cada tarea, esto es, detectores siendo usados para segmentar y clasificar, y segmentación no supervisada es lo que hace que esta base de datos sea interesante en estacionamientos. Además, la estructura XML introducida en este reto es todavía usada por otros retos, por ejemplo ImageNet o KITTI.

4.2. Microsoft COCO

Microsoft COCO es también un esfuerzo conjunto de toda una comunidad [8]. Microsoft COCO es significativamente mayor que PASCAL VOC. Tiene 91 clases de objetos y 328,000 imagenes anotadas con 2.5 millones de regiones de interés.

Esta base de datos ha sido anotada usando segmentación por cada instancia que aparece en la imagen. Los objetos están anotadas con datos reales que incluyen detectar de objetos, segmentar, detectar puntos clave en las personas, generar cosas del ambiente y generar encabezados. Para la tarea de detectar objetos tiene más de 272 clases, incluyendo una clase explicita para caminos y calles.

Estas anotaciones pueden ser descargadas y accedidas a través de paquetes de MATLAB, Python y Lua que usan su API. También pueden ser descargadas

a través de su sitio Web. Para usuarios finales brinda una novedosa interfaz que en su momento ayudo a la comunidad a etiquetar las imágenes e involucrarse con el proyecto.

La base de datos COCO es la más extensa de su clase, al contener anotaciones de caminos y vehículos en una misma imagen. Son comunes las oclusiones como la mostrada en la imagen previa. Para evaluar usan intersección sobre unión y precisión media (un mínimo de 0.5 es requerido).

4.3. Recomendación al detectar alguna parte del vehículo

La partes de un vehículo pueden ser una gran característica para detectar o seguir a un vehículo. Mientras PASCAL VOC puede ayudar a detectar segmentos de vehículos o transeúntes, Microsoft COCO puede ayudar a detectar no solo segmentos, sino también entrenar algoritmos que usen encabezados textuale, puntos clave en personas y cosas del ambiente. Sin embargo, a menos de que estamos lidiando con modelos que son deformables, por ejemplo un transeúnte o la puerta de un coche que se mueve, su costo se vuelve prohibitivo solo para la tarea de detección. No obstante, para tareas de seguimiento si funcionan mejor. Una vez que la parte de un vehículo es detectada, es mucho más fácil tratar con oclusiones sobre esa parte que ya estamos siguiendo si también estamos siguiendo otras partes que se asumen indivisibles y tenemos sus posiciones relativas. En conclusión, usar la parte del vehículo para detección es útil solamente si el algoritmo busca también hacer tracking.

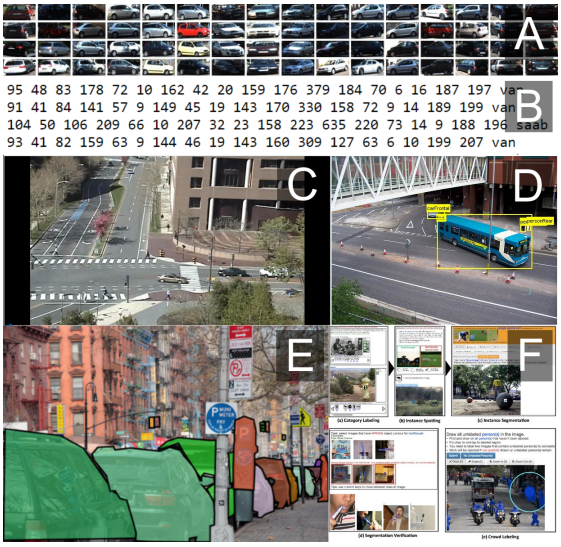

Fig. 2. Bases de datos para detectar vehículos o transeúntes. A) Base de datos KITTI 2012, B) BD. Statlog, C) BD. Tráfico MIT, D) BD. PASCAL VOC, E) BD. Microsoft COCO, F) Interfaz para anotaciones en Microsoft COCO.

La Figura 2 muestra 6 imágenes. La imagen A muestra diferentes orientaciones de coches dentro de KITTI 2012. La imagen B muestra un ejemplo de una instancia en la base de datos Statlog. La última columna corresponde a su clase. La imagen C muestra la única vista disponible dentro de la base de datos de tráfico del MIT. La imagen D muestra un ejemplo de decomposición semántica en la base de datos de partes PASCAL VOC. La imagen E muestra un ejemplo de oclusión entre vehículos dentro de la base de datos COCO, muy similar a lo que vemos comúnmente en un estacionamiento. La imagen F muestra como es la interfaz que COCO brinda a la comunidad para colaborar con anotaciones por cada una de las tareas.

5. Seguir al vehículo o al transeúnte

Un truco bastante usado en la literatura de visión por computadora es el de minimizar las oclusiones al dar contexto a la imagen a partir de seguir otros objetos que ya han aparecido en el video al usar *modelos de movimiento en capas* [18]. Inclusive el seguimiento a vehículos ha sido usado para deducir las acciones que se están llevando a cabo en la escena [16]. Una métrica de precisión comúnmente usada para este tipo de tarea es la de Multiples Object Tracking Accuracy MOTA y la de Multiples Object Tracking Precision que fueron introducidas en [2].

5.1. MOTChallenge 2016

La base de datos del reto MOTChallenge es un comparativo compuesto de 42 secuencias de video. Fueron filmados tanto con cámaras estáticas como dinámicas. Las anotaciones incluyen transeúntes, vehículos, bicicletas, motocicleta y otros ocluidores [11].

En total el benchmark (comparativo) del 2017 tiene 21 secuencias de video, que constituyen 17,757 frames en los cuales hay 2,355 trayectorias y 564,228 cajas fronterizas anotadas.

5.2. Evaluación de seguimiento de objetos KITTI 2012

El comparativo de seguimiento de objetos KITTI 2012 consiste en 21 secuencias de video para training y 29 para testing. Hay 8 clases en total anotadas pero dentro del comparativo solo se usan transeúntes y coches.

Este comparativo está abierto para que cada quien publique sus resultados y de hecho brinda 43 resultados de algoritmos para coches y 21 para transeúntes.

Actualmente el más interesante de estos algoritmos para detección de coches es youtu que está basado en seguimiento a través e detección y que será posteriormente presentado en la competencia de Visión Computacional y Reconocimiento de Patrones.

5.3. Base de datos PETS Arena

La base de datos PETS Arena contiene 14 secuencias de video que muestran 22 comportamientos actuados alrededor de un vehículo estacionado. La base de datos fue filmada usando 4 cámaras RGB que cubren los 360 grados de campo visual. Está en un formato de resolución en 1280x960 a 30 frames por segundo. Cada uno de los 14 videos tiene alrededor de 96 segundos.

5.4. Reto de seguimiento DETRAC

El reto DETRAC ha sido abierto este año 2017. Incluye retos para detección y para seguimiento. La base de datos incluye 10 horas de video en 24 lugares de China. Los videos fueron grabados a 25 frames por con una resolución de 960x540. 8,250 vehículos fueron marcados manualmente. Comprende 4 categorías de vehículos: coche, autobús, van y otros. Considera 4 categorías climatológicas, es decir, nublado, noche, soleado y lluvioso. Los vehículos son agrupados en tres escalas dependiendo del tamaño de su caja limítrofe: pequeña (0-50 pixeles), mediana (50-100 pixeles) y grandes (más de 150 pixeles).

El comparativo enfatiza la necesidad de medir precisión al considerar conjuntamente detección y tracking. Por lo tanto, introducen las métricas UA-DETRAC [21] que están basadas en los métricas CLEAR MOT al medir el área bajo la curva *precisión-recall*. De tal suerte estas métricas fueron llamadas con un prefijo PR, específicamente: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP y PR-FN.

Esta base de datos es interesante en el contexto de estacionamientos ya que la mayoría de los escenarios en ella presentadas vienen de cámaras en postes callejeros que son muy similares a aquellos postes que se encuentran en los estacionamientos. Más aún, los resultados y métricas del reto están organizados para presentar los algoritmos de detección a la par de los de seguimiento.

La combinación más exitosa hasta los momentos es la de CompACT+GOG que obtiene un PR-MOTA relativamente superior de 14.2 % [21].

5.5. Recomendación en seguir un vehículo o transeúnte

Mientras que los vehículos se mueven a grandes velocidades, los transeúntes se mueven a una velocidad mucho más baja. Sin embargo, es más común ver oclusiones dominantes en pequeños objetos como en el seguimiento a transeúntes, mucho más que en el seguimiento de vehículos. Por lo tanto, se recomienda usar un enfoque especial para cada tipo de entidad, es decir, si tu algoritmo se apoya fuertemente en seguir acciones basado en interacciones humanas, entonces puedes usar PETS Arena para probar tus algoritmos de tracking basado en SVM o en características HOG. Sin embargo, si tu algoritmo usa las interacciones vehiculares entonces te conviene usar las bases de datos de MOT o KITTI. Estas bases de datos brindan un mayor comparativo de precisión para redes convolucionales que han sido probadas ser efectivas en seguir vehículos a altas velocidades. También es recomendable usar la base de datos DETRAC ya que

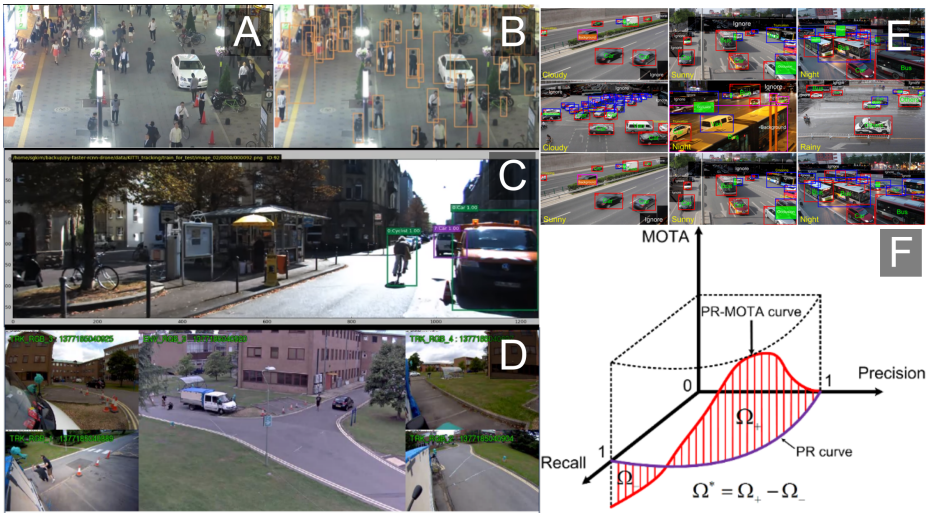

Fig. 3. Bases de datos para seguir vehículos. A) Base de datos MOTChallenge 2016, B) BD. MOTChallenge 2016 con cajas limítrofes, C) BD. KITTI 2012, D) BD. PETS Arena, E) BD. DETRAC, F) métrica PR-MOTA en BD. DETRAC.

brinda comparativos con métricas que se componen tanto de detección como de tracking, sin mencionar el gran número de instancias y diversidad de clases disponibles para entrenar.

La Figura 3 muestra 6 imágenes. Las imágenes A y B muestran la escena MOT17-04-SDP dentro del reto MOT (MOTChallenge) en donde hay muchos transeúntes. La imagen B es lo mismo que la A solo que cuenta con las cajas limítrofes. La imagen C muestra parte del material de filmación tomado por un vehículo en movimiento dentro de la base de datos KITTI, con cajas limítrofes indicando la presencia de un ciclista y dos coches. La imagen D muestra una escena de violencia actuada en un estacionamiento dentro de la base de datos PETS Arena. Las imagenes E y F pertenecen al reto DETRAC. La imagen E muestra las cajas limítrofes al seguir objetos bajo múltiples condiciones atmosféricas. La imagen F muestra la curva PR-MOTA 3D usada por primera vez en el reto DETRAC.

6. Conclusiones

Esta revisión enumera algunas bases de datos útiles para que el desarrollador pueda usarlas al mejorar alguna parte específica de su algoritmo dentro del *pipeline* de visión computacional en estacionamientos.

La Figura 4 muestra un análisis cuantitativo de todas las bases de datos presentadas. Inicialmente, el investigador debe fijarse en la cantidad de clases que su tarea específica está clasificando al igual que el tamaño mínimo de la base de datos para compararla con las que están disponibles en la literatura.

Base de datos	clases	img / vid	Longitud prom.	algoritmos	métricas
PNNL ParkingLot	1	3	1250f	8	5 MOTA
PKLot	2	12417	NA	0	1 OE
Base de datos KIT AIS	2	239	NA	3	2 AP,F1
Base de datos CityScapes	30	5000f (stereo)	NA	68pil +14inl	3 IoU,iIoU,AP
Evaluación de detección de objetos KITTI 2012	3	7481+7518	NA	471	1 AP
Base de datos de tráfico en el MIT	1	20	90min	3	1 ROC
Base de datos Statlog (siluetas vehiculares)	1	946	NA	none	1
Base de datos de partes en PASCAL VOC	20	11,530	NA	73	2 AP,IoU
Microsoft COCO	91	328,00	NA	30+9+5+80	5
MOTChallenge 2016	5	21+21	45seg	24	5 MOTA
Evaluación de seguimiento de objetos KITTI 2012	8	21+29	?	43+21	5 MOTA
Base de datos PETS Arena	22	14	45seg	?	?
Reto de seguimiento DETRAC	4	60+40	10 hr total	10	5 PR-MOTA

Fig. 4. Análisis cuantitativo entre las distintas bases de datos presentadas.

Así mismo, debe buscar una base de datos que cuente con suficientes algoritmos para permitirle hacer un análisis estadísitco de que tan bien se puede comportar su algoritmo, e.g. usando ANOVA (Analysis of Variance). Finalmente, debe fijar una métrica acorde con su tarea.

A continuación, se enumeran algunas de las conclusiones más importantes que brotan de este estudio:

1. Personas de diferentes sectores abordan el tema de visión computacional en estacionamientos con diferentes metas en mente. Dos de las preguntas más importantes para empezar son: 1. Estás preocupado por mejorar la detección, seguimiento o detección de comportamiento? 2. Quieres modelar alguna característica difícil en estacionamientos reales o solo buscas mejorar una parte específica del algoritmo en condiciones controladas?
2. Un espacio de estacionamiento puede ser marcado trivialmente por alguien del personal de seguridad. Sin embargo, esta solución puede también ser muy retadora si consideramos el alto número de cámaras o cuando algunas de ellas permiten movimiento como las PTZ. Por lo tanto, una calibración previa de las cámaras, análisis de movimiento y alguna heurística para tomar en cuenta el tamaño estándar de los espacios son recomendables.
3. En el contexto de tratar la vía de transito también es posible beneficiarse de una previa calibración de las cámaras. Sin embargo, estos constituyen problemas más generales que requieren otras heurísticas provenientes del movimiento, datos satelitales GPS o apariencia de la vía.
4. Para entrenar un algoritmo de detección de vehicular y de transeúntes es posible basarnos las posiciones en donde se detectan las instancias. Hay bases de datos propicias para cada una de estas tareas. Puedes seguir una aproximación piramidal al problema, en la cual primero solo detectas las siluetas o regiones de interés que puedan contener digamos un vehículo, y después proceder con algoritmos más costosos computacionalmente hablando.
5. La detección de un vehículo a través de sus partes es en general costosa. Sin embargo, al momento de seguirlo dentro de una escena con obstáculos visuales, es posible que esta aproximación tenga ventaja porque permite

un mejor trato de las oclusiones por cada parte. Por otro lado, a nivel transeúnte la detección por partes deformables ha sido ampliamente usada en la literatura.

6. Dado que en un estacionamiento podemos encontrar gran variedad de velocidades en diferentes tramos de la vía o inclusive dentro de los espacios para estacionar, recomendamos hacer el seguimiento de vehículos por separado de los transeúntes. Las métricas objetivo para seguimiento deben ser definidas en aras de seleccionar cuál de las bases de datos usar, es decir, usar una que enfatiza interacciones entre humanos contra una que enfatiza interacciones entre vehículos, cada una con su propio conjunto de algoritmos. DETRAC es una excelente base de datos para probar tanto detección como seguimiento al mismo tiempo.
7. Un estacionamiento es un ambiente en donde rara vez el fondo del escenario cambia. Sin embargo, en primer plano los objetos que mueven a un rango muy variable de velocidades. Por lo tanto, es poco recomendable usar el mismo algoritmo para detectar y seguir vehículos al mismo tiempo que transeúntes.
8. Cada algoritmo desarrollado para cierto tipo de cámara debe de ser probado con las bases de datos que contribuyen a la contribución que se espera del algoritmo y que inclusive llevan el mismo tipo de cámara. Así, debes escoger con cuidado que bases de datos valen la pena usar cuando se desarrolla un algoritmo en visión computacional para estacionamientos. En este estudio vimos que las bases de datos varían en altura de la cámara, perspectiva, clima, datos reales o anotaciones proporcionadas, velocidad de la instancia, etc.
9. Las métricas a usar al comparar los algoritmos son igualmente importantes de definir como el conjunto de bases de datos a usar. Sin un buen conjunto de métrica, no solo los comparativos son imposibles de realizar, sino que también la contribución específica de tu algoritmo queda difusa.

Referencias

1. Almeida, P.R., Oliveira, L.S., Britto, A.S., Silva, E.J., Koerich, A.L.: Pklot - a robust dataset for parking lot classification the pklot dataset. Expert Systems with Applications 42(11), 1–6 (Jul 2015)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. Eurasip Journal on Image and Video Processing 2008 (2008)
3. Everingham, M., Eslami, S.M.A., VanGool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision 111(1), 98–136 (2014)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving: The kitti vision benchmark suite (z) (2012)
5. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art (Apr 2017)
6. Krig, S.: Computer Vision Metrics. Apress (2014)
7. Li, Y., Li, Y., Huang, C., Nevatia, R.: Learning to associate: Hybridboosted multi-target tracker for crowded scene. IN CVPR (2009), `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.309.8335`

8. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (May 2014), http://arxiv.org/abs/1405.0312
9. Lin, T.Y., Yin Cui Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization, pp. 5007–5015. IEEE (Jun 2015)
10. Mattyus, G., Wang, S., Fidler, S., Urtasun, R.: Enhancing Road Maps by Parsing Aerial Images Around the World, pp. 1689–1697. IEEE (Dec 2015), http://ieeexplore.ieee.org/document/7410554/
11. Milan, A., Leal-Taixe, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking pp. 1–12 (2016), http://arxiv.org/abs/1603.00831
12. Repository, U.M.L.: Statlog (vehicle silhouettes) data set (2000), https://archive.ics.uci.edu/ml/datasets/Statlog+(Vehicle+Silhouettes)
13. Schmidt, F.: Kit ais data set (2017), http://www.ipf.kit.edu/downloads_data_set_AIS_vehicle_tracking.php
14. Seo, Y.W., Urmson, C.: A hierarchical image analysis for extracting parking lot structures from aerial images (2009), http://ai2-s2-pdfs.s3.amazonaws.com/0db7/5ab657f4d1061878582dce9c8a10284210d8.pdf
15. Sivaraman, S., Trivedi, M.M.: Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Transactions on Intelligent Transportation Systems 14(4), 1773–1795 (2013)
16. Sivaraman, S., Trivedi, M.M.: Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Transactions on Intelligent Transportation Systems 14(4), 1773–1795 (Dec 2013), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6563169
17. Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., Soundararajan, P.: The CLEAR 2006 Evaluation, pp. 1–44. Springer Berlin Heidelberg (2006), http://link.springer.com/10.1007/978-3-540-69568-4_1
18. Szeliski, R.: Computer vision : Algorithms and applications. Computer 5, 832 (2010), http://research.microsoft.com/en-us/um/people/szeliski/book/drafts/szelski_20080330am_draft.pdf
19. Urman, Y., Yampolsky, T.B., Cohen, R.: Unsupervised detection of available parking spots, pp. 1–5. IEEE (Nov 2016), http://ieeexplore.ieee.org/document/7806204/
20. Wang, M., Wang, X.: Automatic adaptation of a generic pedestrian detector to a specific traffic scene. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 3401–3408 (2011)
21. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.C., Qi, H., Lim, J., Yang, M.H., Lyu, S.: Ua-detrac: A new benchmark and protocol for multi-object detection and tracking (2015), http://arxiv.org/abs/1511.04136
22. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts, pp. 1331–1338. IEEE (Nov 2011), http://ieeexplore.ieee.org/document/6126386/