



INTERNATIONAL DOCTORAL
SCHOOL OF THE USC

Lorenzo
Vaquero Otal

PhD Thesis

Visual Multi-Object Tracking through Deep Learning

Santiago de Compostela, 2023



TESE DE DOUTORAMENTO

VISUAL MULTI-OBJECT TRACKING THROUGH DEEP LEARNING

Lorenzo Vaquero Otal

ESCOLA DE DOUTORAMENTO INTERNACIONAL DA UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA

PROGRAMA DE DOUTORAMENTO EN INVESTIGACIÓN EN TECNOLOXÍAS DA INFORMACIÓN

SANTIAGO DE COMPOSTELA
2023



DECLARACIÓN DO AUTOR DA TESE

Don Lorenzo Vaquero Otal

Título da tese: Visual Multi-Object Tracking through Deep Learning

Presento a miña tese, seguindo o procedemento adecuado ao Regulamento, e declaro que:

- 1. A tese abarca os resultados da elaboración do meu traballo.*
- 2. De ser o caso, na tese faise referencia ás colaboracións que tivo este traballo.*
- 3. Confirmo que a tese non incorre en ningún tipo de plaxio doutros autores nin de traballos presentados por min para a obtención doutros títulos.*
- 4. A tese é a versión definitiva presentada para a súa defensa e coincide a versión impresa coa presentada en formato electrónico.*

E comprométome a presentar o Compromiso Documental de Supervisión no caso de que o orixinal non estea na Escola.

En Santiago de Compostela, 22 de maio de 2023

Asdo. Lorenzo Vaquero Otal



AUTORIZACIÓN DO DIRECTOR/TITOR DA TESE **Visual Multi-Object Tracking through Deep Learning**

Don Manuel Mucientes Molina, Profesor Titular da Área de Ciencia da Computación e Intelixencia Artificial da Universidade de Santiago de Compostela

Don Víctor Manuel Brea Sánchez, Profesor Titular da Área de Electrónica da Universidade de Santiago de Compostela

INFORMAN:

*Que a presente tese correspóndese co traballo realizado por **Don Lorenzo Vaquero Otal**, baixo a nosa dirección/titorización, e autorizamos a súa presentación, considerando que reúne os requisitos esixidos no Regulamento de Estudos de Doutoramento da USC, e que como directores/titores desta non incorre nas causas de abstención establecidas na Lei 40/2015.*

De acordo co indicado no Regulamento de Estudos de Doutoramento, declaramos tamén que a presente tese de doutoramento é idónea para ser defendida en base á modalidade de Monográfica con reprodución de publicacións, nas que a participación do doutorando foi decisiva para a súa elaboración e que as publicacións se axústan ó Plan de Investigación.

En Santiago de Compostela, 22 de maio de 2023

Asdo. Manuel Mucientes Molina
Director/a tese

Asdo. Víctor Manuel Brea Sánchez
Director/a tese



Á miña familia

*We're not lost. We're
locationally challenged!*
John M. Ford

AGRADECEMENTOS

En primeiro lugar, quero expresar o meu mais sincero agradecemento aos meus directores de tese Manuel Mucientes Molina e Víctor M. Brea Sánchez, pola confianza e os consellos que me brindaron e que fixeron que este traballo fose posible. Quixera agradecer tamén ao Departamento de Electrónica e Computación, así como ao CiTIUS, pola excelente contorna de traballo e os recursos que fixeron posible o desenvolvemento desta tese. Tamén me gustaría mencionar ao Dr. Xavier Alameda-Pineda e a todo o grupo de Inria Grenoble por acollerme durante a miña estadia.

Non podía faltar un agradecemento especial ás amizades que fixen durante esta etapa por todos os laboratorios polos que fun pasando, tanto a aqueles que xa liscaron coma aos que aínda me seguen a aturar. Tamén agradecer a todo o pack o seu soporte incondicional, así como a todo o persoal de apoio do centro, pois sen eles o CiTIUS non sería o que é. E tampouco me esquezo de DGL, que non é o compañeiro de traballo que merezo, pero nin eu nin ninguén. Estaríamos mellor sen él. Quixera darlle as grazas tamén a todos os amigos da carreira e da aldea, así como ás marabillosas amizades que fixen no colexio e no instituto, e que aínda a día de hoxe perduran. Por último, agradecer o apoio fundamental da miña familia, tanto de meus pais e Niko coma de María, que é parte dela. Non teño palabras para o cariño que me deron non só durante a etapa de tese, senón durante todo o que veu antes e o que virá despois. Grazas.

Primos secretos.

This work has received financial support from the Spanish Ministerio de Ciencia e Innovación (TIN2017-84796-C2-1-R, RTI2018-097088-B-C32, PID2020-112623GB-I00, and PID2021-128009OB-C32), and the Galician Consellería de Cultura, Educación e Universidade (ED431C 2017/69, ED431C 2018/29, ED431G 2019/04, ED431C 2021/048, and accreditation 2016-2019, ED431G/08). These grants are co-funded by the European Regional Development Fund (ERDF). Lorenzo Vaquero Otal is supported by the Spanish Ministerio de Universidades under the FPU national plan (FPU18/03174). We also gratefully acknowledge the support of NVIDIA Corporation for hardware donations used for this research.

Resumo

A capacidade humana para recoñecer e percibir sen esforzo os elementos da nosa contorna visual contrasta marcadamente cos complexos desafíos computacionais que esta tarefa supón para os sistemas artificiais. O desafío abarca non só comprender senón tamén extraer información significativa da escena. Este proceso sofisticado, que segue sendo difícil de reproducir para as máquinas, foi obxecto de amplas investigacións no campo da visión por computador durante as últimas décadas.

As orixes da visión por computador remóntanse a principios dos anos 60, cando os investigadores comezan a explorar a posibilidade de ensinar ás máquinas a “ver” e interpretar información visual. Os primeiros intentos céntranse no recoñecemento básico de patróns e na detección de bordos nas imaxes, explotando esa información para identificar formas xeométricas sinxelas e discernir a estrutura dos obxectos nunha escena determinada. Estes esforzos preliminares proporcionan información valiosa sobre a natureza dos datos visuais e abren o camiño para afrontar tarefas de visión por computador cada vez máis complicadas. Así, cos avances na potencia computacional, nas tecnoloxías de imaxe e o desenvolvemento de algoritmos de aprendizaxe automática robustos, o campo evoluciona gradualmente cara a desafíos máis complexos como a detección de obxectos, a comprensión de escenas e a segmentación semántica.

Non obstante, a información extraída dunha soa imaxe a miúdo pode ser limitada, proporcionando só unha comprensión parcial da escena. En consecuencia, o campo da visión por computador comeza a prestar máis atención á análise de vídeo a principios dos anos 70, posto que permite incorporar unha dimensión adicional: o tempo. Isto facilita a creación de algoritmos que aproveitan a información temporal para mellorar a comprensión da escena e os procesos de toma de decisións. En concreto, xorde a tarefa do seguimento visual de obxectos ou *tracking*, que consiste en preservar a identidade dun ou máis elementos durante os fotogramas dun vídeo. Isto non só permite o recoñecemento das rexións correspondentes a distintos obxectos, senón tamén a análise de traxectorias, cambios de forma e comportamentos ao longo das secuencias de vídeo.

A tarefa do seguimento de obxectos é agora a pedra angular de numerosos sistemas, incluíndo dispositivos de videovixilancia, robots e vehículos autónomos. Existe unha ampla gama de métodos para abordar este desafío, cada un adaptado aos requisitos específicos do sistema

e contorna de aplicación. Dentro da literatura, xorden dous enfoques principais: o seguimento dun só obxecto (*single-object tracking*, SOT) e o seguimento multiobxecto (*multi-object tracking*, MOT), que tradicionalmente se desenvolven en disciplinas separadas.

O obxectivo desta tese de tese é salvar a brecha entre SOT e MOT e aplicar técnicas interdisciplinares para avanzar no estado do coñecemento en MOT. Esta investigación busca crear solucións máis eficientes e precisas que se poidan aplicar facilmente nunha gran variedade de escenarios reais.

Como o seu nome indica, o seguimento dun só obxecto (SOT) preocúpase principalmente de seguir un determinado elemento nunha secuencia. Normalmente, asúmese que se coñece a posición inicial do obxecto, e o obxectivo de calquera algoritmo SOT é manter a identidade do obxecto durante toda a secuencia de vídeo ao tempo que proporciona predicións das súas caixas delimitadoras, ou *bounding boxes*. O obxecto normalmente non pertence a unha categoría coñecida de antemán, polo que estes algoritmos deben ser capaces de manexar obxectos arbitrarios, é dicir, independentes da clase.

Dada a natureza das aplicacións nas que habitualmente se empregan, os rastreadores dun só obxecto adoitan ser en liña ou *online*. Isto significa que xeran predicións fotograma a fotograma sen ter acceso a momentos futuros da secuencia. Xunto a isto, hai un gran énfase en garantir velocidades de inferencia rápidas para acomodar as demandas de rendemento en tempo real existentes en moitos contextos. En consecuencia, a maioría dos SOT establecen inicialmente un modelo da aparencia do obxecto, que posteriormente se compara fotograma a fotograma para determinar as súas coordenadas precisas. Para mellorar a eficiencia, en lugar de buscar en todo o fotograma, estes rastreadores céntranse en rexións específicas coa maior probabilidade de conter o elemento.

Os primeiros traballos no seguimento dun só obxecto baséanse en técnicas clásicas de visión por computador. Os métodos de fluxo óptico calculan o movemento do obxecto atopando desprazamentos de píxeles entre fotogramas. Os esforzos para modelar grupos de píxeles e obxectos rápidos serven como base para outros enfoques, como o uso de métodos baseados en filtros bayesianos. Estes predín o estado actual en función de estados anteriores e perfeccionan aínda máis esas predicións mediante novas observacións. Os filtros de partículas son un tipo de filtro bayesiano que amplía a capacidade dos métodos anteriores abordando escenarios que requiren representacións non gaussianas. Por último, coa introdución de MeanShift, xorden métodos baseados en filtros ou *kernels*, que buscan a rexión con maior semellanza baseándose en histogramas de cor.

As limitacións destes enfoques clásicos inclúen a súa susceptibilidade a cambios nas condicións de iluminación, deformacións, movementos rápidos e oclusións. Non obstante, sentan as bases das técnicas contemporáneas e moitas das súas contribucións seguen sendo moi pertinentes. Así, as estratexias actuais de seguimento dun só obxecto dependen predominantemente

de filtros de correlación discriminante (DCF) e modelos de aprendizaxe profunda.

As primeiras aplicacións dos filtros de correlación remóntanse ao procesamento de sinais, onde o obxectivo é calcular a correlación entre dous sinais. No contexto do seguimento, os rastreadores baseados en DCF predín a posición do obxecto adestrando un filtro que distingue entre o elemento de interese e o fondo da escena. As primeiras versións como MOSSE céntranse nunha única función e representan un elemento arbitrario cun único filtro. As propostas son moi eficientes, xa que as correlacións convértense ao dominio da frecuencia mediante a Transformada Rápida de Fourier (FFT) para acelerar os cálculos.

Baseados nesta proposta, con CSK introdúcense os filtros de correlación kernelizados, proporcionando un excelente equilibrio entre precisión e velocidade. Posteriormente, engádense características dun histograma de gradientes orientados (HOG) para mellorar a precisión, e as características multiescala permiten unha mellor estimación do tamaño da caixa delimitadora. O enfoque refínase aínda máis incorporando características de cores e plantexando a tarefa como un problema de clasificación de fondo ou primeiro plano, pasando de espazo lineal a non lineal grazas a KCF. Isto constitúe un avance significativo no seguimento de obxectos, xa que KCF facilita a diagonalización dos coeficientes do filtro, o que leva a optimizacións máis rápidas.

Para mellorar aínda máis a precisión dos métodos, os efectos de fronteira inducidos pola FFT diminúense integrando características multidimensionais e penalizando as mostras máis afastada do centro a través dun compoñente de regularización espacial. Ademais, para reducir o impacto das oclusións, DeconTrack introduce parámetros que pesan as mostras e regularización, mentres que LCT suxire un modelo de regresión discriminativo para avaliar a fiabilidade dos resultados do seguimento, evitando interferencias durante a etapa de actualización do filtro.

Non obstante, os ambientes complexos presentan serias dificultades para as funcións artesanais. En consecuencia, en DeepSRDCF, as características de entrada da SRDCF derivan dun extractor aprendido, mentres que HCFVT substitúe as sinais HOG con características aprendidas seguindo un enfoque paralelo. Do mesmo xeito, C-COT integra características convolucionais coa imaxe de entrada para producir un mapa de resposta altamente preciso. Este método é mellorado aínda máis en ECO para conseguir unha mellor precisión e eficiencia.

Seguindo esta tendencia, a exploración das características aprendidas en DCF persiste con métodos que asignan roles diferentes en función da súa profundidade, mellorando a regularización e con enfoques que ponderan de forma adaptativa a información do obxectivo. Non obstante, estes enfoques son cada vez máis complexos, perdendo a eficiencia que antes os definiu. Ademais, a súa incapacidade para permitir o seu adestramento íntegro dá como resultado que os modelos contemporáneos baseados na aprendizaxe profunda superen estes enfoques.

A idea central detrás da aprendizaxe profunda para o seguimento dun só obxecto xira en torno ao uso de redes de neuronas para extraer características da escena e clasificar as súas diferentes rexións como obxecto ou fondo. Un dos primeiros enfoques propostos é DLT, que

primeiro adestra a rede usando bases de datos de vídeo etiquetadas e despois axusta o modelo en liña usando o primeiro fotograma etiquetado. Aínda que a precisión que mostra é bastante boa, queda claro que os datos utilizados para adestrar a rede non son suficientes, xa que aínda non existen grandes bases de datos de vídeos etiquetados para o seguimento. Así, SO-DLT explora a idea de usar unha rede AlexNet adestrada previamente en ImageNet como a base do rastreador, o que produce resultados notablemente bos.

Con MDNet propónse un modelo de aprendizaxe multidominio, que emprega unha rede unificada con diferentes etapas finais adaptadas para cada vídeo do adestramento. Durante a inferencia, suprímense todas as capas específicas do dominio e constrúese unha nova capa única, que se axusta en liña durante o seguimento para adaptala ao novo vídeo. Este enfoque explórase aínda máis con esforzos para seleccionar as canles máis adecuadas para describir o obxecto e con melloras no adestramento. Non obstante, o proceso de axuste en liña produce un funcionamento máis lento do rastreador, o que fai que estes métodos sexan inadecuados para escenarios en tempo real.

A familia de rastreadores ATOM propón o uso de compoñentes específicos para a estimación e clasificación de obxectos co obxectivo de refinar de forma iterativa a caixa delimitadora. Estes rastreadores adáptanse en liña para centrarse no obxecto rastrexado, con enfoques que melloran o módulo de clasificación introducindo información distractora e inserindo unha subrede para a predición dunha boa inicialización, mellorando moito a precisión. Non obstante, estes métodos de estimación de obxectivos son moi custosos computacionalmente e implican axustar hiperparámetros altamente sensibles.

Coa aparición de grandes bases de datos de vídeo etiquetados, comezan a aparecer métodos como GOTURN. Estes descartan a idea de afinar o modelo en liña e confían unicamente no adestramento previo da rede de neuronas convolucional (CNN), logrando velocidades de máis de 100 fotogramas por segundo. Para mellorar a precisión do modelo, algunhas solucións crean un modelo de aparencia máis robusto ao conectar varias CNN previamente adestradas a través dunha estrutura en árbore, e outras aproveitan as vantaxes que presentan as redes de neuronas recorrentes (RNN) en tarefas secuenciais. Non obstante, estas técnicas foron substituídas en gran medida pola sinxeleza e eficacia dos algoritmos de aprendizaxe por semellanza.

En concreto, a introdución de redes siamesas con SiamFC supuxo un avance significativo no seguimento dun só obxecto. Este método emprega dous extractores de características idénticos baseados en AlexNet para procesar tanto a área de busca coma o aspecto inicial do obxecto. Posteriormente, estas características únense mediante unha operación de correlación cruzada, obtendo un mapa de puntuacións que representa as probabilidades da posición do obxectivo. A principal vantaxe deste enfoque reside na súa sinxeleza e precisión como solución adestrable íntegramente, que permite o seguimento en tempo real de calquera obxecto arbitrario sen necesidade de axustes ou adaptacións adicionais.

Fixéronse varias contribucións á arquitectura orixinal de SiamFC. Para identificar os cambios na relación de aspecto da caixa delimitadora, incorpórase un cabezal de regresión, tanto utilizando unha rede de proposta de rexións (RPN), un enfoque sen ancoraxes convencional ou un esquema de escalado orientado aos datos. Isto ofrece o beneficio adicional de eliminar a necesidade de probas multiescala para detectar variacións de tamaño dos obxectos. Ademais, a precisión da saída pódese mellorar incorporando un cabezal de segmentación que clasifica cada píxel dentro da área de busca.

Para permitir que estas arquitecturas utilicen extractores de características máis profundos, SiamRPN++ propón unha estratexia de adestramento novidosa xunto cunha técnica de fusión de características multicapa altamente eficiente. A precisión mellora aínda máis ao incorporar mecanismos de atención para enfatizar as características máis relevantes, empregando redes latentes para adquirir información específica do fotograma inicial e identificando distractores para minimizar as interferencias.

Para mellorar a robustez, empréganse funcións obxectivo alternativas durante o adestramento, tal e como suxire IFCSN. Ademais, en SiamFC++ introdúcense técnicas refinadas para a estimación do estado obxectivo e en STMTrack utilízanse redes de memoria espazo-temporal. Tamén se presta atención á selección dos hiperparámetros óptimos para o rastreador e á determinación das características do modelo máis adecuadas. En particular, en HP-Siam utilízase un enfoque de aprendizaxe por reforzo profundo para aprender os hiperparámetros do modelo, mentres que CFNet propón a actualización en liña das características do modelo.

Durante o último ano tamén xurdiron algunhas propostas que comezan a afastarse dos modelos siameses cara a arquitecturas puramente baseadas en Transformer. Estes modelos avanzados aproveitan Transformers de visión previamente adestrados, o que permite unha extracción máis eficaz de características do obxectivo. Inicialmente, estes enfoques empregaban un modelo de codificador-decodificador para manexar características concatenadas tanto para o modelo como para o área de busca. Pero están evolucionando gradualmente cara a metodoloxías de extracción e interacción de características conxuntas, así como a interacción de características xeneralizadas e o modelado de relacións cunha única rama.

Os modelos baseados en Transformers demostran un inmenso potencial e parecen estar a piques de establecer un novo precedente no campo do seguimento visual de obxectos. Non obstante, o importante custo de adestramento e os considerables requisitos de memoria e recursos computacionais presentan retos sen resolver, polo que actualmente son menos axeitados para escenarios nos que o tempo de execución é un factor crítico. Como resultado, as alternativas que usan eficientemente os recursos, como os modelos baseados en SiamFC, seguen sendo as opcións preferidas para as aplicacións de seguimento en tempo real.

Por outra banda, os sistemas de seguimento multiobxecto (MOT) preservan a identidade de varios obxectos presentes simultaneamente nunha escena. A diferenza do seguimento dun só

obxecto, non se proporcionan as localizacións iniciais destes elementos, o que require que o rastreador realice simultaneamente a detección e a asociación. Isto introduce unha complexidade adicional á tarefa, xa que o rastreador debe determinar cando crear novas traxectorias, xestionar os cambios de identidade e manexar obxectos que abandonan a escena para a súa eventual reidentificación.

Nos últimos anos, algunhas propostas tentaron abordar este desafío detectando e identificando obxectos simultaneamente aproveitando o poder de representación das CNN. Para aumentar a súa precisión, algúns enfoques tamén aproveitan a dimensión temporal para capturar mellor a dinámica dos obxectos procesando varios fotogramas á vez usando convolucións 3D. Para detectar e vincular caixas delimitadoras de forma máis eficaz, outros métodos aproveitan a arquitectura de Transformer e usan consultas que se propagan no tempo.

As vantaxes dos métodos de íntegros derivan da súa capacidade para funcionar sen necesidade de coñecementos heurísticos sobre o problema. Non obstante, estas técnicas dependen moito de grandes cantidades de datos etiquetados, que poden estar limitados en certos escenarios. En consecuencia, a conveniencia de adoptar un enfoque totalmente aprendido para o seguimento de varios obxectos aínda é un tema de debate, especialmente cando se poden asumir partes do problema. Esta é a razón pola que o seguimento por detección segue sendo o paradigma imperante no campo do seguimento multiobxecto na actualidade. Este enfoque permite dividir o desafío en dúas tarefas distintas: detección de obxectos e a asociación de deteccións.

Os detectores de obxectos resoven dúas tarefas principais: a localización de obxectos e a identificación das súas clases. Os enfoques tradicionais recorren a métodos e algoritmos de subtracción de fondo que aproveitan características deseñadas manualmente como os descritores de transformación de características invariantes a escala (SIFT) ou os histogramas de gradientes orientados (HOG). Inicialmente, a detección de obxectos realízase en todo o fotograma mediante fiestras deslizantes, pero esta técnica evoluciona posteriormente para incorporar propostas de rexións e así mellorar a eficiencia.

A chegada de AlexNet revoluciona o campo da clasificación, abrindo o camiño para adaptar as redes de neuronas convolucionais para resolver tarefas de detección de obxectos. Isto dá a luz a R-CNN, que aproveita unha AlexNet para clasificar as rexións suxeridas pola busca selectiva. Posteriormente, esta arquitectura evoluciona a Fast-RCNN, na se realiza a extracción de características sobre toda a imaxe para capturar máis información contextual, e Faster R-CNN, que incorpora unha rede de proposta de rexións (RPN) para seleccionar rexións de interese. Colectivamente, estes métodos serven como base para o que agora se coñece como detectores de obxectos de dúas etapas.

Non obstante, a proposta de YOLO presenta unha nova variedade de detectores de obxectos dunha etapa que non dependen das propostas de obxectos. En cambio, YOLO particiona a imaxe de entrada en rexións establecendo unha cuadrícula, ao tempo que predí caixas delimitadoras e

puntuacións de confianza para cada rexión. Posteriormente, SSD introduce un detector multinivel capaz de identificar pequenos obxectos en capas tempranas da rede debido á súa maior resolución, unha estratexia mellorada aínda máis en RetinaNet ao empregar unha función de erro mellorada e unha arquitectura piramidal de características que xera mapas de semánticamente mellorados a todas as escalas.

Coa introdución da arquitectura Transformer, xorden varios enfoques baseados en atención para a detección de obxectos. Algúns sistemas utilizan mecanismos de atención cruzada para substituír certos elementos dos detectores tradicionais de dúas etapas. Non obstante, os métodos máis efectivos ata agora implican empregar Transformers para modelar todo o problema. Estes detectores máis sofisticados ofrecen unha moi boa precisión e robustez nunha gran variedade de escenarios. Non obstante, a súa eficiencia vese comprometida, polo que os enfoques máis tradicionais son os que se empregan habitualmente nas aplicacións que requiren velocidade de execución. Aínda así, aínda que se empregue un método sinxelo, conseguir velocidades de inferencia en tempo real adoita ser moi complicado se hai que empregar o detector en cada fotograma.

Unha vez establecida unha técnica para obter deteccións en cada fotograma da secuencia, a seguinte tarefa consiste en vincular temporalmente estas deteccións e asignar identidades ás traxectorias ou *tracks*. Este proceso de asociación pódese xestionar fóra de liña, —é dicir, mediante o procesamento holístico de toda a secuencia tendo acceso a todos os seus frames—, ou en liña —é dicir, a información futura non está dispoñible, só o fotograma actual e os pasados pódense usar—, xa sexa utilizando información baseada no aspecto ou no movemento.

A posibilidade de acceder a todos os fotogramas dentro dunha secuencia proporciona unha clara vantaxe para as solucións fóra de liña, que normalmente presentan unha maior precisión en comparación cos enfoques en liña. Non obstante, a aplicabilidade dos métodos fóra de liña é algo restrinxida, xa que non son óptimas para aplicacións en tempo real ou escenarios que requiren o procesamento dun fluxo continuo de vídeo. O enfoque predominante para representar este tipo de problemas implica utilizar un modelo baseado en grafos, onde as deteccións constitúen os nós e as relacións potenciais están representadas polos vértices entre eles.

Os primeiros métodos aproveitan as técnicas de asociación conxunta de datos probabilísticos (JPDA) para resolver a tarefa. Despois disto, aparecen unha variedade de enfoques como as aproximacións baseadas en redes de fluxo, modelos de grafos probabilísticos, agrupación xerárquica e métodos que implican a partición do grafo. Algunhas destas tarefas pódense aprender parcialmente mediante modelos artesanais, pero só resolven asociacións por parellas.

Actualmente, os enfoques baseados na aprendizaxe profunda son os métodos máis frecuentes para resolver asociacións de seguimento multiobxecto fóra de liña. Estes métodos normalmente computan asociacións por parellas, empregando CNNs e integrando información a curto e longo prazo para afrontar situacións complexas. Ademais, os Transformers e as redes de neuronas de

grafos (GNN) utilízanse tamén con notable éxito. En particular, as GNN gañaron considerable atención debido ás súas capacidades inherentes para manexar problemas relacionados cos grafos e crear estruturas xerárquicas que abranguen secuencias enteiras. Non obstante, a pesar destes avances, tales enfoques seguen presentando limitacións na súa adaptabilidade para sistemas en liña, o que dificulta a súa aplicación en escenarios en tempo real.

Aproveitando o feito de que os desprazamentos de obxectos son xeralmente mínimos entre fotogramas adxacentes, os métodos en liña baseados en movemento empregan a información de localización das deteccións e traxectorias para a súa asociación. Algúns traballos realizan esta asociación aproveitando o propio detector, sen necesidade de aprender tarefas adicionais, xa sexa explotando as cabezas de regresión ou mediante algoritmos baseados en distancias para os centros de obxectos. O problema destes mecanismos é que non teñen unha verdadeira noción da identidade e son moi susceptibles a grandes desprazamentos e, polo tanto, teñen que ser complementados con modelos de movemento e mecanismos de reidentificación. Por iso é máis común modelar asociacións con métodos desacoplados do detector.

As técnicas de asociación baseadas no movemento poden ser tan sinxelas como calcular as asociacións usando a intersección sobre a unión (IoU) entre deteccións ou empregar algún modelo de movemento xunto coa superposición para facer que as traxectorias sexan máis robustas. Despois, as deteccións adoitan emparellarse coas traxectorias minimizando o custo global das asociacións para ese fotograma en particular, empregando técnicas coma o Algoritmo Húngaro.

Non obstante, existe o problema de determinar que deteccións se teñen en conta na asociación e cales se descartan. En lugar de simplemente establecer un limiar de confianza e ignorar as caixas delimitadoras de baixa confianza, ByteTrack propón un método que as considera todas en diferentes roldas de asociacións, logrando resultados que superan o estado do coñecemento. Non obstante, algúns obxectos aínda non se asocian correctamente, xa que os métodos puramente baseados en movemento teñen problemas en escenarios moi concorridos e vídeos grabados con cámara móbil.

Naqueles casos nos que é necesario asociar identidades en fotogramas non consecutivos separados por unha gran brecha temporal, as técnicas baseadas no movemento tenden a perder eficacia. Ademais, en ambientes desafiantes onde as predicións dos detectores poden ser inexactas ou totalmente ausentes, métodos coma o IoU fanse menos fiables. Nestas circunstancias, tórnase crucial depender de técnicas baseadas na aparencia para asociar adecuadamente as caixas delimitadoras. Para conseguilo, algúns enfoques constrúen unha matriz de aparencia que considera todas as posibles combinacións, adestran un modelo de regresión contraída por obxecto ou recorren a operadores de correlación para comparar propostas.

Non obstante, o enfoque máis común para a asociación baseada na aparencia implica a integración dunha rede de reidentificación e a creación de asociacións a partir das distancias entre características. Non obstante, hai situacións nas que a información visual non é tan fiable debido

a borrosidades ou oclusións completas, de aí que algúns traballos propoñan enfoques híbridos que integran simultaneamente aparencia e movemento. Isto mellora o seguimento, pero estas técnicas dependen de regras artesanais, que non permiten á rede aprender as relacións entre o movemento e a aparencia.

Tendo en conta isto, o presente estudo afonda na creación de enfoques innovadores para o seguimento de varios obxectos, que combinan características de movemento e aparencia para facilitar a aprendizaxe das súas relacións e mellorar a precisión e eficiencia dos métodos existentes. Para lograr estes obxectivos preséntase SiamMT, unha arquitectura innovadora que adapta técnicas de seguimento dun só obxecto para manexar múltiples obxectivos arbitrarios en tempo real. Este enfoque refínase aínda máis con SiamMOTION, que xestiona eficazmente os distractores e acomoda obxectos de diferentes tamaños extraendo características semánticamente máis ricas e propoñendo áreas de busca máis precisas. Por último, esta tese propón ByteFormer, unha arquitectura baseada en Transformer que complementa un detector lixeiro recuperando obxectos non detectados para mellorar o rendemento no seguimento de múltiples elementos. Así, estas aportacións cumpren os obxectivos definidos para a presente tese de doutoramento:

- O1. Seguimento multiobxecto en tempo real: o progreso actual no seguimento multiobxecto enfatiza principalmente a mellora da precisión, priorizándoa a miúdo sobre a velocidade, e propón solucións que dependen en gran medida dos detectores de obxectos, que poden dificultar o seguimento de obxectos independentes de clase. Este enfoque non se alíña cos casos de uso típicos, xa que hai unha demanda crecente de sistemas capaces de funcionar en tempo real ao mesmo tempo que rastrexan varios obxectivos arbitrarios. Para abordalo, é necesario deseñar algoritmos con capacidade para rastrexar varios obxectos simultáneos sen depender excesivamente das deteccións, tendo en conta tamén a súa escalabilidade para manter velocidades óptimas. Estes modelos deben executarse en tempo real mediante fontes de vídeo en directo, xestionando eficazmente as oclusións frecuentes, as flutuacións nas condicións de iluminación da escena e os casos nos que a aparición dos obxectos rastrexados sofre transformacións significativas.
- O2. Seguimento de obxectos de distinto tamaño: os rastreadores de obxectos visuais existentes están considerablemente influenciados polo tamaño do obxecto, o que lles leva a escalar a imaxe de entrada a unha dimensión fixa. Este enfoque funciona ben cando só hai un obxecto para seguir, pero faise problemático cando coexisten varios obxectos de diferentes tamaños na mesma escena, xa que é imposible escalar o fotograma para que todos os obxectos posúan un tamaño uniforme. Ademais, nestas circunstancias, as oclusións continuas e os distractores son habituais, o que fai que o seguimento sexa aínda máis difícil. Para afrontar este desafío, é imperativo desenvolver algoritmos que poidan rastrexar eficazmente varios obxectos pequenos e grandes e preservar a súa identidade. É necesario

facen especial fincapé na optimización da velocidade de inferencia e do consumo de recursos, permitindo un rendemento eficiente en dispositivos integrados para o procesamento en tempo real.

- O3. Recuperación de falsos negativos: hoxe en día, o paradigma imperante no seguimento multiobxecto é o seguimento por detección. Estas solucións presentan gran versatilidade, aínda que a súa eficacia depende en gran medida da precisión das deteccións, e mellorar a calidade dos detectores de obxectos existentes segue sendo un desafío complexo. Aínda así, empregar mecanismos baseados no movemento para asociar deteccións ofrece un enfoque sinxelo pero eficaz. Tendo en conta isto, exploramos a aplicación de modelos de movemento para xerar propostas adicionais en contornas onde as deteccións son escasas, co obxectivo de recuperar falsos negativos ignorados polo detector.

A consecución destes obxectivos durante o desenvolvemento da tese deu lugar ás seguintes publicacións científicas:

- Lorenzo Vaquero, Manuel Mucientes, Víctor M. Brea. *SiamMT: Real-Time Arbitrary Multi-Object Tracking*. Nas actas da 25th International Conference on Pattern Recognition (ICPR), pp. 707–714, Milán (Italia), 2020.

Indicadores de calidade: Congreso indexado por GII-GRIN-SCIE (Valoración GGS): A-, Clase 2.

- Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes. *Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning*. Pattern Recognition, Vol. 121, pp. 108205, 2022.

Indicios de calidade: Índice de impacto da revista: 8.518 (JCR 2021). Revista indexada no JCR 2021 en Ciencia da Computación, Intelixencia Artificial (Q1, 22/145).

- Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes. *Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks*. Nas actas da 26th International Conference on Pattern Recognition (ICPR), pp. 3485–3491, Montreal (Canadá), 2022.

Indicadores de calidade: Congreso indexado por GII-GRIN-SCIE (Valoración GGS): A-, Clase 2.

- Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes. *Real-Time Siamese Multiple Object Tracker With Enhanced Proposals*. Pattern Recognition, Vol. 135, pp. 109141, 2023

Indicadores de calidade: Índice de impacto da revista: 8.518 (JCR 2021). Revista indexada no JCR 2021 en Ciencia da Computación, Intelixencia Artificial (Q1, 22/145).

Contents

Abstract	1
1 Introduction	3
1.1 Motivation	3
1.2 Objectives	11
1.3 Methodology	12
1.4 Contributions	14
1.5 Dissertation Structure	16
2 Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning	19
2.1 Introduction	20
2.2 Related Work	23
2.3 SiamMT Architecture	24
2.4 Experimental Results	34
2.5 Conclusions and Future Work	40
3 Real-Time Siamese Multiple Object Tracker with Enhanced Proposals	43
3.1 Introduction	44
3.2 Related Work	46
3.3 SiamMOTION Architecture	48
3.4 Experimental Results	58
3.5 Conclusions and Future Work	64
4 Beyond Missed Detections: Leveraging Unmatched Trajectories for Enhanced Multi-Object Tracking	67
4.1 Introduction	68
4.2 Related Work	69
4.3 Methodology	72
4.4 Experimental Results	79
4.5 Conclusions and Future Work	83

5	Conclusions	85
	Bibliography	89
A	SiamMOTION Supplementary Material	107
B	Reproduction Rights	111
B.1	Journals	112
B.2	Conferences	114
	List of Figures	117
	List of Tables	119

Abstract

This thesis presents novel deep-learning approaches for tracking multiple simultaneous objects in videos, which is an essential component in several applications such as robotics or video surveillance. Traditional multi-object tracking systems, which rely on frame-by-frame detections and primarily geometric attributes, are ill-suited for real-time environments and open-set scenarios. To overcome these limitations, we introduce SiamMT, an innovative architecture that adapts single-object tracking techniques for handling multiple arbitrary targets in real time. This approach is further refined with SiamMOTION, which effectively manages distractors and accommodates objects of varying sizes by extracting semantically-rich features and proposing more accurate search areas. Lastly, this thesis proposes a Transformer-based architecture that complements a lightweight detector by recovering undetected objects for enhanced performance in multiple object tracking.

1 Introduction

1.1 MOTIVATION

The human ability to effortlessly recognize and perceive the elements within our visual environment stands in stark contrast to the complex computational challenges this task poses for artificial systems. The challenge encompasses not only understanding but also extracting significant information from the viewed scene. This sophisticated process, which remains difficult for machines to reproduce, has been the subject of extensive research within the field of computer vision over the past few decades.

The origins of computer vision can be traced back to the early 1960s [112, 10], when researchers begin exploring the possibility of teaching machines to “see” and interpret visual information. Early attempts focus on basic pattern recognition and edge detection in images [189], exploiting that information to identify simple geometric shapes and discern the structure of objects within a given scene. These preliminary endeavors provide valuable insights into the nature of visual data and pave the way to tackle increasingly intricate computer vision tasks. Thus, with the advances in computational power, imaging technologies, and the development of robust machine learning algorithms, the field gradually evolves towards more complex challenges like object detection [32], scene understanding [13], and semantic segmentation [140].

However, the information extracted from a single image can often be limited, providing only a partial understanding of the scene. Consequently, the field of computer vision starts to pay closer attention to video analysis in the early 1970s [52, 8], since it allows it to incorporate an additional dimension: time. This facilitates the creation of algorithms that leverage temporal information to enhance scene understanding and decision-making processes. Specifically, the task of *visual object tracking* arises, which consists of preserving the identity of one or more elements across video frames, as depicted in Figure 1.1. This not only enables the recognition of regions corresponding to distinct objects but also the analysis of trajectories, shape transformations, and behaviors throughout video sequences.

The fundamental task of tracking is now the cornerstone of numerous systems, including video surveillance devices [117], robotic systems [49], and autonomous vehicles [56]. There is a wide range of methods to tackle this challenge, each tailored to suit specific system requirements

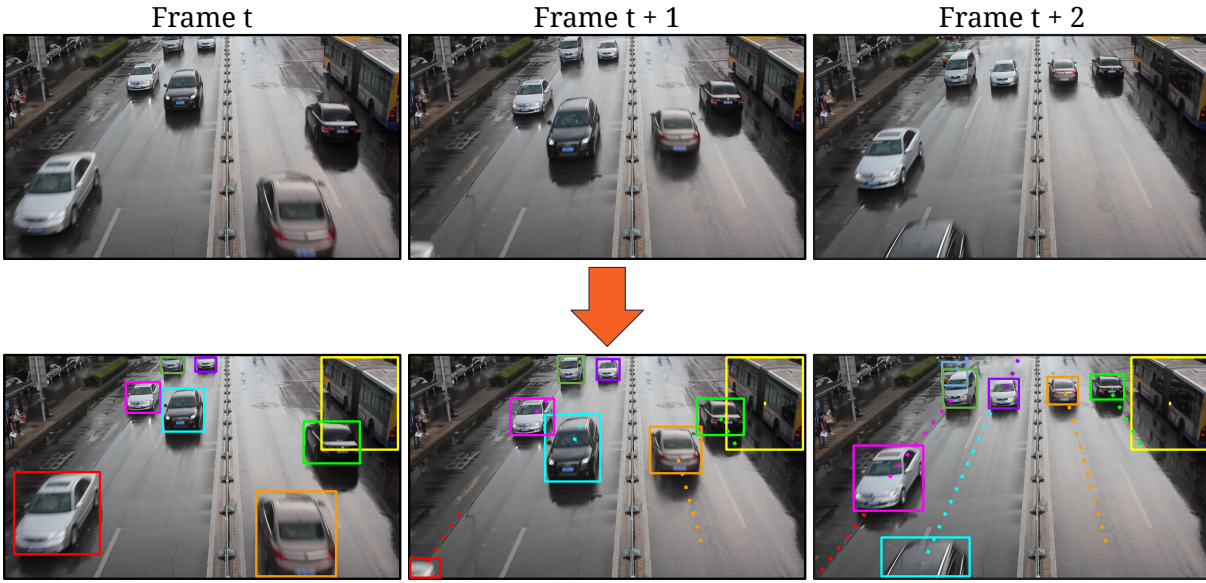


Figure 1.1: Output of a visual object tracking system after processing a video sequence. Objects are delimited by distinct colored bounding boxes, with each color representing a unique identity.

and application contexts. Within the literature, two primary approaches emerge: single-object tracking (SOT) and multi-object tracking (MOT), which have traditionally developed along separate lines with limited cross-disciplinary interaction.

The aim of this thesis dissertation is to bridge the gap between SOT and MOT and apply techniques from both single- and multi-object tracking domains in a unified manner to advance the state of the art in MOT. This research strives to create more efficient and accurate solutions that can be readily applied in a wide variety of real-world scenarios.

1.1.1 Single-Object Tracking

As the name suggests, single-object tracking (SOT) is primarily concerned with following a given element in a sequence. Typically, the initial position of the object is assumed to be known, and the objective of any SOT algorithm is to maintain the object's identity throughout the duration of the video sequence while providing accurate bounding box predictions for each frame. Such an object usually does not belong to a category known beforehand, so these algorithms must be able to handle arbitrary —i.e., class-agnostic— objects.

Given the nature of the applications where they are commonly employed, single-object trackers are typically online [177]. This means they generate predictions on a frame-by-frame basis without having access to future moments in the sequence. Alongside this, there is a strong emphasis on ensuring rapid inference speeds to accommodate the real-time performance demands prevalent in many contexts. Consequently, the majority of SOTs initially establish a model of the object's appearance, which is subsequently compared frame by frame to determine its precise

coordinates. To enhance efficiency, rather than searching the full frame, these trackers focus on specific regions with the highest likelihood of containing the target.

Early works in single-object tracking rely on classic computer vision techniques. Optical flow methods calculate the movement of the object by finding pixel displacements between frames [64]. The efforts towards modeling groups of pixels [98] and fast-moving objects [12] serve as the basis for other approaches such as the usage of Bayesian filter-based methods [102]—e.g., Kalman filters [68]. These predict the current status based on previous states and further refine those predictions using new observations. Particle filters [109] are a type of Bayesian filter that extend the capability of the previous methods by addressing scenarios that require non-Gaussian representations. Lastly, with the introduction of MeanShift [22], kernel-based methods emerge, which search for the region with the highest similarity based on color histograms.

The limitations of these classic approaches include their susceptibility to changes in lighting conditions, deformations, rapid motion, and occlusions. Nevertheless, they lay the groundwork for contemporary techniques, and many of their contributions remain highly pertinent. To this end, present-day single-object tracking strategies predominantly rely on discriminant correlation filters (DCF) and deep learning models.

Discriminant Correlation Filters in SOT. Early applications of correlation filters can be traced back to signal processing [17], where the objective is to compute the correlation between two signals. In the context of tracking, DCF-based trackers predict the position of the object by training a filter that distinguishes between the element of interest and the background of the scene. Early versions like MOSSE [11] focus on a single feature and represent an arbitrary target with a single filter. The proposals are very efficient, as correlations are converted into the frequency domain via Fast Fourier Transform (FFT) to speed up computations.

Based on this proposal, kernelized correlation filters are introduced in CSK [61], providing an outstanding balance between precision and speed. Subsequently, features from a histogram of oriented gradients (HOG) are added to improve accuracy [25], and multi-scale features allow for a better estimation of the bounding box size [85]. The approach is further refined by incorporating color features [31] and by posing the task as a foreground-background classification problem, transitioning from linear to non-linear space with KCF [62]. This constitutes a significant breakthrough in object tracking, as KCF facilitates the diagonalization of kernel coefficients, leading to faster optimizations.

To further enhance the accuracy of the methods, the boundary effects induced by the FFT are diminished by integrating multidimensional features [71] and by penalizing the samples with the furthest distance from the center through a spatial regularization component, as in SRDCF [71]. Additionally, in order to reduce the impact of occlusions, [29] introduces sample weight parameters and regularization, while [101] suggests a discriminative regression model for assessing the reliability of tracking results, preventing interferences during the filter update step.

However, complex environments pose serious difficulties for handcrafted features. Consequently, in [28], the features fed to SRDCF are derived from a learned convolutional backbone, while [100] substitutes HOG cues with learned features following a parallel approach. Similarly, C-COT [30] integrates convolutional features with the input image to produce a highly precise response map. This method is further enhanced in ECO [27] to achieve better accuracy and efficiency.

In line with this trend, the exploration of learned features in DCF persists with methods that assign varying roles based on their depth [7], improved regularisation [159], and approaches that adaptively weight the local target information [171]. However, these approaches are growing progressively more intricate, forfeiting the efficiency that once defined them. Moreover, their inability to undergo end-to-end training—which hinders optimal outcomes—results in contemporary deep learning-based models surpassing and overshadowing these approaches.

Deep Learning in SOT. The core idea behind deep learning for single-object tracking revolves around using neural networks to extract features from the scene and classify its different regions as the object or background. One of the first approaches to achieve this is DLT [150], which first trains the network offline using labeled video databases and then fine-tunes the model using the labeled first frame. Although the accuracy it shows is quite good, it becomes clear that the data used to train the network is not sufficient, since there are still no large databases of labeled videos for tracking. Thus, [149] explores the idea of using an AlexNet network [77] pre-trained on ImageNet [36] as the backbone of the tracker, which yields remarkably good results.

With MDNet [108], a multi-domain learning model is proposed, which employs a unified network with different final stages adapted for each training video. During inference, all domain-specific layers are suppressed and a new single layer is built, which is adjusted online during tracking to adapt it to the new video. This approach is further explored with efforts to select the most suitable channels to describe the object [148] and with improvements in training [97]. However, the online fine-tuning process results in slower tracker performance, making these methods unsuitable for real-time scenarios.

The ATOM [26] family of trackers proposes the usage of specific components for target estimation and classification with the aim of iteratively refining the bounding box. These trackers adapt themselves online to focus on the tracked object, with approaches that improve the classification module by introducing distractor information and inserting a subnetwork for the prediction of a good initialization [6], greatly improving the accuracy. Nevertheless, these target estimation methods are very computationally expensive and involve tuning highly sensitive hyperparameters.

With the emergence of large-scale labeled video databases, methods such as [60] start to surface. These discard the idea of fine-tuning the model online and instead rely solely on convolutional neural network (CNN) pre-training, achieving speeds of more than 100fps. In order

to improve the accuracy of the model, some solutions form a more robust appearance model by connecting multiple pre-trained CNNs through a tree structure [107], and others exploit the advantages that recurrent neural networks (RNN) exhibit in sequence-related tasks [23]. However, these techniques have been largely superseded by the simplicity and efficiency of similarity learning algorithms [137].

Specifically, the introduction of Siamese networks with SiamFC [4] marked a significant breakthrough in single-object tracking. This method employs two identical AlexNet-based backbones to extract features from both the search area and the template, i.e., the bounding box provided in the initial frame. Subsequently, these features are merged through a cross-correlation operation, resulting in a score map that represents the position probabilities of the target. The primary advantage of this approach lies in its simplicity and accuracy as an end-to-end trainable solution, enabling real-time tracking of any arbitrary object without the need for fine-tuning or additional adaptations.

Several contributions have been made to the original architecture of SiamFC. To identify changes in the bounding box's aspect ratio, a regression head is integrated, utilizing either a region proposal network (RPN) [83], a conventional anchor-free approach [55], or a data-driven target-aware scaling schema [20]. This offers the additional benefit of eliminating the need for multi-scale testing to detect object size variations. Furthermore, the precision of the output can be enhanced by incorporating a segmentation head that categorizes each pixel within the search area as pertaining to either the background or the object [167].

To enable these architectures to use deeper backbones and deal with the lack of strict translation invariance that padding introduces, [82] proposes a novel training strategy along with a highly efficient multi-layer feature fusion technique. The accuracy is further improved by incorporating attention mechanisms to emphasize the most relevant features [170], employing latent networks to acquire sequence-specific information from the initial frame [37], and identifying distractors to minimize interference [188].

In order to enhance robustness, alternative target functions are employed during training as suggested by [158]. Furthermore, refined guidelines for target state estimation are introduced in [161] and space-time memory networks are utilized as per [48]. Attention is also given to selecting the optimal hyperparameters for the tracker and determining the most suitable template features. In particular, a deep reinforcement learning approach is utilized in [38] for learning the model's hyperparameters, while [141] proposes the online update of the template features.

The last year has also seen the emergence of some proposals that begin to shift away from Siamese models towards purely Transformer-based architectures [146]. These advanced models leverage pre-trained vision Transformers, allowing for more effective target-specific feature extraction. Initially, these approaches employed an encoder-decoder model to handle concatenated features for both the template and search area [163, 86]. But they are gradually evolving towards

joint feature extraction and interaction methodologies [19, 165], as well as generalized feature interaction and relation modeling with a single branch [50].

Transformer-based models demonstrate immense potential and seem to be on the verge of setting a new precedent in the field of visual object tracking. However, the substantial training cost and considerable demands for memory and computational resources present unresolved challenges, currently making them less suitable for scenarios where runtime is a critical factor. As a result, resource-efficient alternatives such as SiamFC-based models are still the preferred options for real-time tracking applications.

1.1.2 Multi-Object Tracking

Multi-object tracking (MOT) systems preserve the identity of multiple objects concurrently present within a scene. Differing from single-object tracking, the initial locations of these objects are not provided, which requires the tracker to simultaneously perform detection and association. This introduces additional complexity to the task, as the tracker must determine when to give birth to new trajectories, manage identity switches, and handle objects that leave the scene for their eventual re-identification.

In the last few years, some proposals have tried to address the challenge by simultaneously detecting and identifying objects within a frame [155, 177] by leveraging the representational power of CNNs. To increase their accuracy, some approaches also exploit the temporal dimension to better capture object dynamics by processing multiple frames at once using 3D convolutions and then stitching the clips together [132, 1]. To detect and link bounding boxes more effectively, other methods leverage the Transformer [146] architecture and use special track queries that propagate through time [130, 103], with some approaches even eliminating the need for post-processing by spanning multiple frames at the same time [173].

The advantages of end-to-end methods stem from their ability to operate without the need for heuristic knowledge about the problem. However, these techniques are highly dependent on large quantities of labeled data, which may be limited in certain scenarios. Consequently, the appropriateness of adopting a fully-learned approach for multi-object tracking is still a matter of debate, especially when strong priors can be utilized. This is the reason why tracking by detection remains the prevailing paradigm in the field of multi-object tracking today [24]. This approach enables to divide the challenge into two distinct tasks: object detection and detection association.

Object Detection. Object detectors solve two primary tasks: object localization and identifying the corresponding classes. Traditional approaches resort to background subtraction methods [129] and algorithms leveraging manually engineered features like scale-invariant feature transform (SIFT) descriptors [96] or Histograms of Oriented Gradients (HOG) [25] for representing target categories. Initially, object detection is carried out throughout the whole frame

via sliding windows [147], but this technique later evolves to incorporate region proposals to enhance the efficiency [139].

The advent of AlexNet [77] revolutionizes the field of classification, paving the way for adapting convolutional neural networks to solve object detection tasks. This gives birth to R-CNN [54], which leverages an AlexNet for classifying the regions suggested by [139]. Subsequently, this framework evolves into Fast-RCNN [53], where feature extraction is performed on the entire image to capture more contextual information, and Faster R-CNN [121], which incorporates a region proposal network (RPN) for selecting regions of interest. Collectively, these methods serve as the foundation for what is now known as two-stage object detectors.

However, the proposal of YOLO [119] brings forth a new breed of one-stage object detectors that do not depend on object proposals. Instead, YOLO partitions the input image into regions by establishing a grid, while simultaneously predicting bounding boxes and confidence scores for each region. Following this, SSD [93] introduces a multi-level detector capable of identifying small objects in earlier layers of the network due to their higher resolution, a strategy further enhanced in RetinaNet [88] by employing an improved loss and a feature pyramid architecture [87] that generates high-resolution, semantically-rich feature maps across all scales. Furthermore, some other works also delve into execution efficiency, utilizing parameterized designs aimed at accommodating a diverse range of resource constraints [135].

With the introduction of the Transformer architecture [146], various attention-based approaches emerge for object detection. Some systems utilize cross-attention mechanisms to substitute certain elements of traditional two-stage detectors [65]. However, the most effective methods so far involve employing Transformers to model the entire problem, either by using features extracted with a CNN [16] or by feeding patches of the image directly to the attention mechanisms [39]. These more sophisticated detectors offer very good accuracy and robustness in a wide variety of scenarios. Nonetheless, their efficiency is compromised, so more traditional approaches are the ones commonly employed in time-critical applications. Still, even if a simple method is adopted, achieving real-time inference speeds tends to be very complicated if the detector has to be queried at every frame.

Upon establishing a technique for acquiring detections in each frame of the sequence, the subsequent aspect of the task involves linking these detections temporally and assigning identities to the respective tracks. This association process can be managed either offline, —i.e., through holistic processing of the entire sequence with access to all its frames—, or online —i.e., future information is unavailable, only the present frame and the past ones can be used— on a frame-by-frame basis, using appearance and/or motion-based cues.

Offline Association. The ability to access all frames within a sequence provides a clear advantage for offline solutions, which typically exhibit higher accuracy compared to online approaches. Nonetheless, the applicability of offline methods is somewhat restricted, as they are

not optimal for real-time applications or scenarios requiring the processing of a video stream. The prevalent approach to representing this type of problem involves utilizing a graph-based model, where detections constitute the nodes, and potential relationships are depicted by the edges between them.

Initially, methods leverage joint probabilistic data association (JPDA) techniques [47] to solve the task [122]. Following this, a variety of approaches surface, including those based on flow networks [2], probabilistic graph models [136], hierarchical clustering [179], and methods that involve graph partitioning using integer binary programming [79] or minimum cliques optimization [172]. Some of these tasks can be partially learned via conditional random fields [164] or handcrafted models [134], but they only solve pairwise associations.

Currently, deep learning-based approaches are the most prevalent methods for solving offline multi-object tracking associations. These methods typically involve pairwise appearance costs, employing CNNs [127] and integrating both long-term and short-term cues to tackle complex situations [45]. Additionally, Transformers [184] and graph neural networks (GNNs) [14] have been utilized with notable success. Particularly, GNNs have garnered considerable interest due to their inherent capabilities in handling graph-related problems and creating hierarchical graphs spanning entire sequences [18]. However, despite these advancements, such approaches continue to exhibit limitations in their adaptability to online systems, which hinders their application in real-time scenarios.

Online Motion-Based Association. Taking advantage of the fact that object displacements are generally minimal between adjacent frames, motion-based methods employ location information from detections and pre-existing tracks for online frame-by-frame association. Some works perform this association by taking advantage of the detector itself, with no need to learn additional tasks, either by exploiting the regression heads [44] or through distance-based algorithms for the object centers [182]. The issue with these mechanisms is that they do not have a true notion of identity and are very susceptible to large displacements and thus have to be complemented with motion models and re-identification mechanisms [3]. This is why it is most common to model associations with methods decoupled from the detector.

Motion-based matching techniques can be as simple as computing the associations using the intersection over union (IoU) between detections [9] or using some motion model along with the overlap to make the tracks more robust to temporal gaps [5]. Afterward, detections are usually matched to tracks by minimizing the overall cost of the associations for that particular frame, employing techniques such as the Hungarian algorithm [78].

However, there exists the issue of determining which detections are factored into the association and which ones are discarded. Instead of simply establishing a confidence threshold and ignoring the low-confidence bounding-boxes, [174] proposes a method that considers all of them in different rounds of associations, achieving solid state-of-the-art results with a comprehensive

model based on Kalman filters [68]. However, some objects are still not correctly matched as pure motion-based methods struggle in crowded and moving-camera scenarios.

Online Appearance-Based Association. In those cases where it is necessary to associate identities across non-consecutive frames separated by a large temporal gap, motion-based techniques tend to lose efficacy. Moreover, in challenging environments where detector outputs might be inaccurate or entirely absent, methods like IoU become less reliable. Under these circumstances, it becomes crucial to depend on appearance-based cues in order to appropriately link bounding boxes. To achieve this, some approaches build an appearance matrix that considers all possible matching combinations and processes it with a CNN to get the probabilities of each association [131], train a ridge regression model per target to differentiate its visual features [180], or resort to correlation operators to compare proposals [126], leveraging multi-task learning to improve the accuracy [166].

However, the most common approach for appearance-based association involves integrating a re-identification network and building associations from the inter-feature distances [156, 72]. Nonetheless, there are situations in which visual information is not so reliable due to blur or full occlusions, hence some works propose hybrid approaches that tackle this by simultaneously integrating appearance and motion [125]. This improves tracking, but these techniques rely on handcrafted rules, which does not allow the network to model relationships between motion and appearance.

In light of this, the present study delves into the creation of innovative approaches for multi-object tracking, which combine motion and appearance features to facilitate learning their relationships and improve the accuracy and efficiency of existing methods. To accomplish these objectives we first introduce SiamMT, an innovative architecture that adapts single-object tracking techniques for handling multiple arbitrary targets in real time. This approach is further refined with SiamMOTION, which effectively manages distractors and accommodates objects of varying sizes by extracting semantically-rich features and proposing more accurate search areas. Lastly, this thesis proposes ByteFormer, a Transformer-based architecture that complements a lightweight detector by recovering undetected objects for enhanced performance in MOT.

1.2 OBJECTIVES

The aim of this PhD dissertation is to explore the design and implementation of advanced deep learning algorithms and techniques for efficient and accurate object tracking within video sequences. Specifically, the objectives of this doctoral thesis are:

O1. Real-time multi-object tracking: Current progress in multi-object tracking primarily emphasizes enhancing the accuracy, often prioritizing it over speed, and proposes solutions that are heavily reliant on object detectors, which can hinder the tracking of class-

agnostic objects. This approach does not align with typical use cases, as there is a growing demand for systems capable of operating in real-time while concurrently tracking multiple arbitrary targets. To address this, it is necessary to design algorithms with the ability to track multiple simultaneous objects without excessively depending on detections while also taking into account their scalability for maintaining optimal speeds. These models should run in real-time using live video feeds, effectively handling frequent occlusions, fluctuations in scene illumination conditions, and instances where the appearances of tracked objects undergo significant transformations.

- O2. Multi-sized object tracking:** Existing visual object trackers are considerably influenced by object size, leading them to scale the input image to a fixed dimension. This approach works well when there is only one object to track, but it becomes problematic when multiple objects of varying sizes coexist in the same scene, as it is impossible to scale the frame so that all objects appear at a uniform size. Furthermore, in such environments, continuous occlusions and numerous distractors are commonplace, making tracking even more challenging. To tackle this challenge, it is imperative to develop algorithms that can effectively track multiple small-and-large objects in videos and preserve their identity. Particular emphasis will be placed on optimizing inference speed and resource consumption, enabling efficient performance on embedded devices for real-time processing.
- O3. False negative recovery:** Nowadays, the prevailing paradigm in multi-object tracking is tracking-by-detection. These solutions exhibit remarkable versatility and strength, yet their effectiveness is heavily reliant on the accuracy of detections, and improving the recall of existing object detectors remains a complex challenge. Still, employing motion-based mechanisms for associating frame-by-frame detections offers a simple-yet-effective approach. Considering this, we will explore the application of motion models for generating additional proposals within environments where detections are scarce, aiming to recover false negatives overlooked by the detector.

1.3 METHODOLOGY

In this thesis, we employ the standard principles of scientific research grounded in the scientific method to advance the current state-of-the-art in visual object tracking. Our approach involves an iterative process, where the obtained results guide the progression and identify upcoming challenges. This sequence of steps commences with a preliminary prototype mirroring established models, which is refined throughout each iteration to achieve the objectives of the study. It is crucial to emphasize that, owing to the exploratory character of the research, earlier iterations might be discarded as novel understandings of the problem emerge. The whole process encompasses five main phases:

Objectives definition: The initial stage, as specified in the previous section, involves examining the premises raised in this document, with special emphasis on the potential of deep learning-based approaches for multi-object tracking. Consequently, the goals of this study are established, followed by a thorough assessment of each objective to elucidate their specifics. Specifically, we defined the three objectives pinpointed in Section 1.2: real-time multi-object tracking, multi-sized object tracking, and false negative recovery.

Data gathering: The primary data sources leveraged for this study include publicly available datasets, such as YouTube-BoundingBoxes [118], the Generic Object Tracking Benchmark (GOT-10k) [66], MOTChallenge [104, 35, 34], UAVDT [169], VisDrone [186], Joint Track Auto (JTA) [42], or MOTSynth [41]. By employing these comprehensive datasets in conjunction, the developed methods can be thoroughly trained and evaluated, ensuring optimal performance in multi-object tracking scenarios.

Survey of related work: A comprehensive examination of the current state of the art will be conducted to assess the various approaches proposed to date. This analysis conducted in Chapter 1 will facilitate an in-depth understanding of the research domains relevant to this thesis and enable the evaluation of existing methodologies, their drawbacks, and their advantages. Furthermore, it will help identify alternative techniques that could be beneficial in accomplishing the intended goals.

Design and implementation of proposals: Upon establishing the goals and examining the existing state-of-the-art, architectures designed for efficient and scalable tracking will be proposed, accompanied by the introduction of novel operators. The detailed descriptions of the designs and technologies considered at each stage can be found in Chapters 2, 3, and 4.

Validation: In the final stage, following the implementation of the proposed approaches, a set of evaluations will be conducted to verify the effectiveness of the developed models and the outcomes achieved. This assessment will encompass both precision and accuracy, as well as computational efficiency and scalability of the devised architecture. Should the performance prove inadequate, an iterative process involving adjustments to the designed algorithms and subsequent validation will commence until the results meet the desired standards. In Chapters 2, 3, and 4, we present the respective experimental results for each proposed method, employing standard evaluation metrics across an extensive range of benchmarks, which include MOT-17 [34], MOT-20 [35], UAVDT [169], VisDrone [186], and JTA [42]. The main conclusions derived from the research carried out in this thesis are compiled in Chapter 5.

1.4 CONTRIBUTIONS

The research conducted in this PhD thesis has resulted in the following contributions:

C1 SiamMT: A novel class-agnostic multi-object tracker capable of operating in real-time for several dozens of simultaneous targets while eliminating the need for frame-by-frame detections. It comprises:

C1.1 Leverage of visual single-object tracking techniques for their extension to multi-object tracking in an efficient manner. The novel proposal allows its application to SiamFC-based architectures, either by training the networks end-to-end or by maintaining their learned weights via network-based transfer learning.

C1.2 Region of interest (RoI) operator compatible with backbones that do not utilize padding. It reduces the number of sampling points per bin for increased efficiency and recomputes region coordinates to allow for cropping and resizing features obtained from fully-convolutional backbones lacking padding. This enables the tracking of objects with varying sizes within the same frame.

C1.3 Pairwise cross-correlation operator for identity matching. It leverages the properties of depthwise cross-correlation, facilitating the efficient pairwise comparison of dozens of feature maps in real time.

C2 SiamMOTION: A new real-time visual multi-object tracker capable of effectively handling crowded scenes with numerous distractors while simultaneously tracking objects of diverse sizes. The components of this system include:

C2.1 Proposal engine incorporating inertia information and multi-resolution features. This engine utilizes a feature pyramid network (FPN) for feature extraction, enabling the acquisition of significant features for objects of varying sizes. Moreover, search areas are derived from these features by employing an inertia module that considers the past positions of each object.

C2.2 Lightweight attention mechanism that enhances the features most closely related to the objects of interest. In conjunction with a novel pairwise-depthwise region proposal network (PD-RPN), this approach enables real-time prediction of precise bounding boxes for multi-object tracking.

C2.3 Multi-object penalization module designed to suppress distractors and outliers by considering all objects of interest within the scene. This module captures interactions among targets and implements four distinct forms of penalization, each tailored to address a specific type of tracking error.

C3 ByteFormer: A novel multi-object tracking method that focuses on tracks without matching detections and relies on an attention-based architecture that leverages both visual and spatial cues. It encompasses:

- C3.1** Generalization of a standard association mechanism to enable the recovery of tracks even in the absence of available detections. This proposal is able to recover a significant number of objects that were overlooked by the detector, reducing the number of false negatives.
- C3.2** Transformer-based network to associate motion-based candidates to tracks. The novel architecture harnesses multiple-choice question-answering techniques and contextual information to correctly match tracks and detection-free proposals.
- C3.3** Spatiotemporal encoding comprising position, size, and time. This module allows ByteFormer to process multimodal inputs and learn relationships between appearance and motion.

Publications

The following publications constitute the scientific production generated from the research conducted within this thesis.

Journal Publications

- **Lorenzo Vaquero**, Víctor M. Brea, Manuel Mucientes. *Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning*. Pattern Recognition, Volume 121, pp. 108205, 2022.

Quality indicators: Journal Impact Factor: 8.518 (JCR 2021). Journal ranked in JCR 2021, in Computer Science, Artificial Intelligence (Q1, 22/145).

Thesis contribution: C1.

- **Lorenzo Vaquero**, Víctor M. Brea, Manuel Mucientes. *Real-Time Siamese Multiple Object Tracker With Enhanced Proposals*. Pattern Recognition, Volume 135, pp. 109141, 2023.

Quality indicators: Journal Impact Factor: 8.518 (JCR 2021). Journal ranked in JCR 2021, in Computer Science, Artificial Intelligence (Q1, 22/145).

Thesis contribution: C2.

Conferences

- **Lorenzo Vaquero**, Manuel Mucientes, Víctor M. Brea. *SiamMT: Real-Time Arbitrary Multi-Object Tracking*. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), pp. 707–714, Milano (Italy), 2020.
Quality indicators: Conference Ranking (GGS Rating): A-, Class 2.
Thesis contribution: C1.
- **Lorenzo Vaquero**, Víctor M. Brea, Manuel Mucientes. *Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks*. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), pp. 3485–3491, Montréal (Canada), 2022.
Quality indicators: Conference Ranking (GGS Rating): A-, Class 2.
Thesis contribution: C2.

Other Contributions

List of publications not directly related to the main contributions of this thesis.

- S. Nazir, **Lorenzo Vaquero**, M. Mucientes, V. M. Brea, D. Coltuc. *2HDED:Net for Joint Depth Estimation and Image Deblurring from a Single Out-of-Focus Image*. In Proceedings of the 29th IEEE International Conference on Image Processing (ICIP), pp. 2006–2010, Bordeaux (France), 2022.
Quality indicators: Conference Ranking (GGS Rating): A-, Class 2.
- S. Nazir, **Lorenzo Vaquero**, M. Mucientes, V. M. Brea, D. Coltuc. *Depth Estimation and Image Restoration by Deep Learning from Defocused Images*. **Accepted** at IEEE Transactions on Computational Imaging, 2023.
Quality indicators: Journal Impact Factor: 4.708 (JCR 2021). Journal ranked in JCR 2021, in Engineering, Electrical and Electronic (Q2, 73/276).

1.5 DISSERTATION STRUCTURE

This PhD dissertation comprises five chapters and two supplementary sections. The manuscript first delves into the contemporary state of the art in object tracking, highlighting existing research challenges and delineating the objectives of this thesis. Subsequently, various innovative approaches are proposed, with a thorough examination of their results and contributions to the field. Ultimately, the insights drawn from this research are discussed in greater depth. In particular, the document is structured as follows:

Chapter 1: The present chapter. It provides an introduction to visual object tracking. It offers a comprehensive review of the state of the art in both single-object and multi-object tracking, along with an analysis of their potential limitations. Additionally, the scientific challenges and objectives addressed in this PhD dissertation are thoroughly detailed.

Chapter 2: Describes SiamMT, a new deep learning multiple visual object tracking solution that applies single-object tracking principles to multiple arbitrary objects in real-time. To achieve this, it reuses feature computations, implements a novel crop-and-resize operator, and defines a new and efficient pairwise similarity operator. SiamMT naturally scales up to several dozens of targets, reaching 25 fps with 122 simultaneous objects for VGA videos, or up to 100 simultaneous objects in HD720 video. This chapter also explores the performance of SiamMT compared to other state-of-the-art trackers, providing an extensive analysis on five large datasets that demonstrates the superiority of the proposal. This portion of the manuscript focuses on tackling Objective O1 within the context of the PhD research.

Chapter 3: Proposes SiamMOTION, an innovative motion estimation approach that builds on the core features introduced by SiamMT and presents an improved architecture. It includes a novel proposal engine that produces quality features through an attention mechanism and a region-of-interest extractor fed by an inertia module and powered by a feature pyramid network. The architecture also comprises a comparison head that efficiently matches pairs of exemplars and search areas, generating high-quality predictions via a pairwise depthwise region proposal network and a multi-object penalization module. The chapter validates SiamMOTION on five public benchmarks, achieving leading performance against current state-of-the-art trackers. This section addresses Objective O2 of the doctoral dissertation.

Chapter 4: Describes a technique that complements a conventional association mechanism, enabling the handling of tracks without matching detections. In order to solve associations between tracks and motion-based candidates it relies on ByteFormer, a novel Transformer-based architecture that leverages multiple-choice question-answering techniques. The method exploits contextual information to increase its robustness and introduces a novel spatiotemporal encoding that allows ByteFormer to process multimodal inputs —i.e., appearance, size, distance, and time— and learn relationships between them. The chapter also asserts the performance of ByteFormer on the standard MOTChallenge benchmark, where it achieves state-of-the-art results for online multi-object tracking. This part of the manuscript explores Objective O3 of the PhD.

Chapter 5: Presents a summary of the key conclusions drawn from the research conducted in this thesis. Additionally, this chapter outlines potential avenues for future research building upon these insights.

Appendix A: Supplementary material for Chapter 3. Provides additional insights into the qualitative outcomes and training process of SiamMOTION, which was described in Chapter 3.

Appendix B: Encompasses details on licensing and reproduction rights associated with the body of work published during the course of this PhD research. The content of this doctoral thesis is partially founded upon these publications.

2 Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning

In this chapter, we present an advanced framework that employs state-of-the-art visual object-tracking methodologies to effectively track a large number of targets without the necessity for detections in every frame. This is particularly advantageous in video analytics applications with real-time demands, where executing a detector on each frame might be computationally prohibitive. To this end, we introduce SiamMT, a groundbreaking deep-learning-based multi-object tracking system that extends the principles of single-object tracking to simultaneously handle multiple arbitrary —i.e., class-agnostic— objects in real-time.

In order to achieve real-time performance, SiamMT reuses feature computations across various objects, introduces a novel crop-and-resize operator for efficient processing, and establishes a new and computationally efficient pairwise similarity operator. By employing these innovative techniques, SiamMT is capable of naturally scaling to track several dozens of targets in parallel. To demonstrate the effectiveness of SiamMT, we have conducted extensive evaluations on five large real-time benchmarks, where it has shown leading performance in comparison to current state-of-the-art trackers. This makes SiamMT a compelling solution for a wide range of applications, including surveillance, sports analytics, autonomous vehicles, and robotics.

The remainder of the chapter is organized as follows. Section 2.1 delves into the contributions made by SiamMT and the various applications that stand to benefit from its implementation. Section 2.2 provides an overview of the contemporary state-of-the-art and highlights its limitations. Section 2.3 details the architecture and components of SiamMT. Section 2.4 presents the experimental results. Lastly, Section 2.5 offers concluding remarks on the proposal and outlines potential avenues for future research.

The contents of this chapter were extracted from the following publications:

Lorenzo Vaquero^a, Manuel Mucientes^a, Víctor M. Brea^a. *SiamMT: Real-Time Arbitrary Multi-Object Tracking*. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), pp. 707–714, Milano (Italy), 2020. ISSN: 1051-4651. DOI: 10.1109/ICPR48806.2021.9412625.

^aCentro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning*. Pattern Recognition, Volume 121, pp. 108205, 2022. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2021.108205.

^aCentro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.

2.1 INTRODUCTION

Video analytics systems carry out functions such as automatic video summarization [178] or path planning in autonomous vehicles [116], to name a few. Many of these systems follow the multi-object tracking (MOT) framework, which processes a video sequence and generates the track —set of bounding-boxes— for each object appearing in the scene [81]. They perform this task using the different components shown in Figure 2.1. Thus, for each new frame, a pre-trained detector is run to locate the categories of interest in the scene. Then, these detections are compared with the objects identified in the previous frames, generating an affinity score, to finally make an association in which they are matched, looking for the lowest overall cost.

However, when real-time is a must for video analytics, typical MOT frameworks are not suitable due to their high computational cost. For example, in MOT2020 [35] there are only 5 approaches that report real-time speeds, but they do not take into account the running time of the detector. Considering also the detector times, all those approaches are ruled out¹. Typical runtimes for accurate detectors on an NVIDIA TITAN V for an HD720 image are 23 fps for EfficientDet-D3 [135], or 16 fps for RetinaNet [89]. Certainly, there are detectors that can run in real-time, even on embedded GPU systems, like the lightweight Tiny-YOLOv3 [46], but they often have a poor accuracy or resort to shrinking the input image. Therefore, the most extended approach to enable real-time processing is to call the detector at a lower frame rate and perform motion estimation between detector calls, allowing the system to provide the position of the objects in all the frames. Thus, the motion estimation module feeds the affinity block when the detector is not called [46] (Figure 2.1, red path).

¹We consider that a system/module operates in real-time if it runs at least at 25 fps for HD720 resolution on an NVIDIA TITAN V or similar.

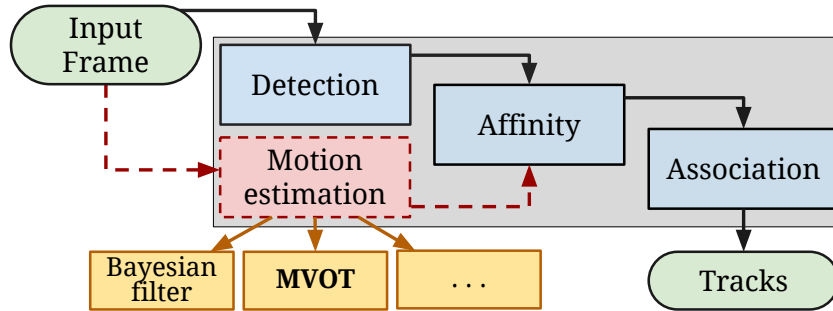


Figure 2.1: Diagram of a typical MOT system. They have three well-defined steps (colored blue) for generating tracks from input frames. However, real-time systems cannot have detections in all frames (dashed path), so most of the time they rely on a motion estimation module (red path). The purpose of SiamMT is to perform this motion estimation using visual object tracking techniques.

There are different approaches to address motion estimation, being Bayesian filters and Visual Object Trackers (VOT) the most widely used [21]. Visual object trackers are usually class-agnostic, so they are able to track any object regardless of their category, without requiring specific retraining or knowing the class to which the object belongs [4], which results in much more general motion models than those detection-inspired [183]. Yet, the problem with motion models based on multiple visual object tracking (MVOT) is that they work by instantiating multiple single-object trackers [166], which is only feasible—in terms of computational time—when there are few targets in the scene.

In order to address motion estimation, we propose SiamMT, an MVOT proposal capable of applying single-object online tracking techniques to multiple simultaneous targets in real-time. SiamMT is based on SiamFC [4]. Our approach solves different challenges (Figure 2.2) to create a scalable solution, allowing the sharing of features between objects, adding a feature crop-and-resize module and including a novel pairwise similarity operation. To the best of our knowledge, SiamMT is the first deep-learning-based real-time arbitrary MVOT. The main novelties of our proposal are summarized as follows:

- We propose SiamMT, a Siamese convolutional neural network capable of real-time tracking multiple arbitrary objects in a scalable manner. Its design allows its application in SiamFC-based architectures, either by training it end-to-end or by maintaining their learned weights via a network-based transfer learning procedure.
- In order to allow the tracking of different sized objects in the same frame, we establish a reformulation of the RoIAlign operator [58], making it capable of cropping and resizing features extracted with a fully-convolutional backbone without padding.
- We define a new similarity operator which, based on the properties of depthwise cross-correlation, enables the efficient pairwise comparison of multiple feature maps.
- We perform a deep analysis on the speed and performance benefits of each operator in the architecture.
- Our approach is able to track 122 simultaneous objects in VGA video and 100 objects in HD720 video at 25 fps, all with robustness and accuracy that exceed the current state-of-the-art.

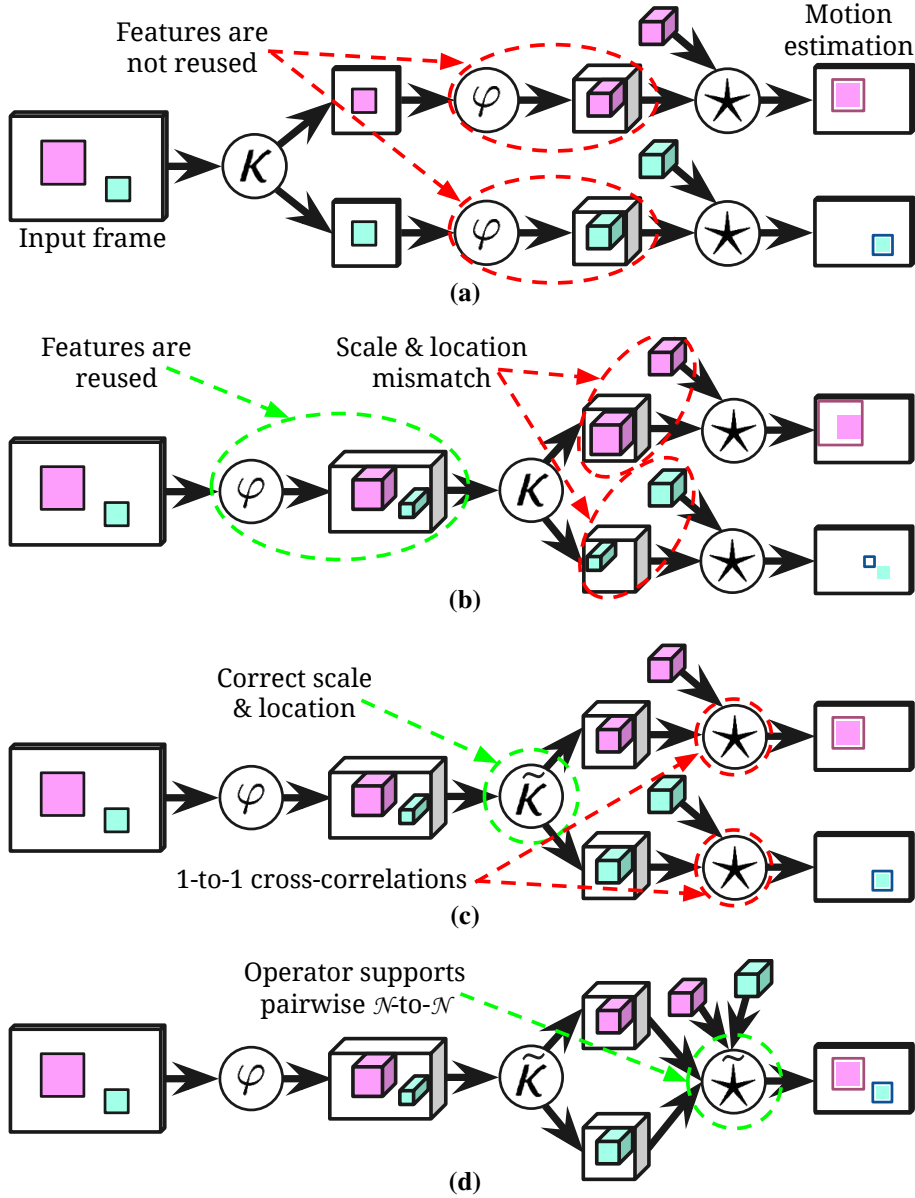


Figure 2.2: Different architectures that show the evolution from SiamFC to SiamMT: (a) instantiates multiple SiamFC trackers; (b) reuses features computations with a single backbone φ ; (c) introduces $\tilde{K}$ to solve the location and scaling problems; and (d) adds $\tilde{\star}$ to support multiple comparisons. Bringing all these three elements together gives rise to SiamMT.

An initial iteration of this framework was described in [144]. The key advancements presented in this work, as compared to the earlier proposal, include:

- The foundations of design decisions are further deepened, extending the motives behind them and why other possible approaches are not appropriate. Along with this, the introduced operators are defined and analysed in more detail, exploring their effects and performance in greater depth.
- Experimental validation is improved, focusing on real-time performance and adding more benchmarks with greater diversity and challenges.
- Training is improved, obtaining better accuracy and robustness metrics.
- The network architecture is further optimized, achieving a $1.69\times$ speedup.

2.2 RELATED WORK

Motion Estimation. The main contribution of this paper lies in the development of a multiple visual object tracker (MVOT) for motion estimation. There are approaches that allow maintaining the identity of multiple objects by associating their detections across frames [80]. However, since they require detections in all the frames, they cannot be considered for real-time systems. Some of them report near-real-time speeds, but they are achieved without taking into account the detector time or by using multiple GPUs in parallel, so we cannot consider them truly viable solutions.

The preferred solutions are those that are completely decoupled from the detector. Within this category, one of the first approaches consisted in the use of Kalman filters to estimate a linear constant velocity model [5]. However, with the rise of visual deep learning models, these Bayesian-based predictors were soon displaced by MVOT methods. Thus, systems emerge that base their motion estimation on single-object trackers, either by including them as an independent stage [185] or by unifying the object motion and affinity model into a single architecture [166]. But the drawback they all have so far is that they simply instantiate multiple single-object trackers, so speeds are low, especially when there are multiple targets in the scene—for example, [166] runs at 5 fps for 21 objects on FHD video.

Visual Object Tracking. Single-object visual object trackers receive the location and the dimensions of the target for the first frame of the sequence. From this point on, they search for the object of interest in each frame of the video, updating its coordinates and size. Specifically, Discriminative Correlation Filters (DCF) trackers predict the position of the object by training a filter that distinguishes between the element of interest and the background of the scene [159], with some works modeling them as convolutional layers [171] to adaptively weight the local target information. However, while these techniques provide great speeds, they have been somewhat displaced by more accurate deep learning approaches.

Deep learning-based trackers employ deep convolutional neural networks (CNNs) to train a similarity function which, starting from the initial appearance of the object, indicates its position in each new frame. The ATOM [26] family of trackers is one example of this, using specific components for target estimation and classification with the aim of iteratively refining the bounding-box. These trackers adapt themselves online to focus on the tracked object, with approaches that improve the classification module by introducing distractor information and inserting a subnetwork for the prediction of a good initialization [6], greatly improving the accuracy. Nevertheless, these target estimation methods are very computationally expensive and involve tuning highly-sensitive hyperparameters.

Siamese Object Trackers. The most widely adopted trend for object tracking involves the use of a Siamese structure to compare the initial appearance of the object with each frame's search area, with SiamFC [4] being the forerunner of the state-of-the-art. Several contributions

Tracker	Real-time	Deter.	Backbone	Similarity
SiamFC [4]	✓	✓	AlexNet	XCorr
SiamRPN [83]	✓	✓	AlexNet	RPN
SiamRPN++ [82]	✓	✓	ResNet-50	Depthwise-RPN
ATOM [26]	✓		ResNet-18	
DiMP [6]	✓		ResNet-50	
AGUnet [167]		✓	Custom	Mask
SiamCAR [55]	✓	✓	ResNet-50	Cls+Reg+Cen
SiamFC++ [161]	✓	✓	GoogleNet	Cls+Reg+Qua
ASCT [171]			VGG16	
TIFC [158]	✓		VGG16	XCorr
SiamMT	✓	✓	AlexNet	Pairwise-XCorr

Table 2.1: State-of-the-art approaches for MVOT.

have been made to the original architecture, allowing the regression of the object’s bounding-box—either with [83] or without anchors [55]—, incorporating more powerful backbones [82], including segmentation information [167], adopting new objective functions during training [158], or defining new guidelines for target state estimation [161]. However, most of these new approaches tend to add considerable complexity or significantly decrease the tracking speed, so they are less suitable for extension towards MVOT systems.

Table 2.1 shows a brief summary of the approaches discussed above with their main features. “Deter.” denotes whether the approach is deterministic and “Similarity” specifies the similarity operation of the algorithm (“XCorr”. for cross-correlation, “RPN” for Region Proposal Network, “Cls” for class, “Reg” for regression, “Cen” for center-ness, “Qua” for quality assessment, and empty if the network is non-Siamese). Non-deterministic models lean towards online learning, so they are often less popular due to their higher overhead. The lightest and simplest models are currently the most widely used alternatives for MVOT. This is why we have developed our multi-object tracking approach from SiamFC [4], because it is fast, simple, effective, and is the basis of the current state-of-the-art in tracking. Therefore, present and future Siamese correlation architectures could follow our proposal.

2.3 SIAMMT ARCHITECTURE

SiamMT is built around SiamFC [4], redesigning its architecture to allow the efficient tracking of multiple simultaneous objects (Figure 2.3), all while following the techniques employed by individual object trackers. Hence, we will first make a brief description of the foundations of SiamFC and then describe how SiamMT applies and adapts them to multiple targets.

2.3.1 SiamFC’s Network Architecture

SiamFC is the forerunner of the current state of the art in single object tracking. Its architecture, described in Figure 2.4(a), uses deep-learning similarity metrics to track individual objects at a high number of frames per second. The tracking process starts with the initial location of the

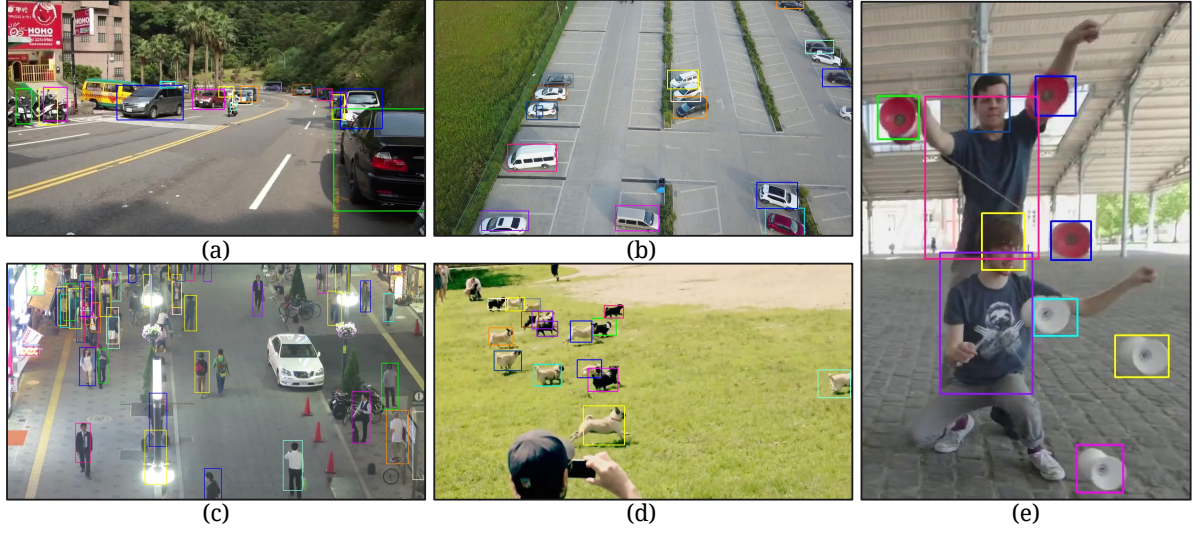


Figure 2.3: Qualitative examples of SiamMT on different sequences from (a) ILSVRC [123], (b) VisDrone [186], (c) MOT-2017 [104], and (d), (e) YT-BB [118].

object at Frame 0. Based on this information, this first frame is cropped and resized, obtaining the exemplar image of the tracked object. This exemplar image consists of the bounding box containing the object plus a context margin ζ , all scaled to a size of 127×127 pixels. Following this, the features of this image are extracted using an AlexNet-based [77] fully-convolutional neural network without padding—as no padding is applied, the strict translation invariance property of the convolutions is maintained [82]. The resulting 6×6 pixels and 256 channels features tensor defines the appearance of the object, and will be reused throughout the whole tracking process.

Once obtained the exemplar features, the tracking process itself begins, and is repeated for each frame in the sequence. The new frame is loaded and a search area is defined around the last known position of the object. This region is then cropped and resized with κ , obtaining the search area image— 255×255 pixels—, which is processed using φ to obtain a features tensor of 22×22 pixels and 256 channels. After this, the previously extracted exemplar features are compared with each one of the regions of the search area by means of a cross-correlation operation $\star$, generating a score map of 17×17 elements. Lastly, this map is bicubically upsampled to 272×272 pixels and large displacements are penalized to finally apply non-maximum-suppression and obtain the new position of the object.

This process has a number of important insights. First, if we adhere to the above description, the process would only recognize displacements, not changes in scale. In order to detect scale changes, SiamFC considers two additional search areas that cover regions of different sizes (slightly smaller and slightly larger). Once the similarity between the exemplar features and the 3 search areas is computed, the score map with the highest probability is chosen as the new size of the object.

Second, due to the way the exemplar image and the search area images are defined, they always represent any object with the same proportion. This is an essential factor in this type

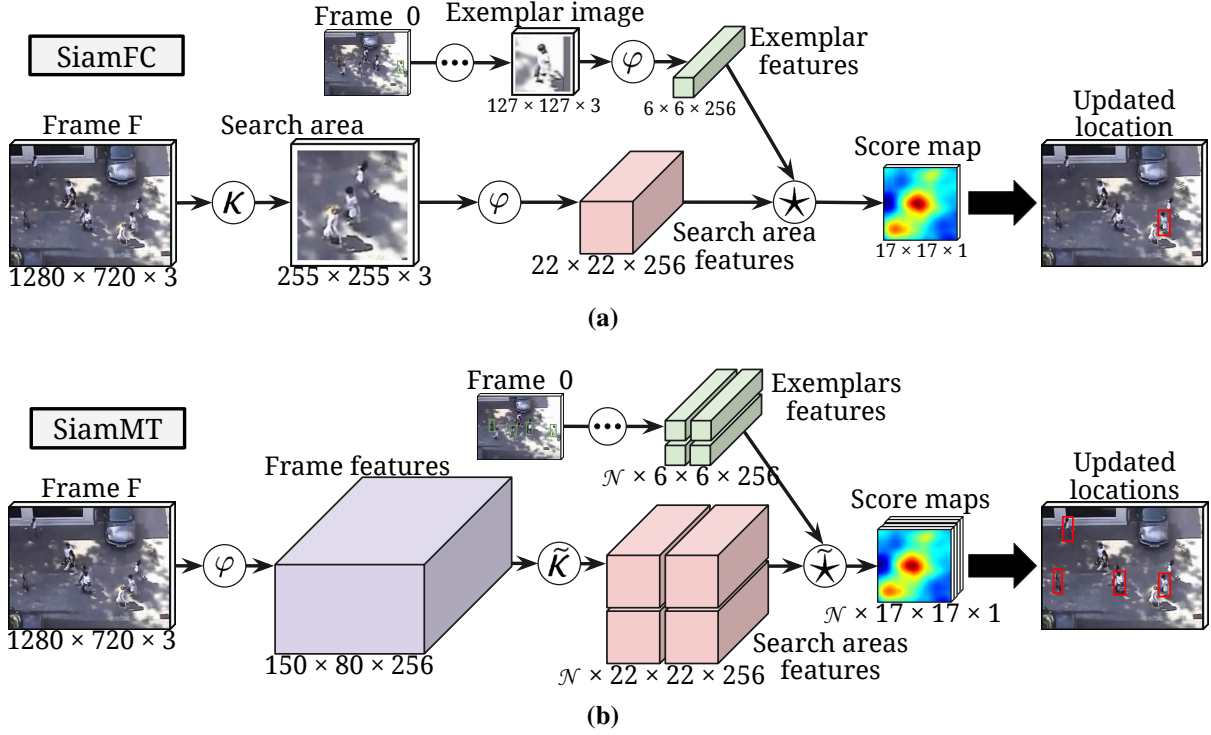


Figure 2.4: (a) SiamFC and (b) SiamMT network architectures during the inference phase. SiamMT first extracts the features of the entire frame via a φ backbone, enabling the reuse of features. After this, the features of the various search areas are cropped and resized with the $\tilde{\kappa}$ operator. Finally, these features are combined with those of their respective exemplars through $\star$, obtaining score maps that indicate the new positions of the $\mathcal{N}$ objects.

of architectures, since it allows to: (i) learn a feature extractor that will always consider similar sized objects; (ii) and define a similarity operation that can assume that the size of the object in exemplar features and in search area features is approximately similar. As a result, the network gains a lot of precision and is able to maintain the object’s scale between frames.

2.3.2 Modifying the SiamFC Architecture to Multiple Objects

The architecture of SiamMT emerges from modifying the architecture proposed by SiamFC as shown in Figure 2.4(b), allowing the efficient tracking of multiple simultaneous objects. The main features of SiamMT and its differences from SiamFC are described below.

Global Features Extraction. In the pipeline of SiamFC and its subsequent evolutions, for each new frame, the first step consists of cropping and resizing (κ) the object’s search area to then extract its features (φ). For a frame with $\mathcal{N}$ objects this would be inefficient. Therefore, the solution proposed by SiamMT consists in removing the image crop-and-resize module κ and applying the φ operator directly on the input frame (Figure 2.4(b)). This allows the reuse of features when there are multiple objects on the scene, which enables the scalability of the system since φ is the most expensive operation in the architecture and its execution per search area—as in SiamFC—would be unfeasible.

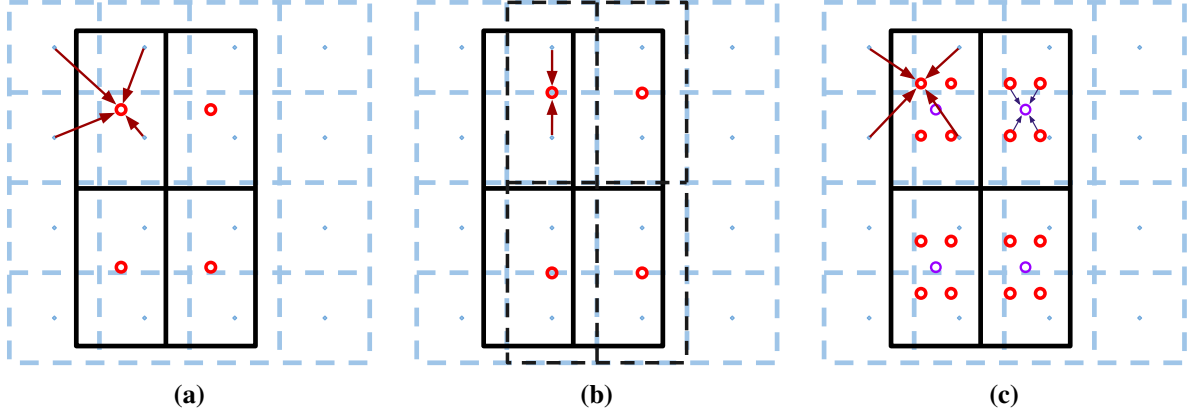


Figure 2.5: Pooled pixels when applying (a) $\tilde{\kappa}$, (b) RoIPool, and (c) RoIAlign over a 4×4 features tensor (depicted as a blue dashed grid). The region of interest (represented in black) has 2×2 bins, and the sampling points (colored red) have arrows linking them to the pixels they query. Quantizations in RoIPool produce misalignments, and RoIAlign uses 4 sampling points per bin that are merged into one value (colored purple).

Cropping and Resizing of Features. Once the features of the whole frame have been extracted, the following naïve step would consist in directly comparing said features with those of each exemplar using $\tilde{\kappa}$. However, doing so would result in two major problems: (i) it is inefficient to analyze the entire frame looking for an object, as it is unlikely that said object has undergone a large displacement relative to the previous frame—thus it is better to search for each object in a reduced area—; (ii) the similarity function would become less efficient as the object changes in scale, since the main trackers through similarity [4, 83, 82, 161] have a relatively low tolerance for discrepancies between the object’s size in the exemplar image and in the search area—supporting a maximum difference of up to a 15%, according to our experiments. To solve this, in SiamMT it is necessary to introduce a features crop and resize module $\tilde{\kappa}$ prior to the similarity operation (Figure 2.4(b)). This $\tilde{\kappa}$ operator is applied to the frame features and creates new fixed-size tensors in which each object is represented with a constant size, analogously to SiamFC’s κ operation. However, while crop and resize operations result straightforward in images, they are particularly challenging when carried out on features.

The first proposal to obtain a set of fixed-size feature maps from a nonuniform-sized input and a series of regions of interest comes with RoIPool [53]. RoIPool delimits each region of interest and divides it into a predetermined number of sections (bins)—e.g., 2×2 —, to then assign each bin the value corresponding to the highest value of the pixels it contains (Figure 2.5(b)). As an RoIPool successor, aiming for higher precision, RoIAlign [58] arises, which prevents misalignments by avoiding quantizations and obtains more representative values by computing the values of each bin by aggregating—maximum or average—4 bilinearly interpolated sampling points (Figure 2.5(c)).

Following RoIAlign’s approach, in SiamMT we introduce the $\tilde{\kappa}$ operator in order to crop and resize features. It is an RoIAlign variant that applies a region calculation capable of handling features extracted with a backbone without padding—see below—, and that employs a single sampling point per bin (Figure 2.5(a)). This allows $\tilde{\kappa}$ to maintain the inference speed while

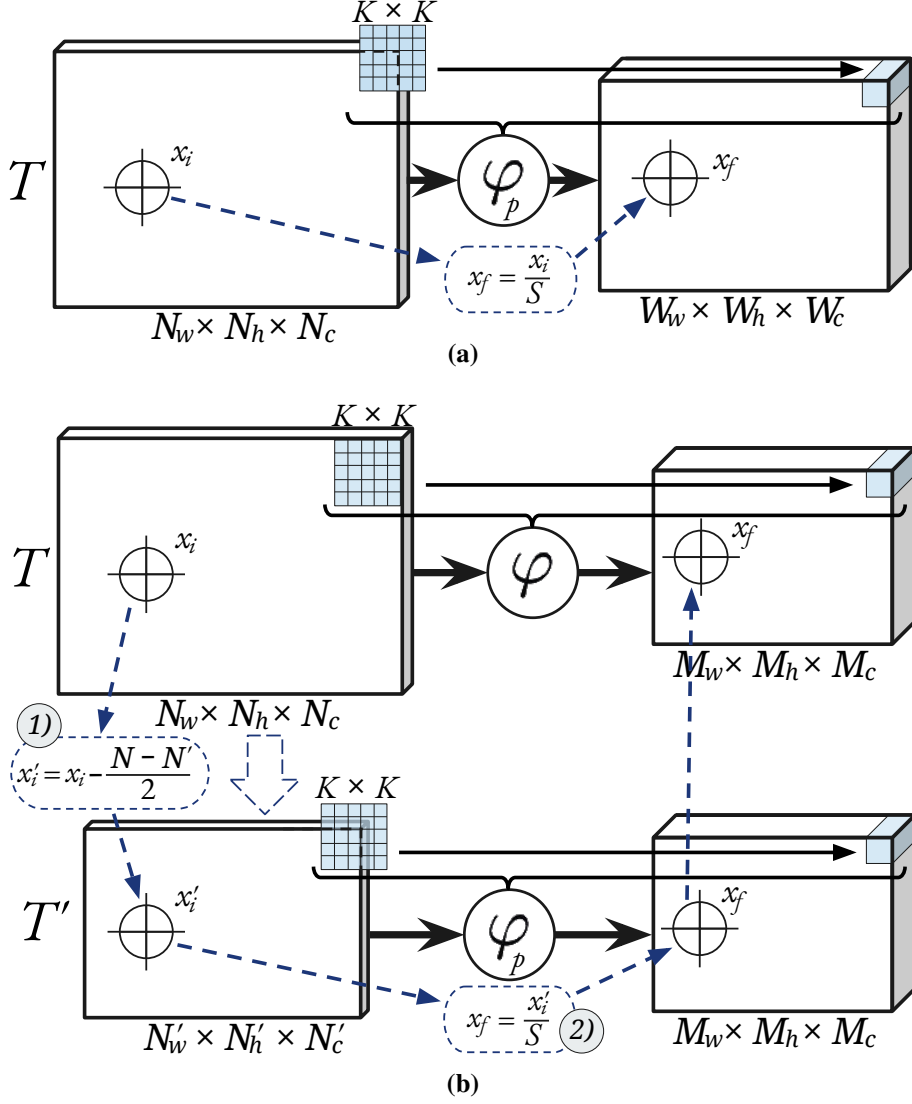


Figure 2.6: Transformation between pixel coordinates and features coordinates for (a) a backbone with padding using RoIAlign and (b) a backbone without padding using $\tilde{\kappa}$. When the backbone has no padding, the transformation cannot be performed directly using RoIAlign. Thus, we first clip the input tensor to its effective size (step 1)), and then apply the RoIAlign calculation on that tensor (step 2)).

obtaining more precise and representative values than with RoIPool. The ultimate goal of this operator is to reconstruct the information that would have been obtained from the direct features extraction of each rescaled image.

Features Coordinates Calculation. For the correct determination of the regions of interest, $\tilde{\kappa}$ redefines the way in which areas are delimited. In RoIPool and RoIAlign, the transformation between image coordinates x_i and features coordinates x_f is done by simply dividing the pixel coordinates by the backbone's (φ_p) global stride S (Figure 2.6(a)). This, whilst it works when the feature extractors are architectures with padding such as ResNet [59] or DarkNet [120], is incorrect if the backbone has no padding, as is the case of AlexNet [77], the backbone employed in SiamFC. Therefore, in order to calculate the region coordinates for backbones without padding, it is necessary to apply the transformations considering the *effective* size of the input tensor.

Let T be an input tensor of size N that produces an output of size M after passing through a fully-convolutional backbone φ without padding. We define N' , the *effective* size of T with respect to φ , as the size of the minimum subtensor of T that produces an output of size M after passing through φ applying padding in all its operations—we denote this configuration of the backbone as φ_p . It is possible to prove that every tensor T would have an effective size of:

$$N' = \left\lceil \frac{N - K + 1}{S} \right\rceil \cdot S - (S - 1) \quad (2.1)$$

where K and S are the receptive field and the global stride of φ , respectively.

This implies that T generates an M -sized tensor after passing through φ , and T' (the N' -sized clipping of T) generates a tensor with exactly the same size M after passing through φ_p . As we define N' as the minimum size that fulfills this property, if T' shrank, its features extracted using φ_p would be smaller than M . Conversely, if T' grew (but no more than $S - 1$ pixels in each dimension), the size of its features extracted with φ_p would remain M . By defining N' in this way, we can use T' as an intermediate step to transform coordinates extracted with a backbone without padding (φ), as shown in Figure 2.6(b).

Therefore, step 1) consists in calculating the new position of the object inside T' , namely x_i' . After this, it is possible to pose the problem as the extraction of the features of T' with φ_p , so the position x_f of the object in features will be obtained by dividing its new coordinates x_i' by the global stride of the backbone S (step 2)). Finally, these coordinates will become the new position of the object, since the dimensionality of the tensor generated by φ_p is the same as that of the tensor generated by φ . Consequently, given some input coordinates in pixels x_i , their respective coordinates in features extracted with the fully-convolutional backbone φ without padding will be calculated as follows, by concatenating said steps:

$$x_f = \frac{1}{S} \cdot \left(x_i - \frac{N - N'}{2} \right) \quad (2.2)$$

This transformation reveals two issues:

- In the SiamFC architecture, the size of the exemplar features tensor is smaller than the object it represents. As seen in Figure 2.7(a), on step 3) we extract the features of a 127×127 image containing a 63.5×63.5 object. As the backbone has a global stride of 8, the object becomes 7.9×7.9 pixels in size. However, since the backbone does not employ padding, the resulting exemplar features tensor has a size of only 6×6 pixels, and thus some of the object's edge information is washed away. Hence, during the similarity operation, the central regions of the object will receive the most attention and its edges will be mostly ignored.

– In the SiamMT architecture, if the region sizes were computed over the frame, the scale of the object would differ between exemplar and search area features. This becomes evident

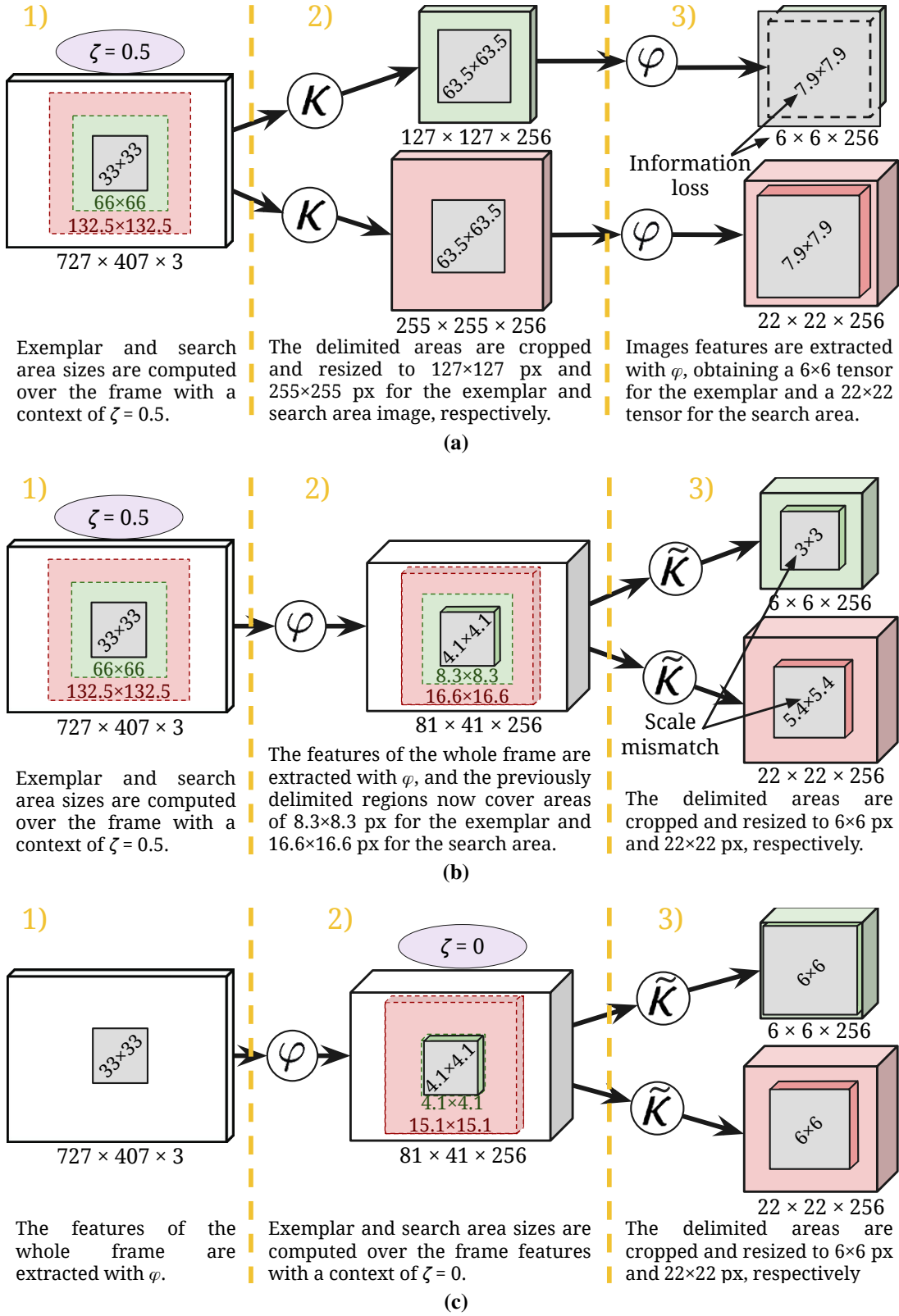


Figure 2.7: Numerical example for a 33×33 -pixel object across different architectures. Object is represented in gray, while its exemplar and search area regions are represented in green and red, respectively. (a) depicts the standard SiamFC architecture, (b) illustrates the mismatch when computing regions over the input frame, and (c) shows the method followed by SiamMT.

on Figure 2.7(b), step 3), as the exemplar features undergo a rescaling of a factor of 0.73, while the factor for the search area features is 1.33. This scaling difference is due to the fact that the size ratio between the defined areas is different from the size ratio between the features tensors. Thus, at the end of the process we obtain an object of size 3×3 at the exemplar branch, while the object has a size of 5.4×5.4 pixels at the search area branch. Ultimately, a difference like this ends up being very harmful to the network's performance, as the similarity operation will not be able to appropriately match the exemplar and the search area.

In summary: (i) SiamFC's exemplar features tensor is smaller than the object it represents, causing an information loss at its edges; (ii) in our architecture, if we computed the exemplar and search area regions over the input frame, the size of the object would differ between the exemplar and the search area tensors. Therefore, in order to solve these two issues, we calculate the regions over the frame features and without any additional context ($\zeta = 0$), as shown in Figure 2.7(c), step 2). This way, at the end of the process, the size of the object is the same for both tensors, and there is no loss of information at the edges.

Similarity Operation. Finally, of particular interest is the reformulation of the features comparison operation applied at the end of the architecture. This is because, in SiamFC and its derived architectures, it is defined as a cross-correlation operation between the exemplar features and the search area features tensor. In the case of SiamMT, as it is necessary to cross-correlate multiple pairs of tensors, this operation would have to be replicated throughout the batch size, which is computationally inefficient. As a solution, taking advantage of the properties of GPU architectures, we propose the use of the pairwise cross-correlation $\tilde{\star}$.

The proposed $\tilde{\star}$ operation (PairwiseXCorr) is described in Algorithm 1, and takes as inputs the exemplar and search area features (T_E and T_A) of N objects and efficiently compares them, generating N score maps. As we want this operator to be general and applicable to other architectures, the exemplar features tensor supports an additional dimension o , to vary the depth of the resulting score maps. This dimension allows the application of $\tilde{\star}$ on architectures that require many outputs for each cross-correlated pair, like the anchors regressions in a Region Proposal Network [82].

Operator $\tilde{\star}$ is made possible by the properties of two-dimensional depthwise cross-correlation (DwXCorr). The latter takes as inputs a three-dimensional tensor T_Y and a set of two-dimensional filters T_Z , and applies a different filter to each of the c channels of T_Y . Therefore, by appropriately stacking the objects' features contained in T_A and T_E , it is possible to obtain the T_Y and T_Z tensors (lines 2 and 3), which are the inputs to the depthwise cross-correlation operation. After the operation, an output T_P with $N \cdot c$ channels is generated (line 4). Finally, this output is reshaped (line 5) and aggregated for each object (line 6), which would correspond to the final sum of the c channels for each filter in a standard cross-correlation.

Algorithm 1: Pairwise cross-correlation.

- Let T^σ , be the permutation of the dimensions in tensor T , where σ is the permutation in cycle notation.
- Let $\text{vec}(T)$, be the vectorization of tensor T , which converts it into a column vector.
- Let $\text{vec}_{a_1, \dots, a_i, \dots, a_n}^{-1}(V)$, be the folding of vector V into an n -dimensional tensor, where a_i is the size of each dimension.
- Let $\text{DwXCorr}(T_Y, T_Z)$, be the two-dimensional depthwise cross-correlation between tensors T_Y and T_Z , with T_Z acting as the filters.

1 Function PairwiseXCorr(T_A, T_E)

Input: A T_A tensor of dimensions $[N, w_A, h_A, c]$ containing the features of N search areas, and a T_E tensor of dimensions $[N, w_E, h_E, c, o]$ containing the features of N exemplars to obtain o output channels.

Output: A T_S tensor of size $[N, w_A - w_E + 1, h_A - h_E + 1, o]$ containing the similarity score map between each search area in T_A and its corresponding exemplar T_E .

- 2 $T_Y \leftarrow \text{vec}_{1, w_A, h_A, N \cdot c}^{-1} \left(\text{vec} \left(T_A^{(1,2,3)} \right) \right)$
- 3 $T_Z \leftarrow \text{vec}_{w_E, h_E, c \cdot N, o}^{-1} \left(\text{vec} \left(T_E^{(1,2,3)} \right) \right)$
- 4 $T_P \leftarrow \text{DwXCorr}(T_Y, T_Z)$
- 5 $T_Q \leftarrow \text{vec}_{w_{\mathcal{A}} - w_E + 1, h_{\mathcal{A}} - h_E + 1, N, c_E, o}^{-1} \left(\text{vec}(T_P) \right)^{(1,3,2)}$
- 6 $T_S \leftarrow \sum_{l=1}^c T_Q(i, j, k, l, m)$ where $1 \leq i \leq N$, and $1 \leq j \leq w_{\mathcal{A}} - w_E + 1$, and $1 \leq k \leq h_{\mathcal{A}} - h_E + 1$, and $1 \leq m \leq o$
- 7 **return** T_S

This new operator is mathematically equivalent to applying $\star$ to each pair and has a great impact on the speed of the architecture, allowing it to scale to several dozen objects. Also, as the exemplar of an object remains constant during the inference phase, SiamMT can cache T_E and reuse it during the tracking process. Lastly, if multiple scales were to be considered for each object—as is the case with SiamMT—the batch dimension of the exemplar and search area features would have a size of $\nu \cdot N$, where ν is the number of scales considered.

2.3.3 System Training

SiamMT supports reusing the weights learned by SiamFC through a network-based transfer learning procedure, which consists in collecting all the parameters learned by a previously trained SiamFC network—backbone and similarity operation’s weights and biases—and copying them into their respective SiamMT operations. This alone is sufficient for performing inference, and provides good results as demonstrated in Section 2.4.2, hinting at the direct extensibility of SiamMT to other SiamFC-based architectures. However, due to the use of the crop-and-resize operator $\tilde{\kappa}$, it is convenient to fine-tune the last layers of the network. This tuning is very similar to an end-to-end network training, but freezing the first 3 convolutional layers and with a lower learning rate.

Regardless of whether the training is performed end-to-end or just fine-tuning, SiamMT's training is very similar to the one designed by SiamFC. For each iteration, two different frames containing the same object are fed into the network, one at the exemplar branch and the other at the search area branch. The features of the frames are extracted with φ , to later create the exemplar and search area features tensors through $\tilde{\kappa}$ using the locations provided by the dataset annotations. Following this, these two tensors are compared using $\tilde{\kappa}$, obtaining an $m \times n$ -size score map lg that seeks to represent the likelihood that the object is present in each region of the search area. Since the training is defined as a binary classification problem, the error is obtained by comparing this score map with a ground truth map gt , through a sigmoid cross-entropy loss:

$$loss = \sum_{j=1}^m \sum_{i=1}^n \max(lg_{ij}, 0) - lg_{ij} \cdot gt_{ij} + \log(1 + e^{-|lg_{ij}|}) \quad (2.3)$$

Since the exemplar and search area regions are defined around the locations provided by the dataset annotations, during training the object will always stand in the center of the search area tensor. Therefore, we can define the groundtruth map gt as an $m \times n$ -size matrix with the positive classes in an area of radius R around the center v of the tensor. Taking into account the global stride S of the network, the value of each element of this matrix can be defined as follows:

$$gt_{i,j} = \begin{cases} 1, & \text{if } S \cdot \|(i, j) - v\| \leq R \\ 0, & \text{if } S \cdot \|(i, j) - v\| > R \end{cases} \quad (2.4)$$

where $1 \leq i \leq n$, and $1 \leq j \leq m$

In addition to this, we also contemplate the use of negative examples, to prevent the network from learning to detect objects rather than distinguishing among targets. We achieve this by introducing pairs of frames that do not display the same object.

During training, there is the risk that the network erroneously learns that the tracked object always has the same proportion with respect to the search area. This is because, even though the frames go through a data augmentation process—involving translations, scale changes, rotations, color variations, motion blur and noise addition—, as the $\tilde{\kappa}$ operator uses the locations provided by the dataset annotations, it always generates a perfectly cropped and rescaled search area tensor. As this does not accurately represent a real tracking scenario, during training, a random factor is applied to the calculation of the dimensions of each search area. Therefore the system is able to gain tolerance towards object scale changes.

Finally, handling the disparity in size of the different objects across the datasets is not straightforward, as the training would have difficulties to converge if the same feature extractor was used for objects with very different sizes. Therefore, in order to reduce training time and ensure stability, the input frames are rescaled so that the considered objects always have a size among a predefined range. This phenomenon does not affect SiamFC-based architectures [4, 83, 82, 161], as they always rescale the images before extracting their features.

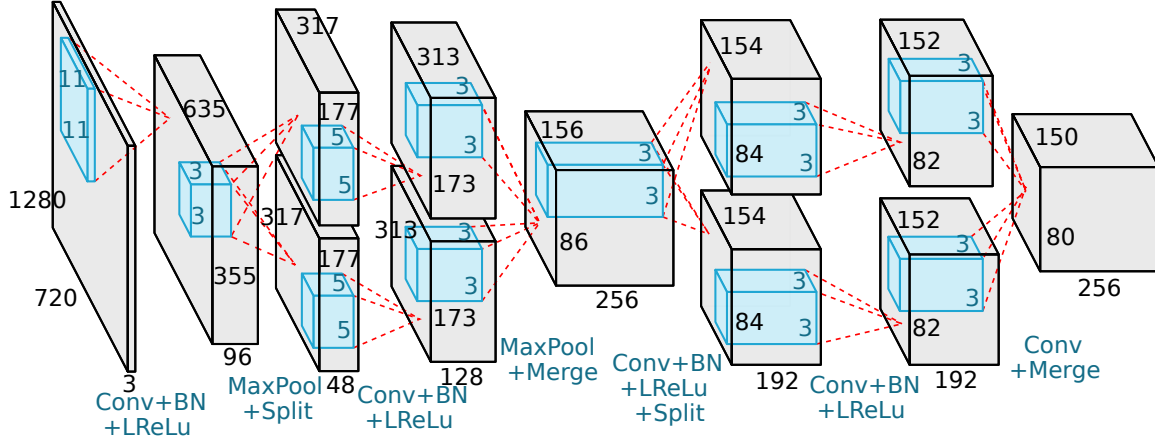


Figure 2.8: SiamMT’s feature extractor. The tensors and their sizes are represented in black, while the operations and their names are colored in blue.

2.4 EXPERIMENTAL RESULTS

This section evaluates SiamMT under different scenarios. The experiments were conducted on a computer with an Intel Core i7-9700K, 16 GB of DDR4 RAM and an NVIDIA TITAN Xp. The chosen deep learning framework was TensorFlow.

2.4.1 Implementation Details

Feature Extractor. The choice of the backbone is a determining factor in the network performance. While other state-of-the-art architectures such as ResNet [59] or DarkNet [120] would allow for richer features, they would result in a much longer inference time. Therefore, following the example of SiamFC [4] and some of its successors, we have opted for a backbone based on AlexNet for its simplicity and speed, but with a few modifications.

The specific architecture of the adopted backbone is shown in Figure 2.8. It incorporates batch normalization and non-linear Leaky ReLU activation functions after each convolutional intermediate layer. Moreover, convolutions in layers 2, 4 and 5 are grouped, as it makes training faster and helps in learning better representations of the data [67]. Finally, it is particularly important to note that no padding is introduced into the network, as this would lead to the loss of translation invariance [82].

Exemplar and Search Area Sizes. For the formulation of the exemplar and search area sizes, a similar approach to SiamFC [4] is taken. Thus, if we denote the size of an object’s bounding box as (w, h) , its exemplar will correspond to the region centered on its location, with an area

$$A = (w + \zeta(w + h)) \times (h + \zeta(w + h)) \quad (2.5)$$

where ζ is a context factor. After cropping the exemplar region, it is resized to the size of the destination tensor. In SiamFC, ζ is set to 0.5 and the rescaling is done to 127×127 pixels, as

the crop-and-resize is performed on images. However, since SiamMT's calculations are made directly on features, a value of $\zeta = 0$ with a destination tensor of size 6×6 provides the best results. In the case that the weights of the original SiamFC [4] architecture are directly reused, ζ should be adjusted accordingly to replicate SiamFC's field of view.

Similarly, the search area size is calculated as the quotient between its destination tensor and the destination tensor for the exemplar, all multiplied by the exemplar's crop region. In SiamFC, where the crop is made on an image, the destination tensor has a size of 255×255 pixels. On the other hand, as SiamMT crops features, a tensor of size 22×22 is chosen as the destination.

Training Process. The system was trained using the video databases provided by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [123], the YouTube-BoundingBoxes Dataset (YT-BB) [118], and the Generic Object Tracking Benchmark (GOT-10k) [66]. Together, these databases have about 255,000 sequences with over 9.8 million labelled bounding boxes, populating more than 560 object classes. The initial weights of the network are generated following a normal distribution. During training, we employ an Adam optimizer [74] to minimize the error function over 1,000 epochs, starting from a learning rate of 3×10^{-4} that is geometrically annealed to 3×10^{-6} , and applying L2 regularization ($\lambda = 5 \times 10^{-5}$) to the learned weights. Each epoch consists on 65,536 pairs of images—one for the exemplar and the other for the search area, containing the same object and spaced no more than 100 frames for the positive samples—arranged in size-32 batches. The best model is selected based on the validation error of the last 250 epochs.

Inference Process. The inference phase aims to be as simple as possible to achieve high tracking speeds. To accomplish this, the features of each exemplar are extracted at the beginning of the process and reused throughout it. This greatly reduces the number of operations, and also allows the tracker to be formulated as a one-shot detector. Indeed, it has been demonstrated that simple methods for updating the exemplars do not offer great advantages for short-term tracking. However, high-level updating techniques could be very useful for improving the performance on long-term crowded videos [26].

To tackle small objects, we resize the input frames so that the median size of the targets in the scene is above 32 px^2 . And lastly, SiamMT allows for the partial extraction of the frame features. With this technique, it considers only the minimum region in which the followed objects are found, plus a slight margin. This offers large speed gains when the objects are close together.

2.4.2 Tracking Quality Evaluation

Since SiamMT is an MVOT, it initially receives the starting location for each object and provides its new position in each frame, without further interaction with a detector. This, by definition, is how single-object trackers operate, and thus SiamMT's quality can be evaluated using the protocols of single object tracking benchmarks. As MVOTs are used in situations where processing

	MOT-17				MOT-20				UAVDT				VisDrone				JTA			
	@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Acc.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SiamFC++	53.9	70.2	52.3	67.6	30.4	70.9	28.8	69.7	52.4	86.1	48.5	81.2	34.2	32.8	32.0	31.5	44.4	59.8	42.1	56.2
SiamCAR	47.5	60.0	46.0	58.2	27.8	67.5	27.2	67.0	40.3	70.0	37.1	64.9	28.9	29.8	27.8	29.3	38.3	48.7	36.6	46.9
SiamRPN++	48.5	60.1	47.0	58.9	27.8	67.4	27.1	66.9	38.2	67.1	35.2	62.5	28.2	29.6	27.2	29.1	37.8	48.7	36.1	46.9
SiamRPN	50.9	70.7	49.5	67.9	30.3	71.0	29.1	69.7	53.5	89.9	49.8	85.7	36.2	34.4	33.9	32.8	43.7	60.3	41.5	56.8
SiamFC	45.8	58.4	45.2	57.2	26.9	67.3	26.6	66.9	41.4	73.3	37.5	67.3	27.9	29.5	27.0	29.0	36.3	46.3	35.2	45.1
SiamMT-W	52.7	73.4	52.4	71.2	50.0	74.6	50.0	75.3	53.8	92.6	53.9	92.6	46.8	59.3	46.5	58.3	47.3	57.1	47.2	57.0
SiamMT	54.5	76.2	54.5	73.8	52.6	81.7	52.5	82.2	54.6	92.3	54.5	92.3	45.7	52.2	45.4	51.5	46.9	61.5	46.9	61.5

Table 2.2: Quantitative results for tracking quality. Red, blue and green, represent 1st, 2nd and 3rd respectively.

time is a constraint, the metric that most faithfully represents this behavior is VOTChallenge’s VOT-RT [76], considering each object in each video as an individual sequence at the time of aggregating the values. This method reports two values: accuracy —amount of overlap between the prediction and the groundtruth— and robustness —percentage of frames in which the tracker has not lost the object, with an exponential sensitivity of $S = 30$ —, both with a real-time threshold of 20 fps. We have also run the tests with a 25 fps threshold, to be consistent with the definition of real-time of the rest of this work.

The experiments were run in various public multi-object video databases, seeking to cover a wide variety of scenarios where motion estimation systems are commonly used: MOT-2017 [104], MOT-2020 [35], UAVDT [169], VisDrone [186], and JTA [42]. We have compared SiamMT and SiamMT-W —i.e., SiamMT with its weights and parameters obtained directly from a previously trained SiamFC network through a transfer learning procedure— with their baseline, SiamFC [4], and the state-of-the-art methods SiamRPN [83], SiamRPN++ [82], SiamCAR [55], and SiamFC++ [161], whose multiple instantiations are frequently used as MVOTs for motion estimation tasks. The results are shown in Table 2.2.

As can be seen, the accuracy and robustness obtained by SiamMT are superior to those of the existing approaches, with differences as large as 21.2 points if we compare the accuracy of SiamMT against SiamFC++ —the state-of-the-art tracker with the highest score— at MOT-20 @25 fps. If we consider the experiments with the threshold @20 fps, SiamMT obtains results that are, on average, 7.2 points higher for the accuracy and 7.5 points higher for the robustness, comparing it with the state-of-the-art tracker that scores the highest at each dataset. Those datasets where SiamMT really excels are those with a larger number of targets, most notably MOT-2020 and VisDrone, which have an average density of 127 and 46 objects per frame, respectively. In such crowded scenarios the use of traditional MVOT systems is unfeasible, with our proposal scoring the best results by a large margin.

The difference between SiamMT and the other approaches becomes significantly larger if we increase the real-time threshold. For a video at 25 fps, the quality of the state-of-the-art methods drops noticeably, while SiamMT only drops on average 0.1 points in accuracy and 0.5 points in

robustness. Thus, if we compare SiamMT against the best-performing state-of-the-art models @25 fps, SiamMT’s accuracy and robustness are 9.3 and 9.7 points higher on average. These experiments show that traditional MVOT systems are unfeasible when there are many objects in the scene, being our approach the only able to scale naturally in real time.

Finally, although SiamMT-W obtains slightly worse results than SiamMT, they are still remarkably good considering that it has not been retrained or fine-tuned. In fact, for VisDrone, SiamMT-W consistently scores slightly better results than SiamMT, which may be attributed to it being the dataset with the largest number of distinct categories. This demonstrates the versatility of SiamMT’s paradigm, and suggests its potential extension to other SiamFC-based architectures.

2.4.3 Tracking Speed Evaluation

As previously stated, the inference speed of an MVOT solution is a key aspect since it determines whether a system is suitable for real-life scenarios or not. The main objective of SiamMT is to enable the real-time tracking of a large number of arbitrary objects in a scalable manner, something that has not yet been achieved in MVOT. Consequently, we have designed two different test sets to represent the most typical multiple object tracking scenarios: the MT-VGA benchmark, composed of videos with a size of 640×480 pixels, and the MT-HD benchmark, with 1280×720 -pixel videos. The purpose of these experiments is to test the scalability of a tracker with respect to the number of tracked objects. Thus, the benchmark starts by measuring the time required to track a single object for each video, and gradually increases the number of tracked targets all the way up to 100.

To demonstrate the scalability of the proposed solution, the SiamMT architecture will be compared against the multiple instantiations of SiamFC [4] and SiamFC-MI. SiamFC-MI is a proposal of ours that optimizes the SiamFC algorithm for multiple instantiations by stacking objects along its batch size to maximize GPU parallelization. However, it does not reuse features computations, nor does it implement any of our novel operators.

It is worth noting that SiamMT’s partial extraction of the frame features has not been applied during these tests. This technique provides a significant speed boost on most situations, so it would make SiamMT perform better for some sequences. However, we believe that employing it for these benchmarks would not be fair, as it is the network architecture itself the one that should stand out for its scalability, not the extra add-ons.

MT-VGA Benchmark. The results for the MT-VGA benchmark are shown in Figure 2.9(a). For a single object, all three architectures offer speeds above 60 frames per second, with SiamFC being the fastest due to its simplicity, reaching 112 fps. SiamFC-MI tracks one object at 89 fps—it is slower than SiamFC due to its batch management—and SiamMT tracks one object at 63 fps as, unlike the other two, it extracts the features of the whole frame.

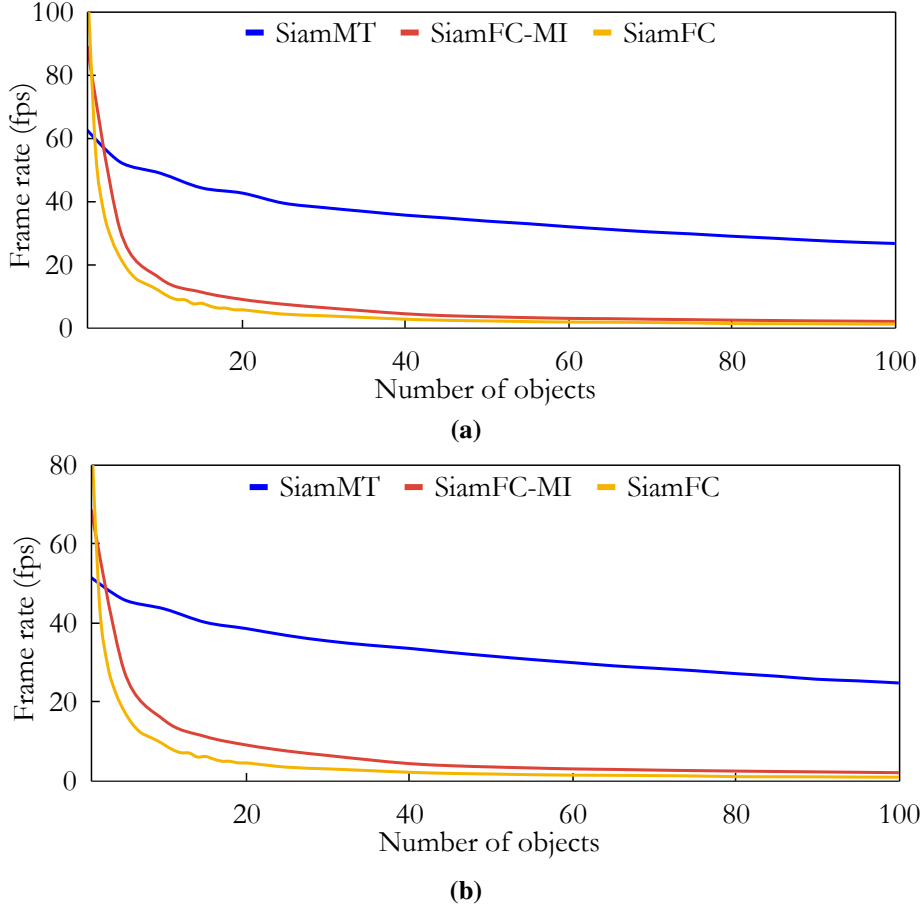


Figure 2.9: Speed results for (a) MT-VGA and (b) MT-HD benchmarks with respect to the number of tracked objects.

As more objects are added, the SiamFC and SiamFC-MI tracking speeds are reduced exponentially to just 11 and 16 fps for 10 objects, respectively. Meanwhile, the scalability of the SiamMT architecture becomes evident, dominating the rest of the graph and allowing to track 10 objects at 49 fps. This is possible because SiamMT runs the backbone only once per frame thanks to its features reuse, and because its final similarity operator exploits the properties of the depthwise cross-correlation, allowing multiple pairwise comparisons.

Thus, SiamMT scales almost linearly with the number of objects, being able to track 70 targets at 30 fps and —100 objects— at 27 fps. Furthermore, SiamMT is able to maintain speeds over 25 fps with up to 122 simultaneous objects. By contrast, even though SiamFC-MI offers higher speeds than SiamFC, its performance is far below SiamMT, tracking 100 objects at only 2 fps.

MT-HD Benchmark. The results for the MT-HD benchmark are shown in Figure 2.9(b). Since SiamMT performs the global features extraction of the 1280×720 -pixel frames, it has a higher overhead that becomes more noticeable for a low number of targets, tracking 1 object at 51 frames per second while SiamFC and SiamFC-MI do it at 89 and 69 fps, respectively. The latter are virtually unaffected by the resolution of the frame, as they only process a 255×255 crop of the image.

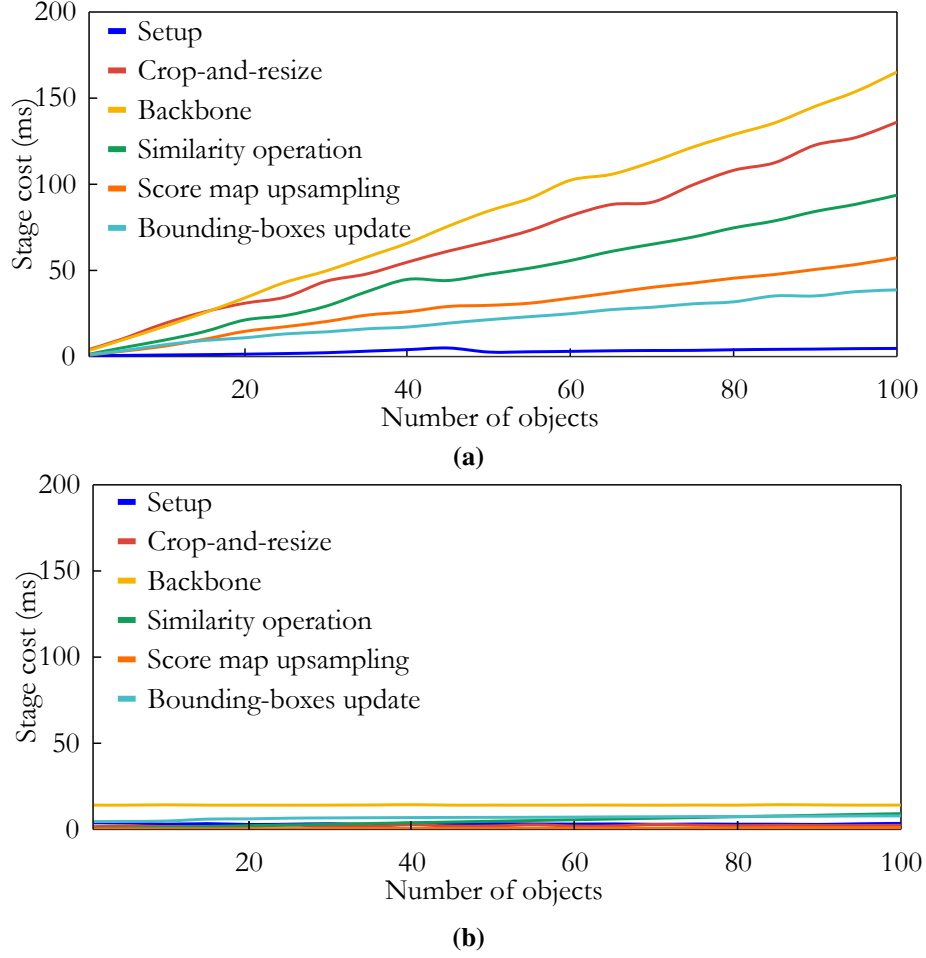


Figure 2.10: Comparison of average times per stage in the MT-HD benchmark for (a) SiamFC-MI and (b) SiamMT, with respect to the number of followed objects. In SiamFC-MI, the cost of most stages increases linearly. Meanwhile, in SiamMT, few are the stages whose time depends on N .

However, once above 3 objects, SiamMT outperforms the other architectures, with a computational cost that remains almost constant thanks to its reuse of features computations. Thus, SiamMT is capable of tracking 60 objects at 30 fps and reaches a speed of 25 fps for 100 objects. Meanwhile, SiamFC and SiamFC-MI do not exceed 2 fps with 100 objects, evidencing the scalability benefits offered by the SiamMT architecture.

2.4.4 Computational Cost per Stage Analysis

Figure 2.10 shows the cost of the main stages of SiamMT and SiamFC-MI for the MT-HD benchmark. As depicted, all the SiamFC-MI operations incur an execution time that increases with N , as opposed to the SiamMT ones, whose cost remains almost constant. Below is the description of these stages and the explanation for the cost difference:

- **Setup.** In this stage, all the necessary data are allocated to GPU memory in order to process the next frame. Both networks take the same tensors as inputs, yet the cost of SiamMT for this stage remains constant since it caches the exemplars' features — T_Z in Algorithm 1— directly in GPU memory.

- **Crop-and-resize.** This stage constitutes a crop-and-resize operation that creates the search areas for each tracked object. SiamMT solves this by applying $\tilde{\kappa}$ on the frame features, efficiently calculating the region sizes and generating small 22×22 tensors. However, SiamFC-MI implements this with κ , performing this operation directly over the frame image and generating three 255×255 images per tracked object. Therefore, the cost of κ is more sensitive to the number of tracked objects, making it grow with a greater slope.
- **Backbone.** Both SiamFC-MI and SiamMT employ the same AlexNet-based backbone φ to extract the images' features. However, while SiamFC-MI must use it on each of the $3N$ search areas, SiamMT only runs it once for the whole frame. The latter is less efficient when there are few objects on the scene, but is significantly faster when N is large.
- **Similarity operation.** This operation is responsible for generating a score map from the exemplar and search area features. SiamMT employs a similarity operator $\tilde{\star}$, specially designed for the comparison of pairs of tensors, as it performs the entire comparison within the same depthwise cross-correlation. This is very different from the sequential approach followed in SiamFC-MI, which requires one additional step per followed object.
- **Score map upsampling.** In order to reduce the coarseness of the score maps, they are upsampled to 272×272 pixels. SiamFC-MI's score maps are upsampled via bicubic interpolation. On the other hand, although slightly less accurate, SiamMT applies bilinear interpolation, which allows for much higher throughput.
- **Bounding-boxes update.** Lastly, the score maps are penalized in scale and translation, and the best candidate for each object is chosen. SiamFC-MI implements these operations sequentially, while SiamMT is able to embed them inside $\tilde{\star}$. Hence, when N is large, the non-maximum-suppression is substantially faster in the SiamMT architecture.

2.5 CONCLUSIONS AND FUTURE WORK

Current motion estimation modules work by instantiating multiple individual trackers, so they are only a viable option when there are few targets in the scene. In this paper we present SiamMT, the first deep-learning-based real-time arbitrary MVOT (multiple visual object tracker). It applies individual tracking techniques to multiple objects in an efficient and scalable manner, tracking 122 simultaneous objects in VGA video, and 100 objects in HD720 video, both at 25 fps. This is made possible through its global frame features extraction, its RoiAlign-based crop-and-resize operator, and a novel pairwise cross-correlation operation.

SiamMT has been evaluated in several video databases, obtaining a real-time accuracy and robustness superior to those of the current state-of-the-art —9.5 points more on average when compared to the best-performing counterpart at 25 fps—, and demonstrating the architectural

improvements that enable a speed-up of more than $20\times$ per object. In addition, as it extends on SiamFC, SiamMT allows reusing SiamFC's weights without requiring retraining, which further highlights its versatility and ease of adoption for multiple types of scenarios.

As future work, since SiamMT opens up the possibilities of extending real-time video analytics applications to many more objects, it would be worth integrating the present architecture on an embedded system, possibly trading accuracy and computation capacity for area and power consumption, but enabling video analytics directly at the edge without resorting to large uplinks for data transmission. The migration to backbones with padding would also be rewarding, as it would allow to run deeper neural networks at the cost of reducing processing speed.

3 Real-Time Siamese Multiple Object Tracker with Enhanced Proposals

The pursuit of real-time arbitrary multi-object tracking has led to significant advances in the field, with SiamMT representing an important milestone as detailed in Chapter 2. Despite its notable achievements, SiamMT faced several challenges that limited its applicability to a wider range of tracking scenarios. These challenges stemmed from its reliance on multi-scale testing of feature maps for object size estimation, as well as its inability to model interactions between targets or efficiently handle objects of extreme sizes —either too big or too small.

To address these limitations, this chapter proposes SiamMOTION: a powerful and versatile architecture built upon the foundations laid by SiamMT. SiamMOTION integrates state-of-the-art techniques inspired by object detection, enabling it to effectively tackle the aforementioned challenges while improving overall tracking performance. The efficacy of SiamMOTION has been rigorously tested and validated on five public benchmarks, showcasing its impressive performance in comparison to existing state-of-the-art trackers. Through its novel approach and improved capabilities, SiamMOTION sets a strong foundation for future research and development of more robust and efficient tracking solutions¹.

The organization of this chapter proceeds as follows. In Section 3.1, we explore the advantages of motion estimation in real-time multi-object tracking and discuss the contributions made by SiamMOTION. Section 3.2 surveys the current state-of-the-art and emphasizes its drawbacks. The architecture and modules of SiamMOTION are detailed in Section 3.3. Experimental results are presented in Section 3.4. Finally, concluding observations on the proposed method and possible directions for future research are provided in Section 3.5.

¹Code available at: <https://github.com/lorenzovaquero/SiamMOTION>.

The contents of this chapter were extracted from the following publications:

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks*. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), pp. 3485-3491, Montréal (Canada), 2022. ISSN: 2831-7475. DOI: 10.1109/ICPR56361.2022.9956673.

^a Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Real-Time Siamese Multiple Object Tracker With Enhanced Proposals*. Pattern Recognition, Volume 135, pp. 109141, 2023. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2022.109141.

^a Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.

3.1 INTRODUCTION

Visual object tracking consists in maintaining the identity of one or more targets throughout a video. This is among the first steps in video analytics applications, enabling systems to carry out functions that range from video surveillance to robot navigation [43]. Traditionally, multiple object tracking (MOT) has been addressed through the association of detections. Namely, for each new frame, a pre-trained detector is run in order to locate all the objects of interest in the scene. Following this, the current detections are associated with those of the previous frame, revealing the displacements and scale changes of the objects of interest.

However, when considering the use of a tracker to tackle a real-world problem there can be numerous constraints to consider. One of the most common and, at the same time, one of the most challenging restrictions is the real-time processing of the video, as traditional MOT systems are computationally very expensive. Amongst all the methods proposed at the MOT2020 Challenge [35], only 5 are able to run in real-time without considering the detector time. If we also take into account the detector runtimes, all these approaches should be discarded², as accurate detectors alone already struggle to keep up with real-time —EfficientDet-D3 [135] runs at 23 fps on an NVIDIA TITAN V for HD720 images.

Thus, it is necessary to adopt mechanisms capable of providing the position of the objects in all the frames without relying on continuous detections (Figure 3.1). This is what is often referred to as *motion estimation* between detections, which nowadays is powered by Visual Object Tracking (VOT) approaches, giving rise to the concept of multiple visual object tracking (MVOT). These methods are applied across various types of systems, as they benefit from the

²We consider that a system/module operates in real-time if it runs at least at 25 fps for HD720 resolution on an NVIDIA TITAN V or similar.



Figure 3.1: Comparison of (a) a multi-object tracking system lacking motion estimation between detections, (b) SiamMT [145], and (c) our proposal (SiamMOTION). Notice how (a) is unable to handle all frames and (b) does not detect changes in the aspect ratio of the objects.

latest advances in single-object tracking [21]. However, since these trackers are designed for a single target and their multiple instantiation is costly, such solutions are only suitable for uncrowded scenarios. To tackle this issue, the approach of addressing the problem more globally arises, trying to share as many computations as possible between objects [145]. This allows to keep up with several dozens of targets in real-time while applying single-object tracking approaches. Still, the full potential of this concept has not yet been exploited.

In order to expand the current trend of visual object trackers for motion estimation we propose SiamMOTION (**Siamese Multiple Object Tracker with Inertia and attentiOn Network**). SiamMOTION relies on the fundamentals defined in [145] for tracking multiple objects in an efficient and scalable manner and integrates them with the latest single object tracking methods, all while solving some core problems of the aforementioned architecture. SiamMOTION’s architecture includes a proposal engine (PE) that integrates an inertia module, a region-of-interest extractor, and an attention mechanism; and a comparison head (CH) composed of a region proposal network and a multi-object penalization module. The main novelties of our proposal are summarized as follows:

- The feature extraction is performed through a feature pyramid network (FPN) that allows to obtain meaningful features for all object sizes. Search areas are extracted from these features using an inertia module that takes into account the previous positions of each object. To the best of our knowledge, this is the first time these methods are employed in a detection-independent visual object tracker.

- A lightweight attention mechanism is employed to enhance the features most closely related to the objects of interest. This, together with a pairwise-depthwise region proposal network (PD-RPN), allows the prediction of accurate bounding-boxes in real-time.
- A novel multi-object penalization module is used to suppress distractors and outliers by taking into account all objects of interest in the scene. It models interactions between targets and applies 4 different types of penalizations, each one devoted to addressing a specific type of tracking error.
- We validate our proposal on five public datasets using VOT-RT metrics [76], achieving leading performance against current state-of-the-art trackers.

An initial iteration of this proposal was previously outlined in [142]. The main innovations presented in the current work, relative to the prior solution, encompass:

- The basis for design choices is further elaborated upon. Additionally, all the introduced operators are thoroughly defined and investigated, delving deeper into their effects and performance impact.
- Key issues with the previous architecture are pinpointed, incorporating specialized modules to address distractors and rapid movements.
- Experimental validation is enhanced by including a broader range of benchmarks with increased diversity and challenges.
- The proposed architecture provides better tracking results, achieving an average increase of +2.9 points in both accuracy and robustness.

3.2 RELATED WORK

3.2.1 Motion Estimation

Traditional multi-object tracking systems depend heavily on the quality of the detector employed, requiring accurate —and therefore costly— detectors to reach their full potential [46]. This poses a problem when there are real-time or hardware constraints, as it is not feasible to obtain detections for every frame. In this paper, we propose a novel multiple visual object tracker (MVOT) architecture to address this limitation. Such models can be integrated into full tracking systems, being able to estimate the motion of objects for those frames with no detections.

Initially, the preferred approach for estimating the motion of multiple targets between detections was through Bayesian filters [5]. However, current MVOT methods nowadays perform motion estimation through visual single-object tracking methods [21]. The most straightforward approach consists in instantiating a new individual tracker for each object that appeared in the scene. This can be carried out either by including the tracker as an independent component [185] or by embedding it into the architecture [166]. However, the caveat is that these methods cause the system to slow down with each additional object, so they are only suitable when the number of targets is small.

In order to allow the tracking of several dozens of objects and alleviate the problems mentioned above, [145] emerges. This tracker aims to share the most expensive operations of the architecture between the objects, reusing feature computations. This, combined with the new operators it introduces, results in most of the network having a constant computational cost, regardless of the number of objects. Thus, while [166] runs at 5 fps for an FHD video with 21 objects, [145] is capable of handling 100 objects at 25 fps. However, these approaches still present some problems and do not currently benefit from the latest advances in single-object tracking.

3.2.2 Visual Object Tracking

Motion estimation between detections is mainly performed using single-object online trackers. They comprise an initialization step, in which they receive the exemplar appearance of the object in order to integrate it into a similarity function. Then, for each new frame, this similarity function looks for the object of interest, reporting its new bounding box. Originally, this task was carried out using Discriminative Correlation Filters (DCF) which, by representing the object with a single filter, were able to distinguish the background from the target [11]. These approaches became increasingly sophisticated, modeling these filters as convolutional layers [171] and applying optimization frameworks to speed up the learning process [159].

However, the state of the art in tracking is currently driven by deep learning approaches. These trackers implement their similarity function through deep convolutional neural networks. Given the need to compare an exemplar image with a search area defined in the frame, Siamese neural networks emerge as the most natural choice, with [4] being the precursor of the current state of the art. To further enhance this architecture, contributions from other fields of computer vision were gradually incorporated. Thus, there are trackers allowing changes in the aspect ratio of the bounding box —using a Region Proposal Network (RPN) [83] or anchor-free [55]—, employing more sophisticated backbones [82], featuring attention modules [170], with segmentation information [167], or including branches for estimating the proposal quality [161].

Within the CNN approaches, some trackers somewhat deviate from the traditional Siamese formula of [4]. For example, [26] presents an architecture with dedicated components for object classification and estimation. The classification module specializes in discriminating between distractors, while the estimator component tries to predict the overlap between the prediction and the object. These types of networks are nondeterministic and thus are strongly affected by the initialization of the exemplar. This is why there have been efforts to incorporate modules capable of predicting the quality of such initializations [6]. The accuracy they offer is good, but as they iteratively refine the bounding box, these methods are computationally expensive and very sensitive to hyperparameters, which discourages their use as MVOT solutions.

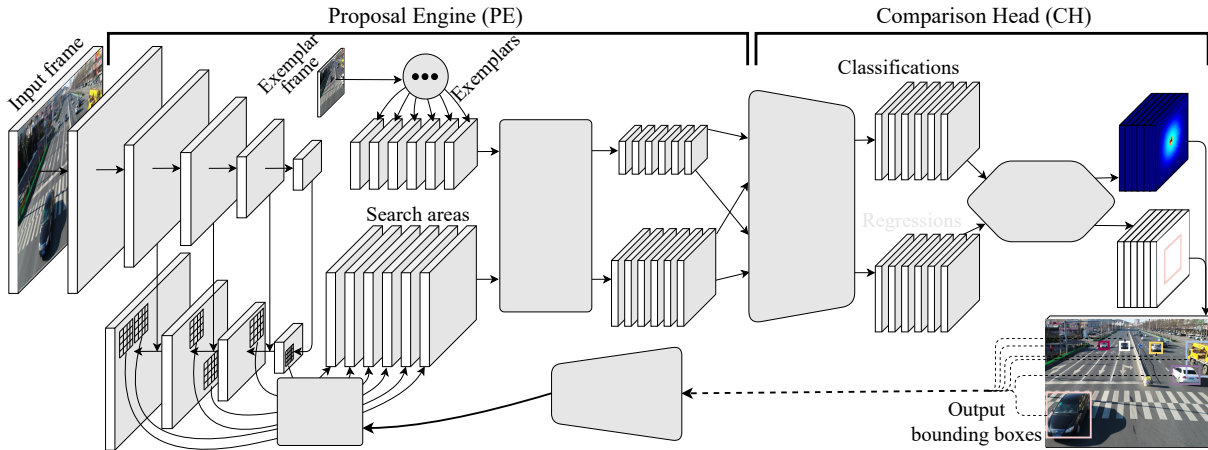


Figure 3.2: SiamMOTION’s architecture. First, it extracts the global features of the input frame. Then, it obtains the search area features of each object through the RoI Extractor, using the locations provided by the Inertia Module. The exemplar features are computed analogously at the beginning of the sequence and are reused throughout the rest of the video. Next, the attention mechanism is applied over these tensors and the previously extracted exemplars, before comparing them using the Pairwise-Depthwise RPN. Finally, the RPN output is fed through the Multi-Object Penalization Module to obtain refined bounding boxes for each object.

3.3 SIAMMOTION ARCHITECTURE

As shown in Figure 3.2 SiamMOTION’s architecture comprises: a backbone for feature extraction, a proposal engine (PE) that embeds a Feature Pyramid Network-based region-of-interest extractor, an inertia module for the definition of search areas, and an attention module; and a comparison head (CH) consisting of a region proposal subnetwork for classification and regression, and a multi-object penalization module. All these components are addressed in the following subsections.

3.3.1 Feature Extractor

Feature extractors transform an image from an (usually) RGB color space to a semantic embedding space, in which the various channels encode more meaningful information about the objects in the scene. Backbones that lack padding dilute the information at the edges of the image (Figure 3.3), but this does not pose a problem for single-object trackers since they define their exemplar and search area images with some margin. However, as SiamMOTION extracts the features of the whole frame, relying on a backbone without padding would result in poorer tracking quality for the targets close to the boundaries. This is a major problem considering that, in multi-object scenarios, targets enter and exit the scene mostly through the edges of the image. For this reason, SiamMOTION adopts a padded backbone based on ResNet [59]. Yet, unlike other Siamese trackers that employ deep padded backbones [82, 55], we keep the last convolutional blocks with their standard stride of 2.

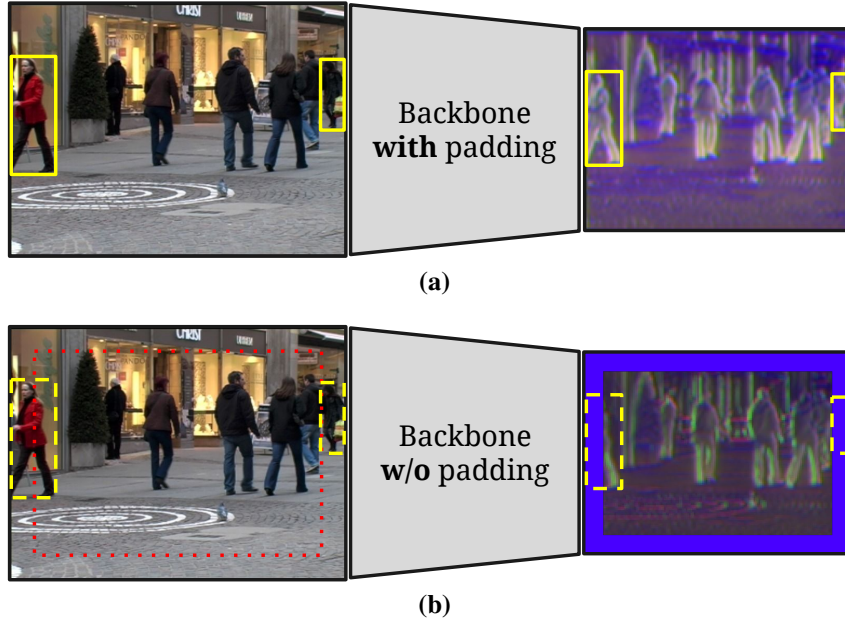


Figure 3.3: Features extracted with (a) a backbone with padding and (b) a backbone without it. The absence of padding results in a blind area —outside the red-dotted region— at the edges of the frame, which produces low-quality features for the highlighted objects.

3.3.2 Proposal Engine

Visual object trackers generate feature tensors for each search area, which will eventually be compared with the exemplars to locate each object in the scene. Approaches such as SiamRPN++ [82] or SiamAttn [170] clip fragments of the input frame and then extract their features, which produces accurate results but deems very slow for several objects. SiamMT [145] generates the search areas by cropping the frame features with a modified RoI Align [58], which improves the efficiency but produces low-quality tensors. The proposal engine (PE) presented in this paper efficiently generates high-quality features that properly fit the objects and capture their details with adequate resolution. It integrates an inertia module, a region-of-interest (RoI) extractor, and an attention mechanism.

3.3.2.1 Region-of-Interest Extractor

As previously stated, single-object trackers define a search area in the frame. This area is cropped and resized to a fixed size —usually 255 px^2 or 271 px^2 — before feeding it to the neural network. As a result, the target is displayed with a nearly constant size, which is the reason why [83] and its evolutions do not need to take into account different anchor sizes. However, since SiamMOTION extracts the features of the whole frame for several simultaneous objects, it cannot take advantage of this property. In [145] a new operator is proposed for the creation of search areas in the aforementioned conditions, yet it does not perform well with very small or very large objects, sometimes making it necessary to rescale the entire frame.

In order to obtain meaningful features for all search area sizes, we implement the region-of-interest (RoI) crop-and-resize operation through a feature pyramid network (FPN). Thus, instead of relying just on the final layer of the backbone, some of its intermediate layers also serve as inputs, capturing the scene at different resolutions. However, as higher-resolution layers contain poorer semantic information, we integrate them into a feature pyramid network to produce rich multi-scale representations. To the best of our knowledge, this is the first time such an approach is employed in a detection-independent object tracker.

Our RoI extraction mechanism has several similarities with those found in object detectors [87]. However, it presents a number of important differences:

- The input coordinates for the RoI extractor are not provided by a proposal generator based on the frame features, but by an inertia module that predicts the positions of objects based on their motion (Section 3.3.2.2).
- The regions covered by the RoI extractor are always square. This ensures that the objects are always depicted with the same aspect ratio relative to the frame (1:1), which improves the learning of the similarity function.
- The choice of the pyramid level k from which to extract the features is calculated as:

$$k = \left\lceil k_0 + \log_2 \left(\frac{\sqrt{A_w A_h}}{8 \lfloor 255/8 \rfloor} \right) \right\rceil \quad (3.1)$$

Here 255 and 8 are the canonical sizes of the search area and network stride in SiamFC-based trackers, respectively, k_0 is the FPN level with a resolution of $1/8$ (i.e., level 3), and A_w and A_h are the area's dimensions. In our architecture we consider values of k in the range $[1, 4]$.

- The cropping and resizing of the region is performed using RoI Align [58] with one sampling point per bin, for better computational efficiency. Following this operation, a 1×1 convolution is applied to transform the maps to a comparable representation, regardless of their original resolution.

The proposed RoI extractor yields tensors that will always depict the targets with an approximately constant size. This, along with the fact that the FPN provides a good feature hierarchy, means that tensors generated at different levels will have rich semantic information and will be comparable with each other. Therefore, it allows the rest of the network architecture to be independent of the level k from which each search area has been extracted—as object detectors that rely on a single shared class/box prediction head for all resolution levels do—, requiring fewer learned parameters and simplifying the convergence of the algorithm. This final remark is particularly relevant, as it makes our approach the first object tracker that compares features

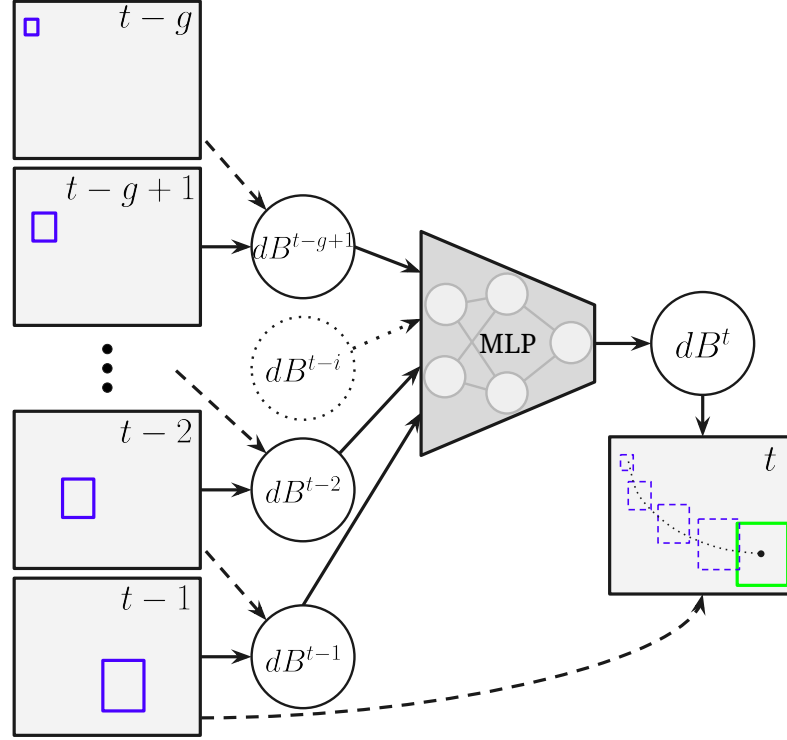


Figure 3.4: SiamMOTION's inertia module. Using a multilayer perceptron (MLP), it is able to make a coarse prediction of the object's future position based only on its previous coordinates.

extracted at different resolutions, since other systems that apply multi-layer similarities —e.g., SiamCAR [55] or SiamRPN++ [82]— adapt their backbones so that the considered layers have the same spatial resolution.

3.3.2.2 Inertia Module

For every new frame, single object trackers define an area to search for the object. The reason is twofold: (i) to keep the object constant in size —we address this with more detail in the next section—; (ii) and to avoid scanning the entire frame, as this would be computationally very expensive. Since its introduction in [4], following networks have adopted the same naïve approach for the definition of this area, centering it on the last known position of the object. This method provides good results but presents problems when the target moves fast or is very small —[83] alleviates it by making the search area larger when these cases show up, but carries a higher cost. To solve this problem, we propose the use of an inertia module that smartly adjusts the placement of the search area.

The inertia module makes a coarse prediction of the object's location and size at timestamp t —i.e., $B^t = \{(B_x^t, B_y^t, B_w^t, B_h^t)\}$ — solely from its coordinates in the previous frames, without resorting to visual information. Thus, it is fed by the predictions of the multi-object penalization module to roughly estimate the position of the object in the future. The inertia module is implemented through a multilayer perceptron (MLP) that receives the differences between con-

secutive coordinates dB for the last g video frames (Figure 3.4). These differences between pairs of coordinates are computed as bounding-box regressions, considering the previous instant as the anchor:

$$\begin{aligned} dB_x^{t-i} &= \frac{B_x^{t-i} - B_x^{t-i-1}}{B_w^{t-i-1}}, & dB_y^{t-i} &= \frac{B_y^{t-i} - B_y^{t-i-1}}{B_h^{t-i-1}} \\ dB_w^{t-i} &= \log_e \left(\frac{B_w^{t-i}}{B_w^{t-i-1}} \right), & dB_h^{t-i} &= \log_e \left(\frac{B_h^{t-i}}{B_h^{t-i-1}} \right) \end{aligned} \quad (3.2)$$

where $i \in [1, g)$. These parameterizations ensure scale-invariant locations and positive-sized bounding boxes.

Using a neural network provides an advantage over other classical models, as it is able to deal with noise in the input and easily adapt to the tracker’s particular behavior. Moreover, as it is fast and lightweight, its predictions can be used as a fallback mechanism if the tracker returns low-confidence outputs or struggles to keep the real-time performance. This module is trained on video sequences —it cannot exploit detection-only data as in [188]— and its weights are adjusted alternatingly with the rest of the tracker, as the the MLP inputs must come from the RPN proposals.

3.3.2.3 Attention Module

The similarity operator (Section 3.3.3.1) performs a sliding-window comparison between the exemplar features of each object and the tensor containing its search area. This very simple process is prone to problems, as the tracker has no way of knowing which features of the exemplar are truly relevant and tends to weigh everything equally. Thus, it is common to experience a degradation in tracking quality when the background is complex or contains distractors. To solve these problems, SiamMOTION integrates into its architecture an attention module with the objective of mitigating drifting and enhancing the features it relies on. This module is based on [170], but we have adapted it for multiple objects, suppressing the most expensive operations, and incorporating some optimizations.

Most of the attention mechanisms currently employed are based on the self-attention operation introduced in [146] for natural language processing. In short, this operation takes a tensor X as input and projects it into three different spaces, yielding three tensors denoted Query (X_Q), Key (X_K) and Value (X_V) —as in information retrieval, we can think of Query as the search term, Key as the information that defines each of the candidates, and Value as the content of those candidates. Next, it computes the cosine similarity between Query and Key, resulting in an interrelationship matrix (A) with high values for those features of X that are related and relevant to the current problem, and low values otherwise. This attention mask then is applied over Value, generating a filtered tensor ($\tilde{X}$) in which unnecessary details are suppressed and important information has been highlighted.

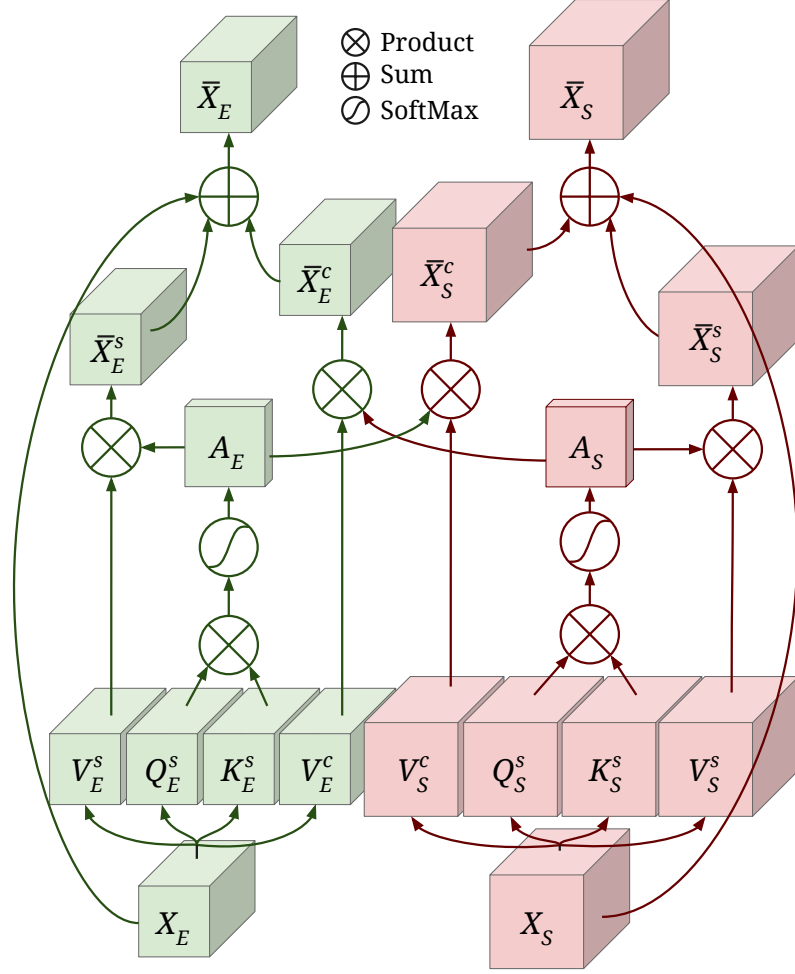


Figure 3.5: SiamMOTION's attention module has one branch for the **exemplar** and another one for the **search area**. Within each branch, channel-wise self-attention and cross-attention features are computed, which are then merged to obtain the exemplar and search area enhanced features. The above operations are performed concurrently for each target.

To apply this concept to visual object tracking, it is necessary to adapt the aforementioned operation. In [170] it is proposed a deformable subnetwork comprising 3 complementary mechanisms: a channel-wise self-attention, a spatial-wise self-attention, and a channel-wise cross-attention. According to our experiments, each of these components has a very different impact on system performance when dealing with several dozens of targets. Most notably, spatial-wise self-attention requires several additional convolutions and reshapes that culminate in the resource-expensive multiplication of two $(W * H) \times (W * H)$ matrices—for an input tensor X of size $W \times H \times C$ —, which diminishes the speed of the entire tracker by as much as 20%. For this reason, SiamMOTION bases its attention module solely on the channel-wise self-attention and cross-attention operations, and customizes their architecture for a more efficient performance (Figure 3.5).

SiamMOTION applies its attention module in parallel for all $\mathcal{N}$ pairs of tensors obtained through the RoI extractor. However, for the sake of simplicity, we will focus the explanation on

the computations involved for a single object. Thus, the attention module receives as inputs the exemplar (X_E) and search area (X_S) features of an object. For the self-attention computation—whose tensors are denoted with the superscript s —, 3 tensors are built in each branch— Q_E^s , K_E^s , and V_E^s for the exemplar; and Q_S^s , K_S^s , and V_S^s for the search area—, which have the same dimensions as X_E and X_S , respectively. Next, the cosine similarity between Query and Key is computed—this is obtained via the dot product of the tensors scaled by their magnitude—, and then a SoftMax is applied, restricting the values to the interval $[0, 1]$, thus creating the attention masks A_E and A_S . Lastly, A_E is multiplied with V_E^s , and A_S with V_S^s , yielding the self-attention features for the exemplar ($\tilde{X}_E^s$) and the search area ($\tilde{X}_S^s$).

Regarding the cross-attention computation—whose tensors are denoted with the superscript c —, Value tensors for the exemplar (V_E^c) and the search area (V_S^c) branches are generated from the input features. The latter are simply multiplied with the attention masks of the opposite branch, resulting in the cross-attention features of exemplar ($\tilde{X}_E^c$) and search area ($\tilde{X}_S^c$). Finally, the attention features of each branch are combined with the original tensors through a weighted sum, producing the final enhanced features ($\tilde{X}_E$) and $\tilde{X}_S$, whose sizes are the same as X_E and X_S , respectively.

To improve the performance and speed up the learning, SiamMOTION relies on standard 2D convolutions, as the accuracy gain that deformable convolutions offer does not compensate for the drop in throughput when dealing with several dozens of objects. In addition to this, the aggregation of X with $\tilde{X}^s$ and $\tilde{X}^c$ is performed in a single step, maximizing the use of computing resources. Regarding the exemplar branch, in SiamMOTION, exemplar features (X_E) are extracted once and reused throughout the whole tracking process. Hence, it is possible to cache the tensors X_E , $\tilde{X}_E^s$, V_E^c , and A_E to speed up the computations of $\tilde{X}_E$ and $\tilde{X}_S^c$ for the subsequent frames of the sequence. The latter, although it implies a higher memory consumption per object, results in a significant increase in tracking speed.

3.3.3 Comparison Head

Once the features of the search areas are available, it is necessary to compare them with those of the exemplars in order to get the new coordinates of the objects. Networks such as SiamMT [145] or SiamFC [4] employ a simple cross-correlation for determining the location of the targets and rely on multi-scale testing for determining their size, which is fairly straightforward, but does not allow detecting changes in aspect ratio. Approaches such as SiamRPN [83] or SiamFC++ [161] employ more sophisticated region proposal networks, but perform slowly for multiple objects and tend to produce noisy outputs when distractors are present. We propose a comparison head (CH) that efficiently compares multiple pairs of exemplars and search areas, generating quality predictions in multiple object scenarios. The head comprises a Pairwise Depthwise Region Proposal Network (PD-RPN) and a multi-object penalization module.

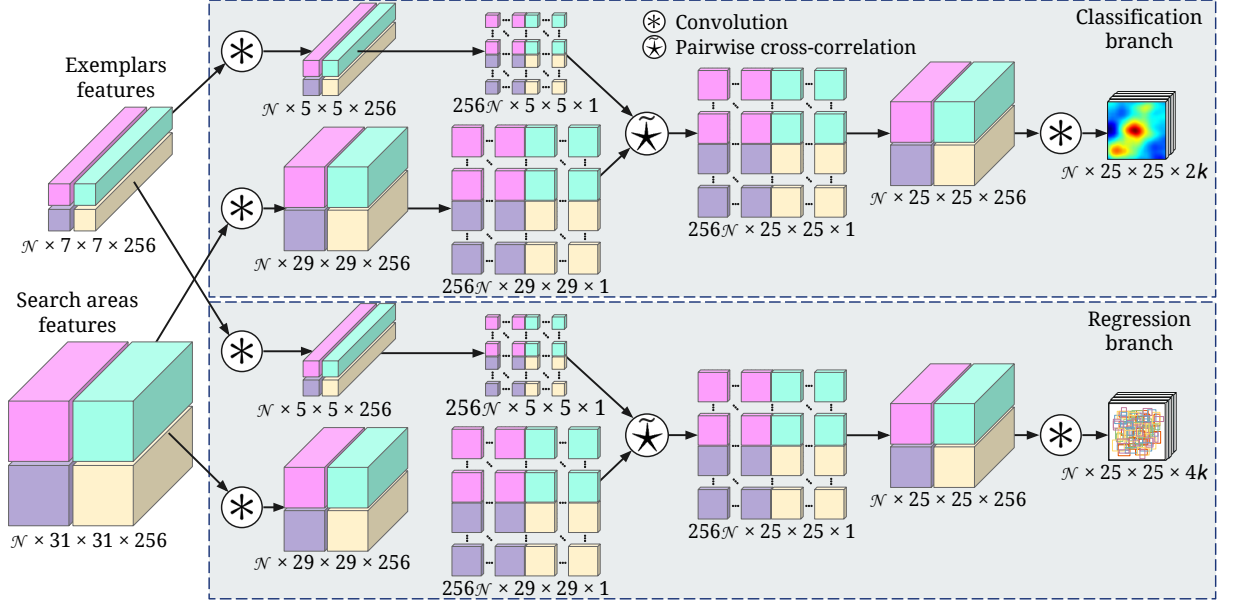


Figure 3.6: Pairwise-Depthwise-RPN. The input features pass through 3×3 convolutions that specialize them for each branch. After this, they are reshaped to then perform a pairwise-depthwise-cross-correlation using $\star$. Finally, the features are reshaped a second time and 1×1 filters are applied to obtain $2K$ or $4K$ values per location—depending on the branch—, yielding the classification and regression information of the different anchors.

3.3.3.1 Similarity Operation

SiamMOTION requires a fast and accurate similarity operator, capable of comparing dozens of tensors in real time. Previous works [4, 145] computed the similarity between the search area and the exemplar straight up through a cross-correlation, obtaining a two-dimensional score map stating the probability of the object being in each region of the search area. This can only identify translations, so multi-scale testings are carried out to detect changes in size. The method is far from optimal, as it cannot detect aspect ratio variations and requires to compare $\mathcal{S} * \mathcal{N}$ feature maps, where $\mathcal{S}$ and $\mathcal{N}$ are the number of considered scales and objects, respectively.

To deal with changes in ratio and scale in a natural way, other methods such as [83] employ a region proposal subnetwork. They first transform the input features according to the number of considered anchors K and then cross-correlate them, directly yielding the objects' classification and regression information. The downside of this approach is that it requires $6KN$ comparisons— $2KN$ for the objectness classification plus $4KN$ for the bounding box regression—, which greatly increases the computational cost when there are many targets involved. To address the aforementioned problems, [82] proposes a depthwise-RPN, which first cross-correlates the feature maps, to then apply the anchor-dependent transformations to their outputs. This has the advantage of involving just $2N$ comparisons, requiring an order of magnitude fewer parameters, and making the different channels more discriminative.

Owing to these benefits, SiamMOTION implements its similarity operation based on the one introduced in [82]. However, as this operator is designed for single-object trackers, it is necessary to modify it as depicted in Figure 3.6 in order to enable it to process dozens of targets

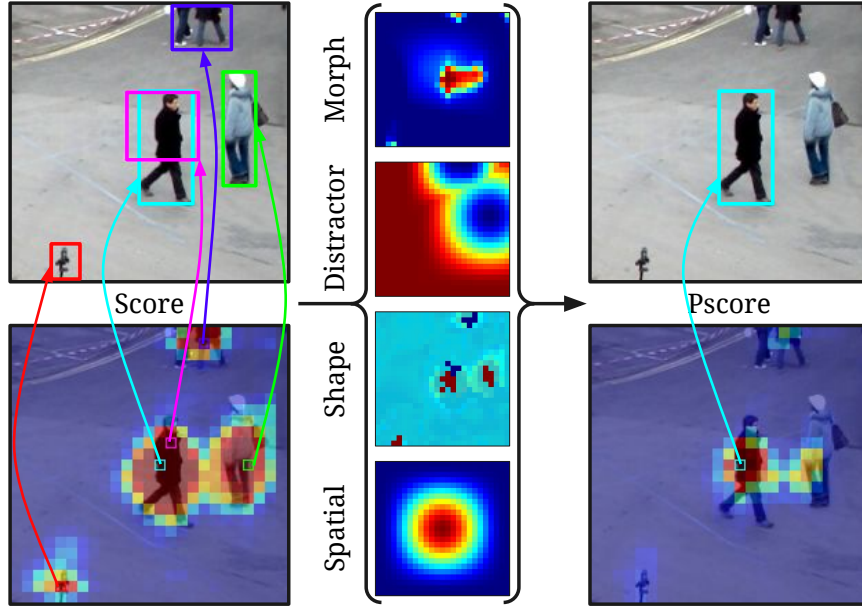


Figure 3.7: SiamMOTION's multi-object penalization module. The left side of the image depicts the output of the RPN classification branch with some of its associated bounding boxes, while the right contains the penalized score resulting from applying the four proposed penalties.

in real-time. To achieve this, SiamMOTION relies on the pairwise cross-correlation operator ($\tilde{\times}$) described in [145] as the core of this Pairwise-Depthwise-RPN, which enables the computation of all objects and sub-windows in a single evaluation. In order to perform these correlations in a depthwise manner, it is however necessary to reshape the inputs and outputs of $\tilde{\times}$. Lastly, since the search areas are extracted through multi-level regions of interest, SiamMOTION does not need an ensemble of heads at different stages of the backbone —as [82] does—, which greatly improves the efficiency of the algorithm.

3.3.3.2 Multi-Object Penalization Module

During inference, most single-object trackers [4, 83, 6, 55, 161] refine their predictions using some form of heuristic knowledge. The most commonly adopted methods are spatial and shape penalizations. However, since SiamMOTION is an MVOT and will consider several objects at once, it can exploit this information and develop a more sophisticated and powerful penalization module. Thus, as shown in Figure 3.7, SiamMOTION applies two new types of penalties —distractor-aware and morphological—, resulting in a multi-object penalization module with four different types of refinements. Additionally, the predictions produced by this component are fed to the inertia module, which uses them to roughly forecast the position of objects in future frames.

Spatial Penalization. Object velocities are usually moderate and situations in which a target undergoes a sudden large acceleration are extremely rare. Thus, with a high probability, the object will remain in the central portions of the search area. To model this knowledge, we follow the approach in [4] and apply a two-dimensional Hanning window to the output of the

classification branch in order to penalize large spatial displacements. With this method, we are boosting proposals such as the blue, pink, and green found in Figure 3.7.

Shape Penalization. Whether it is due to changes in perspective or deformations, the objects in the scene change shape gradually in most cases. Thus, to avoid accepting proposals with too different sizes or aspect ratios, we adopt a slightly modified version of the penalty described in [83]:

$$penalty = e^{-\beta * \left(\max\left(\frac{r}{r'}, \frac{r'}{r}\right) * \max\left(\frac{s}{s'}, \frac{s'}{s}\right) - 1 \right)} \quad (3.3)$$

where β is the hyper-parameter that adjusts the strength of the penalty, r and r' are the aspect ratios of the proposal and the object in the last frame, and s and s' are their overall scales, respectively. This penalty, which is computed based on the regression branch and applied to the output of the classification branch, allows to suppress proposals which have a high objectness score but are linked to a poorly adjusted bounding box, such as the pink one in Figure 3.7.

Distractor-Aware Penalization. As SiamMOTION is an MVOT, it is aware of all the tracked objects that might potentially be similar to each other. The proposed architecture takes advantage of this information and defines a novel distractor-aware penalization mask that minimizes identity-switches. Thus, unlike other methods that handle each target independently, SiamMOTION employs a simple-yet-effective approach to model interactions between objects.

The existence of distractors is modeled by a global penalization window $G_{W \times H \times N}$. This is a tensor whose N channels contain the contribution of each object to the distractor model, centered on their location and proportional to their size. The window has a size of $(W, H) = \lceil \frac{M * F}{S} \rceil$ —where M , F , and S are the sizes of the RPN output classification map, the input frame, and the canonical search area size employed in Siamese trackers [4], respectively—and the value of each element is computed as:

$$\begin{cases} 1 - \left(\sin\left(\pi \frac{A_p^x - 0.5A_p^w - i}{A_p^w - 1}\right) \sin\left(\pi \frac{A_p^y - 0.5A_p^h - j}{A_p^h - 1}\right) \right)^2, & \text{if } i \in A_p^x \pm \frac{A_p^w}{2} \text{ and } j \in A_p^y \pm \frac{A_p^h}{2} \\ 1, & \text{otherwise} \end{cases} \quad (3.4)$$

where $i \in [0, W)$, $j \in [0, H)$, $p \in [0, N)$, and $A_{4 \times N} = \{(A_p^x, A_p^y, A_p^w, A_p^h)\}_{p \in [0, N)}$ is the objects' search area coordinates mapped over the penalization window.

From the model provided by $G_{W \times H \times N}$, it is possible to estimate how the distractors will influence the model predictions. Thus, to mitigate their influence, a window R_p of size M is computed for each object p as follows:

$$R_p = \min_{d \in \mathcal{D}} \kappa_M(G_d, A_p) \quad (3.5)$$

where $\mathcal{D} := \{\forall d \in [0, N), d \neq p\}$ and κ_M is the preferred crop-and-resize operator with an output of size M —in our case, RoI Align [58]. This yields a mask that can be applied to the probabilities of the classification branch, suppressing proposals such as the green and dark blue ones found

in Figure 3.7. This novel approach for modeling interactions between objects proves itself to be effective and has a low computational cost, which makes it ideal for environments with dozens of targets.

Morphological Penalization. When computing the similarity between two feature maps, there are some situations in which outliers appear. These are characterized by being isolated false positives —unlike correct matches, which are spread over a wide area— and having a high score. To suppress these points, we propose a novel penalization based on morphological operations. Specifically, we rely on erosion, whose main goal is to shrink the shapes contained in grayscale images. Thus, we slide a 3×3 erosion kernel, which removes isolated peaks —such as the red proposal in Figure 3.7— and has the added benefit of reducing the area of the correct matches, as very wide boundaries can lead to problems in crowded scenarios. We apply said window independently on each of the score map channels, as the activations for different anchor shapes are weakly related.

3.4 EXPERIMENTAL RESULTS

In this section, we assess the performance of SiamMOTION under different scenarios. The experiments were conducted on a computer with an Intel Core i7-9700K, 16 GB of DDR4 RAM and an NVIDIA TITAN Xp. The chosen deep learning framework was TensorFlow.

3.4.1 Implementation Details

Feature Extractor. We opted for ResNet [59] as it offers good results with low resource requirements, something to consider when including level 1 of the FPN, which has a large memory footprint. Regarding the specific network configuration, we chose ResNet-18, as it offers a good tradeoff between accuracy and speed. It might seem that we can grow the backbone further without much impact on the throughput of the network since the following ResNet levels hardly increase the number of parameters. However, according to our observations, FLOPS do not translate well to fps, thus using more complex backbones would make SiamMOTION lose the ability to operate in real time.

Inertia Module. The inertia module receives the information relative to the last 6 known positions of the objects, since according to our experiments they are enough to generate reasonable predictions. It comprises 2 hidden fully-connected layers with hyperbolic tangent activations and ends in 4 neurons with linear activations. At the beginning of a sequence, the inputs of the MLP network are initialized to 0 —as the module receives differences between coordinates, it simply assumes that the objects were stationary up to this point in time. Regarding the training of the MLP, standard smooth L_1 loss is used for each component of dB^t .

Exemplar and Search Area Sizes. The sizes of the exemplar and the search area influence many architecture decisions and, thus, deserve careful consideration. Depending on their size, the output tensors will have a different granularity, which will directly impact the accuracy and speed of the network. As proposed in [82], an output of size 25×25 obtained from the comparison of exemplar and search area features of sizes 7×7 and 31×31 , respectively, offers a good balance.

These tensors are created from the frame features for each object, using the RoI extractor. Specifically, for a target with dimensions (w, h) , its exemplar will cover an area of:

$$A^2 = (w + \zeta(w + h)) \times (h + \zeta(w + h)) \quad (3.6)$$

where $\zeta = 0.5$ is the context factor. This area will be mapped to 15×15 bins. However, since the cropping is performed on features, there is no need for extra context to accommodate for further convolution operations, so only the central 7×7 region is kept. For the search area, a region of size $\left(\frac{31A}{15}\right)^2$ is cropped, which provides the same resolution as the exemplar.

Training Process. SiamMOTION is trained on a single GPU using the classification and regression losses defined in [83] and employing an Adam optimizer [74], starting from a learning rate of 3×10^{-5} that is exponentially decayed to 3×10^{-7} with a batch size of 16. We build on a backbone pre-trained on ImageNet, which we freeze for the first 15 epochs during the warmup stage. Following this, we train the network end-to-end for 30 epochs, applying a 0.1 correction factor to the backbone gradients. Lastly, we perform a fine-tune for 15 epochs in which we favor samples drawn from video sequences and intra-class discrimination, and during which the inertia module is alternatingly trained with a dropout rate of 50%. We maintain moving averages of the trained parameters with an exponential decay of 0.9998, and add a weight decay of $5e - 5$ to the loss function.

The system is trained on COCO[90], ILSVRC [123], YT-BB [118], and GOT-10k [66], which comprise approximately 250K sequences and 1.5M images, for a total of roughly 11M bounding boxes. It receives pairs of images that will serve as exemplars and search areas, updating the weights of the network as it learns —the feature extractor, RoI extractor, and attention module share their parameters for both exemplar and search area branches in a Siamese manner. To speed up the training, the network is not fed with pairs of full frames, but with only those portions that contain objects. These images undergo a data augmentation process that ensures their correct distribution into one of the levels of the FPN. Another consideration to keep in mind is that the performance of the network will be severely harmed if the RoI Extractor is fed directly with the ground truth coordinates in those cases where the inertia module is not queried. Therefore, it is very important to introduce noise in such circumstances, in addition to applying the spatial-aware sampling strategy described in [82].

Inference Process. In order to be as efficient as possible, the exemplar features are extracted only once and reused throughout the rest of the inference process. There is no need to update

	MOT-17		MOT-20		UAVDT		VisDrone		JTA	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
Baseline (13.9M)	50.1	74.6	47.9	82.1	42.0	75.8	42.8	50.2	45.3	62.1
B+F (15.2M)	55.4	76.8	48.1	82.3	51.0	92.5	46.3	56.0	48.0	66.1
B+F+R (16.4M)	57.3	76.6	51.2	83.0	56.5	93.2	48.7	57.0	49.8	67.1
B+F+R+A^s (17.0M)	57.4	76.5	50.8	83.3	56.6	93.1	48.9	57.2	49.9	67.1
B+F+R+A^c (17.0M)	57.2	76.7	51.8	83.2	57.2	93.2	48.7	57.1	49.8	67.1
B+F+R+A (17.0M)	57.3	76.6	51.4	83.2	57.0	93.2	48.9	57.1	50.0	67.2
B+F+R+A+P (17.0M)	58.0	77.5	51.5	82.7	57.8	93.2	50.8	59.3	51.0	66.2
B+F+R+A+P+I (17.0M)	57.9	77.2	52.7	82.2	57.8	93.3	50.9	59.8	51.0	66.3

Table 3.1: Ablation study with VOT-RT metrics @20 fps. A version of SiamMT with a ResNet-18 as backbone is the baseline (B). F: Feature-Pyramid-based RoI extractor. R: PD-RPN similarity operation. A: Attention module —superscripts s and c indicate self-attention or cross-attention only, respectively. P: Multi-object penalization. I: Inertia module. The number of parameters is shown together with the name of each configuration.

them, since they are already implicitly adapted for each frame thanks to the cross-attention mechanism. The component of the network that is continuously updated is the inertia module, which is fed with the outputs of the predictor. However, since it does not make use of visual information, it does not have a noticeable impact on the speed of the architecture. Lastly, we apply non-maximum suppression and bounding box voting with an IoU of 80% on the final output maps for a more precise localization.

3.4.2 Ablation Study

SiamMOTION is an MVOT, which implies that it receives the initial position of each object and provides their locations in the following frames, with no further feedback from the detector. These algorithms are mostly used in scenarios where time is a critical resource, so it must be taken into account when evaluating their performance. As such, we evaluate the performance of SiamMOTION through VOTChallenge’s VOT-RT metrics [76] adapting it for sequences containing multiple targets. Thus, for a throughput threshold —i.e., 20 fps or 25 fps— we obtain the accuracy —average overlap between predictions and ground truths— and the robustness —ratio of frames where the tracker did not lose the object, with an exponential sensitivity of $\gamma = 30$ — of each algorithm.

In order to analyze the impact of each component of SiamMOTION, we have conducted an ablation study on several scenarios in which motion estimation systems are commonly used. Specifically, we have selected the multi-object public datasets: MOT-2017 [104], MOT-2020 [35], UAVDT [169], VisDrone [186], and JTA [42]. The results are shown in Table 3.1.

As the **baseline (B)** for our study, we build on an architecture very similar to [145], but with a backbone based on ResNet-18 [59] and without dynamically adjusting the input frame size. The results are fairly good, but the performance is especially poor in videos with very small or very large objects, like the ones found in UAVDT, as the learned filters are not able to extract meaningful information.

The addition of the RoI extractor embedding a **feature pyramid network (F)** is a very significant change, increasing the accuracy and the robustness in all the datasets listed in Table 3.1 by an average of +4.1 and +5.8 points, respectively. The most substantial gain is found in UAVDT—increase of +9.0 points in accuracy and +16.7 points in robustness—, as the network is now able to generate meaningful features at different resolutions and, therefore, handle small objects, which are very abundant in this dataset.

The similarity operator also has a major impact on network performance, as the change from a pairwise cross-correlation to our **pairwise-depthwise-RPN (R)** results in an average increase of +2.9 points in accuracy. This is mainly due to the fact that the system is now able to detect changes in the aspect ratio of objects, all while keeping the real-time performance. The average increase in robustness is also significant (+0.6 points), since the output of the pairwise-depthwise-RPN objectness branch is very similar to that of pairwise cross-correlation, with both focusing on locating the center of the targets.

The integration of the **attention module (A)** results in an average increase of +0.2 points in accuracy and +0.1 points in robustness. As many scenes contain multiple objects with a high degree of overlap—which makes the center of the objects difficult to identify—the attention mechanism focuses on producing richer features capable of better delimiting the boundaries of each object. There are certain datasets in which one attention mechanism performs better than the combination of both. However, the best overall results are obtained through the use of both approaches.

The addition of the **multi-object penalization module (P)** also provides a significant improvement in the performance of the network. The suppression of distractors and outliers as well as the reduction of the activation area of detections provides clean results for the bounding box voting, which results in an average increase of +0.9 points in accuracy and +0.3 points in robustness.

Finally, the inclusion of the **inertia module (I)** also provides an improvement in the performance of the network, increasing the accuracy in +1.2 points and the robustness in +0.5 points for MOT-2020 and VisDrone, respectively. Specifically, as the inertia module is able to roughly predict the future coordinates of objects, it allows a better placement of the search areas, which prevents faster targets from leaving the network's field of view from one frame to the next. On the other hand, in environments with a high density of objects where the visual information cannot be fully trusted due to continuous occlusions, the inertia module plays a major role in enhancing those RPN predictions that present low confidence. Although the other datasets do not feature these characteristics, VisDrone has many fast targets and MOT-2020 presents a large number of overlaps between objects (Figure 3.8), so the introduction of the inertia module in these cases offers a great advantage.

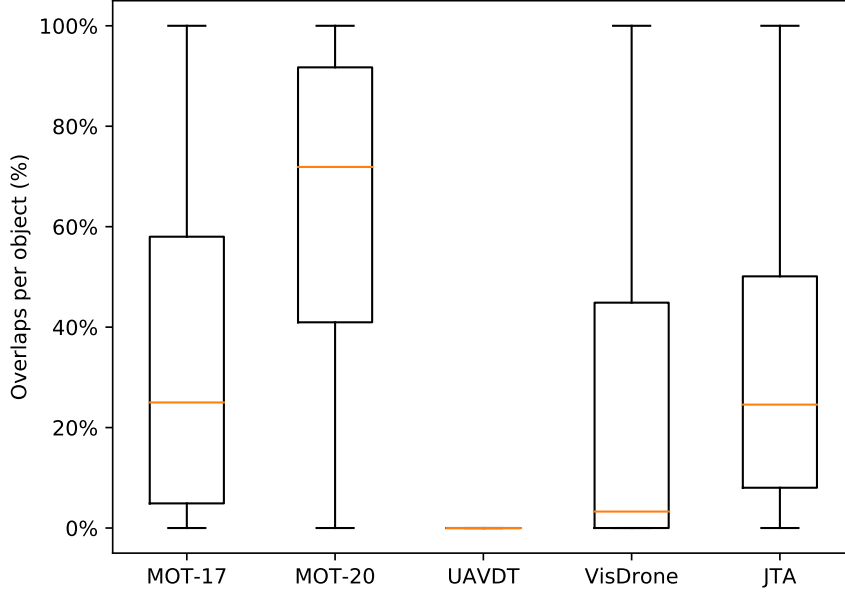


Figure 3.8: Box plot of the amount of overlap per object for the analyzed databases. An object in a frame is regarded as overlapped if more than 50% of its area is shared with other bounding boxes.

Overall, the proposed components improve the baseline by an average of +8.4 points in accuracy and +6.8 points in robustness. The components that comprise the proposal engine add +4.6 points in accuracy and +5.8 points in robustness. The comparison head is responsible for the increase of +3.8 and +1.0 points in accuracy and robustness, respectively.

These contributions not only improve the tracking quality, but they also preserve the efficiency of the algorithm. The computational complexity of SiamMOTION is of $O(WH)$ for the feature extractor, $O(WH + N)$ for the RoI extractor, $O(N)$ for the inertia module, $O(N)$ for the attention module, $O(KN)$ for the pairwise-depthwise-RPN, and $O(N^2)$ for the multi-object penalization. All this yields a computational complexity of $O(WH + KN + N^2)$ for the end-to-end network. If we take into account that $W \simeq H \gg N$, then the complexity is dominated by the size of the input frame.

Lastly, although the proposed architecture is already quite lightweight, it can be further simplified for deployment in industrial environments with constrained resources. The core of the proposal engine is the region-of-interest extractor (**F**), which leverages a feature pyramid network to yield features with richer semantic information. On the other hand, the main component of the comparison head is the similarity operator (**R**), which exploits a pairwise cross-correlation to fuse the exemplar and search area tensors while scaling seamlessly with the number of targets. This configuration (Table 3.1, row “**B+F+R**”), while simple, still outperforms the state of the art in all of the tested benchmarks but MOT-2020 [35] by an average of +2.7 points in accuracy and +2.9 points in robustness. Nonetheless, the additional methods introduced in the paper are still beneficial for achieving superior performance, especially in crowded environments such as MOT-2020.

	MOT-17				MOT-20				UAVDT				VisDrone				JTA				
	@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps		#ob
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	
SiamMT [145]	54.5	76.2	54.5	73.8	52.6	81.7	52.5	82.2	54.6	92.3	54.5	92.3	45.7	52.2	45.4	51.5	46.9	61.5	46.9	61.5	>99
SiamBAN [20]	45.9	59.9	44.2	58.4	27.5	67.4	26.7	66.9	38.2	67.1	35.1	62.3	26.8	29.7	27.3	29.2	36.3	48.5	34.8	46.8	44
STMTrack [48]	48.7	61.8	46.6	59.9	27.3	67.3	26.9	66.9	35.6	62.8	33.4	59.2	28.1	29.5	27.0	29.2	38.2	49.5	36.6	47.6	23
SiamAttn [170]	45.0	57.3	43.9	56.1	26.7	66.7	26.3	66.5	35.0	60.9	32.7	57.4	26.8	29.0	26.2	28.7	35.0	45.7	33.9	44.7	21
SiamFC++ [161]	53.9	70.2	52.3	67.6	30.4	70.9	28.8	69.7	52.4	86.1	48.5	81.2	34.2	32.8	32.0	31.5	44.4	59.8	42.1	56.2	30
SiamCAR [55]	47.5	60.0	46.0	58.2	27.8	67.5	27.2	67.0	40.3	70.0	37.1	64.9	28.9	29.8	27.8	29.3	38.3	48.7	36.6	46.9	45
SiamRPN++ [82]	48.5	60.1	47.0	58.9	27.8	67.4	27.1	66.9	38.2	67.1	35.2	62.5	28.2	29.6	27.2	29.1	37.8	48.7	36.1	46.9	22
DaSiamRPN [188]	48.2	72.5	46.9	69.3	31.2	73.0	29.5	71.4	52.0	89.5	48.6	85.4	35.3	34.7	33.7	33.1	42.0	61.6	40.54	58.8	33
SiamRPN [83]	50.9	70.7	49.5	67.9	30.3	71.0	29.1	69.7	53.5	89.9	49.8	85.7	36.2	34.4	33.9	32.8	43.7	60.3	41.5	56.8	33
SiamFC [4]	45.8	58.4	45.2	57.2	26.9	67.3	26.6	66.9	41.4	73.3	37.5	67.3	27.9	29.5	27.0	29.0	36.3	46.3	35.2	45.1	39
SiamMOTION	57.9	77.2	57.8	76.7	52.7	82.2	52.7	82.2	57.8	93.3	57.7	93.4	50.9	59.8	49.2	57.6	51.0	66.3	51.0	66.3	>99

Table 3.2: VOT-RT metrics results. Red, blue and green, represent 1st, 2nd and 3rd respectively. Column #ob shows the maximum number of targets that can be instantiated before the tracker resorts to GPU over-subscription.

3.4.3 Comparison With the State of the Art

Since SiamMOTION is an MVOT, it constitutes a dedicated component that performs a specific task within the MOT framework —motion estimation when there are no detections available. Therefore, in this section, we compare it with current MVOT solutions on the previously discussed databases, employing VOTChallenge’s VOT-RT metric @20fps and @25fps [76]. Specifically, this comparison covers SiamFC [4] —used in UMA [166]—, SiamRPN [83] —used in both DeepMOT [160] and DAMOT [185]—, and SiamRPN++ [82] —used in SiamMOT [126]—, as well as SiamMT [145] and current state-of-the-art single-object trackers that can be tailored for MVOT —DaSiamRPN [188], SiamCAR [55], SiamFC++ [161], SiamAttn [170], STMTrack [48], and SiamBAN [20]. The results for each algorithm are shown in Table 3.2. All of these MVOTs have been run on multiple-object environments —as the aforementioned papers do— and maximizing GPU usage. Column #ob shows the maximum number of targets that can be instantiated before GPU memory oversubscription.

According to the experimental results, SiamMOTION outperforms previous state-of-the-art MVOT approaches. These improvements are in both accuracy and robustness, resulting in fewer identity switches and better delimiting the tracked objects. Some approaches such as SiamFC++ are able to fit the bounding boxes to the objects very tightly, but have problems maintaining their identities for several frames —i.e., high accuracy but low robustness. Conversely, other approaches such as DaSiamRPN exhibit the opposite behavior. These, while correctly identifying the center of the targets, report a bounding box with an incorrect size often vaguely related with the objects —i.e., high robustness, but low accuracy. The latter, while certainly not ideal, may be acceptable in those scenarios in which the boundaries of the objects are not a critical factor, but the main objective is to maintain their identities.

Owing to the global frame features extraction and the use of a specialized similarity operator ($\star$), SiamMT fares better than its predecessors. Consequently, our approach SiamMOTION

exploits these breakthroughs and builds on them to create a novel architecture. Thus, the global features extraction allows to share the backbone computations —the most expensive operation in the network—, and we incorporate the $\tilde{\times}$ operator as the core of our pairwise-depthwise-RPN. This, together with the enhanced RoI extractor, the multi-object penalization module, and the inertia system, gives SiamMOTION advantages of up to +5.2 points of accuracy and +7.6 points of robustness compared to its predecessor. When compared to the second best state-of-the-art architecture, these differences can become as large as +23.2 and +25.1 points, respectively.

One of the datasets in which SiamMOTION best performs is UAVDT —improvement of +3.2 points in accuracy and +1.0 points in robustness @20 fps. This is because it contains a large number of small moving vehicles, for which the inertia module and the inclusion of the FPN in the RoI extractor are ideal. The inertia module allows for better placement of the search area when there are fast movements —this is critical for small objects, as the field-of-view of the network is smaller for them (Equation 3.6)—, while the FPN provides meaningful features for small objects, which would normally be washed-out at deeper levels of the backbone. Other datasets in which SiamMOTION excels are VisDrone —improvement of +5.2 points in accuracy and +7.6 points in robustness @20 fps— and JTA —improvement of +4.1 points in accuracy and +4.8 points in robustness @20 fps. The large number of objects they contain and the fact that the sequences are captured with a moving camera mean that the bounding boxes of the targets will continuously change in aspect ratio. Thanks to SiamMOTION’s built-in PD-RPN, we are able to detect these changes in an efficient and effective manner.

On the most challenging dataset, MOT-2020, SiamMOTION achieves the highest difference in accuracy and robustness with the state of the art but SiamMT, surpassing them by over +21.5 points in accuracy and +9.2 points in robustness @20 fps. This dataset is quite complex due to the large amount of overlap between objects and erratic movements it contains, making it completely different from the rest of the benchmarks —the initializations themselves often contain parts of other tracked objects. In fact, Figure 3.8 shows how half of the objects in MOT-2020 present overlaps in 70% or more of their detections. This statistic is very different from the rest of the databases, in which most of their objects are overlapped in less than 25% of their detections. An extreme example of this is UAVDT, where 90% of its objects have an overlap below 3%, since it consists of zenithal recordings of vehicles.

3.5 CONCLUSIONS AND FUTURE WORK

We have presented SiamMOTION, an MVOT (multiple visual object tracker) capable of tracking several dozens of objects in real-time with high accuracy and robustness, regardless of their category and size. This is made possible thanks to a proposal engine that generates quality features —well-framed, with the correct resolution, and highlighting the most relevant channels for each object— and a comparison head that efficiently outputs quality predictions —detecting

changes in aspect ratio and suppressing the effect of distractors, all without resorting to multi-scale testing.

SiamMOTION has been evaluated on various video databases, achieving a real-time performance that surpasses the current state of the art —increase of +2.9 points in both average accuracy and average robustness when compared to the best performing counterpart at 25 fps, and differences of more than +11.6 points when compared to the rest. Furthermore, an exhaustive ablation study was carried out on all the tested databases, analyzing the contribution of each new component. This showed that the parts comprising the proposal engine —inertia module, RoI extractor, and attention mechanism— are responsible for the 68% of the improvement brought by the architecture —+4.6 points in accuracy and +5.8 points in robustness over the baseline—, while the components that make up the comparison head —Pairwise Depthwise RPN and multi-object penalization module— contribute the remaining 32% —+3.8 and +1.0 points in accuracy and robustness, respectively.

As SiamMOTION is an MVOT and not a fully-fledged MOT system, it solves a specific task within the MOT framework —motion estimation when there are no detections available. It is therefore not responsible for carrying out other tasks such as track initialization and termination, or drift detection. This is why as future work it would be interesting to develop a complete MOT system integrating SiamMOTION as well as an object detector, an affinity estimator, and an association mechanism; making it fully integrated and end-to-end trainable. This would most likely allow for better learning for all components, as well as ease its deployment in resource-constrained environments. It would also be interesting to transform SiamMOTION to produce more thorough outputs, enabling rotated bounding boxes or per-pixel identification —segmentation. This would allow for more refined predictions, making tracking more valuable in those situations where objects have unusual shapes or appear in large crowds. However, this is not straightforward, as such methods usually have a severe impact on the speed of the system, preventing it from running in real-time.

4 Beyond Missed Detections: Leveraging Unmatched Trajectories for Enhanced Multi-Object Tracking

In Chapters 2 and 3, we tackled scenarios in which detections are not available, leading to the development of motion estimation systems that enabled multi-object tracking. However, there are other situations where, although detections can be obtained for every frame, certain objects may be overlooked, resulting in numerous untracked objects. In the present chapter, we address this additional aspect of object tracking and explore how to recover such false negatives without resorting to offline solutions or having to re-consider the detector employed.

To this end, this chapter introduces a generalization of the simple-yet-effective BYTE association method, enabling its operation in situations without detections. Specifically, we focus on tracks without matching detections and keep them alive through their association with candidates that are proposed using a motion model. The association between proposals and tracks is performed using ByteFormer, our innovative Transformer-based architecture that exploits both motion and appearance cues. Our fully online system exhibits solid performance, surpassing prior state-of-the-art techniques and showcasing its potential for a wide variety of multi-object tracking applications.

This chapter is organized as follows. In Section 4.1, we discuss the potential of motion-based proposals for recovering false negatives and emphasize the contributions brought forth by ByteFormer and our generalization of BYTE. Section 4.2 reviews the current state-of-the-art in multi-object tracking and highlights its limitations. The extension of BYTE and the ByteFormer architecture are elaborated in Section 4.3. Section 4.4 presents the experimental results. Finally, Section 4.5 offers concluding remarks on the proposed methodology and suggests possible avenues for future research.

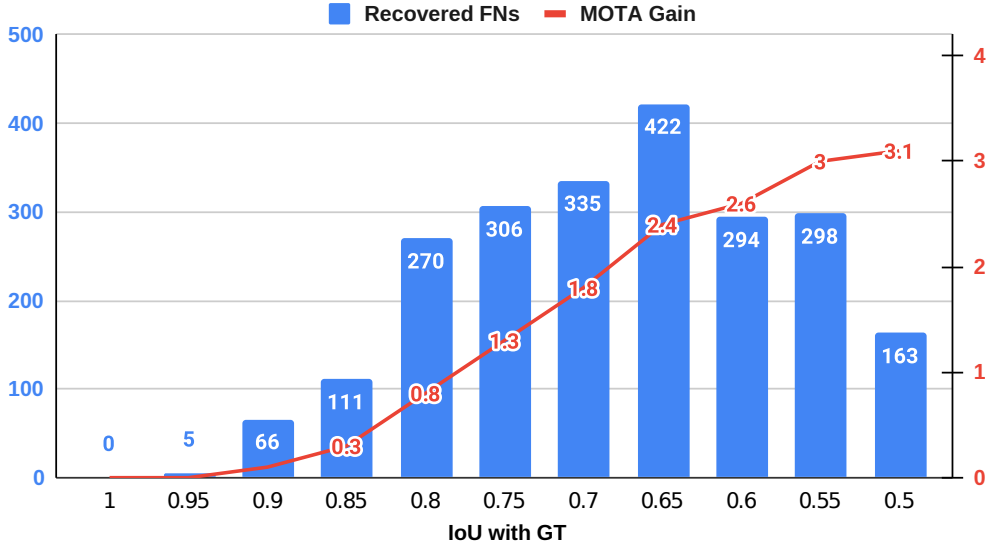


Figure 4.1: Impact on MOTA when successful motion-based predictions without an associated detection are reported. The more lenient is the IoU to consider a prediction correct, the more FNs are recovered. Measured on MOT17-val [34] using [174] as a baseline.

4.1 INTRODUCTION

Multi-Object Tracking (MOT) is a fundamental problem in computer vision with applications spanning various domains such as robotics [49], autonomous vehicles [56], and video surveillance systems [117]. MOT involves the identification and continuous tracking of multiple objects of interest within a scene over time. The prevalent approach to tackle this task is tracking-by-detection [24], which comprises two key steps: (i) detecting objects within the scene and (ii) associating these detections to persistent identities to assemble tracks. The strength of these types of solutions lies in the fact that they decouple the problem, allowing far more flexibility than end-to-end models and enabling the leverage of heuristic knowledge.

Recent studies [174, 125] have demonstrated that straightforward frame-by-frame matching techniques, as simple as the intersection over union (IoU) between tracks and detections or the cosine distance between their appearance, can yield state-of-the-art results. Consequently, nowadays there is a big focus on obtaining accurate proposals for all the objects within the scene. Maximizing the recall of current object detectors is not straightforward and naïve approaches often lead to a large number of false positives, which significantly hampers tracking performance. Therefore, current solutions often propose increasingly convoluted pedestrian detection pipelines [91] or offline systems that rely on future frames to recover objects under severe occlusions [181].

However, we argue that it is not necessary to scout the entire frame or resort to offline methods to retrieve such objects. If a motion model can associate proposals, it can also recover missing detections as long as the object maintains its trajectory. In fact, with an oracle for determining when to kill a motion-based trajectory, we can save up to 2,270 false negatives out of the 9,090 currently missed by [174] in MOT17 [34], as shown in Figure 4.1.

Consequently, we propose an intuitive and effective method to recover false negatives in MOT. Following the guidelines outlined in BYTE [174], our approach emphasizes the importance of exploiting all the information available during tracking. In particular, we generalize the association method to also focus on the frequently neglected mismatched trajectories. This allows us to propose new motion-based candidates in situations where no detections are available. Lastly, to determine whether a candidate belongs to a track, we develop a mechanism that is robust against heavy occlusions and loosely defined bounding boxes.

To this end, we introduce ByteFormer, an online Transformer-based [146] architecture designed for matching candidates and tracks by leveraging both visual and spatial information. Contrary to other methods [125, 143], ByteFormer does not require postprocessing nor relies on hand-crafted rules for determining which information to prioritize. Instead, its spatiotemporal encoder enables it to process multimodal inputs and learn sophisticated relationships between them. ByteFormer is able to process the full information of a track, which allows it to pay more attention to the most relevant observations on a case-by-case basis. This, along with its use of contextual information, enables it to effectively recover many false negatives missed by the detector, all without conducting an exhaustive analysis of the entire frame or resorting to offline methods. The main contributions of this work are as follows:

- We generalize the association mechanism —i.e., BYTE [174]— to address situations with no matching detections. This enables us to recover a significant number of objects that were overlooked by the detector.
- We propose ByteFormer, a Transformer-based [146] architecture that harnesses multiple-choice question-answering techniques to associate motion-based proposals to tracks.
- We introduce a novel spatiotemporal encoding that allows learning complex relationships between motion and visual features.
- We demonstrate the effectiveness of ByteFormer through extensive experiments and comparisons with state-of-the-art MOT algorithms, showcasing its potential for real-world applications.

4.2 RELATED WORK

4.2.1 End-to-End Detection-and-Tracking

Over the past few years, leveraging the growth in computational capabilities, researchers have increasingly focused on learning as much as possible from large-scale datasets [104, 33, 35, 34, 41] in the multi-object tracking task. These approaches provide little to no prior information to the algorithm and seek to solve the problem in an end-to-end manner [155, 175, 132, 1, 130, 103, 173]. The cornerstone of this trend were convolutional neural networks (CNNs) capable of simultaneously detecting and identifying objects within a frame [155, 175]. Following this, the

temporal dimension was introduced to better capture object dynamics by processing multiple frames at once using 3D convolutions and then stitching the clips together [132, 1].

More recently, with the advent of Transformer architectures [146], the field has shifted towards DETR-inspired [16] detection-and-tracking methods. These models incorporate track queries that propagate through time, linking bounding boxes and capturing intricate temporal relationships [130, 103]. Remarkably, some solutions can even eliminate the need for post-processing, as the models learn complex temporal interactions via collective aggregation losses that span multiple frames [173]. The benefit of these end-to-end approaches is that they do not require any heuristic knowledge about the problem. But they are highly data-hungry and require large sets of labeled samples, which are scarce for some scenarios. This is why tracking by detection is the most common paradigm currently used in MOT.

4.2.2 Tracking by Detection

Tracking by detection is currently the most popular approach for solving multi-object tracking (MOT) problems [3, 179, 182, 57, 184, 174, 125]. This class of methods separates the tracking problem into two stages: object detection and data association. This has the benefit of allowing both fields to evolve independently and also to collaborate at the same time. It is therefore common to find classical detectors [119, 121] as the basis for some of these methods, but generally what offers better overall results is to use more modern networks [174] or detectors tailored to a specific category [70]. After this first detection step, proposals are associated with tracks, which can be performed offline via hierarchical clustering [179], attention mechanisms [184], or graph neural networks [57].

However, the most versatile methods are usually the online ones, which process the video stream without having access to frames in the future. There are some works that tackle this by taking advantage of the detector itself with no need to learn additional tasks, either by exploiting the regression heads [44] or through distance-based algorithms for the object centers [182]. The issue with these mechanisms is that they do not have a true notion of identity and are very susceptible to large displacements and thus have to be complemented with motion models and re-identification mechanisms [3]. This is why associations among detections are commonly modeled with methods decoupled from the detector.

Motion-Based Association Taking advantage of the fact that object displacements are generally minimal between adjacent frames, motion-based methods employ location information from detections and pre-existing tracks for frame-by-frame association [5, 9, 174]. These methods can be as simple as computing the associations using the intersection over union between detections [9] or using some motion model along with the overlap to make the tracks more robust to temporal gaps [5]. Afterward, detections are matched to tracks by minimizing the overall cost of the associations, with techniques such as the Hungarian algorithm [78].

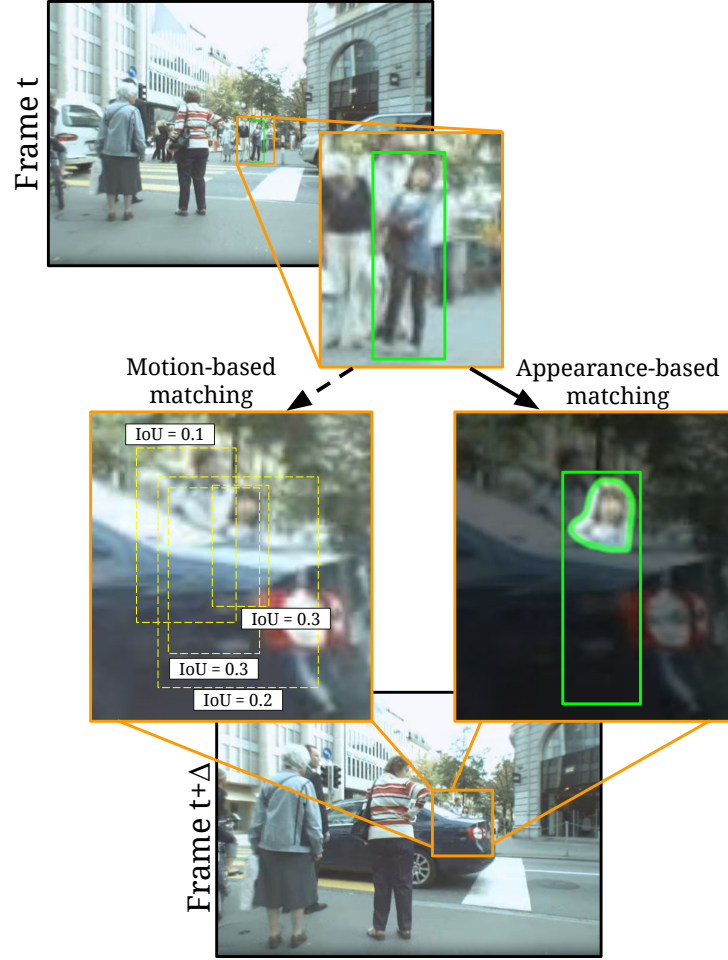


Figure 4.2: Comparison between motion-based and appearance-based associations. Counter-intuitively, during low visibility situations, visual features may pose essential to correctly identify an object, as bounding-box boundaries are highly unreliable. Sequence extracted from the MOT17 dataset [34].

A problem also exists in determining which detections are factored into this association and which ones are not. Instead of simply establishing a confidence threshold and discarding the remaining bounding-boxes, [174] considers all of them in different rounds of associations, achieving solid state-of-the-art results with a comprehensive model based on Kalman filters [68]. However, while the best results are obtained by leveraging all detector outputs, still some objects are not correctly detected, and pure motion-based methods struggle in crowded and moving-camera scenarios.

Appearance-Based Association In order to maintain identities over multiple frames and provide robustness in scenarios with heavy occlusions, where bounding-box boundaries cannot be reliably estimated (Figure 4.2), some methods resort to appearance-based associations [156, 72, 126, 145, 125, 143]. This can be achieved simply by integrating a re-identification network and building associations from the inter-feature distances [156, 72]. Alternatively, there are methods that rely on single-object tracking techniques and use Siamese networks to associate pairs of proposals [126, 145].

However, there are situations in which visual information is not so reliable due to blur or full occlusions, hence some works propose hybrid approaches that tackle this by simultaneously integrating appearance and motion [125, 143]. This improves tracking, but these techniques rely on handcrafted rules, which does not allow the network to model relationships between motion and appearance. In order to tackle this issues, we propose ByteFormer, which fuses spatiotemporal and appearance information to effectively learn the intricate relationships between them, allowing the model to capture complex interactions between object proposals and tracks. This innovative method stands out from past techniques as it does not involve any aggregation of temporal data. Instead, it maintains a comprehensive memory of the past positions and appearances of each object, allowing the model to have a more complete understanding of the object’s trajectory and characteristics throughout the tracking process.

4.3 METHODOLOGY

In the following sections, we present our proposal for false negative recovery. This novel method extends BYTE to also consider those tracks without corresponding detections. The new proposals are obtained using a motion model and they are associated with tracks using ByteFormer, a novel architecture that incorporates a spatiotemporal encoder to aggregate multimodal inputs.

4.3.1 Associations Beyond Detections

There are numerous techniques for associating detections and tracks, and BYTE [174] has demonstrated its prowess as one of the most potent methods in modern multi-object tracking. Its appeal lies not only in the accuracy it offers but also in its simplicity and generality, making it applicable to a vast array of scenarios. The philosophy behind BYTE is to take advantage of all detector outputs, rather than relying only on high-confidence detections. Thereby, it comprises two rounds of associations, which are shown in the upper part of Figure 4.3.

The input to the association mechanism is a set $\mathcal{T} = \{\tau_1, \dots, \tau_N\}$ of active tracks, and a set $\mathcal{D} = \{d_1, \dots, d_M\}$ of detections at frame F . Tracks are represented by their estimated future bounding-box $\tau_i = \{\hat{b}_i\}$ and detections are defined by their coordinates and confidence score $d_j = \{b_j, s_j\}$. Detections are partitioned based on s_j using a threshold σ , yielding sets $\mathcal{D}_H$ and $\mathcal{D}_L$ composed of high and low confidence detections, respectively. For the first round of associations, elements of $\mathcal{D}_H$ and $\mathcal{T}$ are associated based on their intersection over union (IoU) through the Hungarian Algorithm [78], requiring a minimum overlap of ϵ_1 . Tracks associated with a detection are kept while the remaining ones $\mathcal{T}_2$ move to a second round that matches them with $\mathcal{D}_L$. In this second round, the minimum overlap ϵ_2 is stricter as detections are less reliable and are prone to false positives. Lastly, successfully associated tracks are updated with the detection coordinates, whereas those without a matching detection are deactivated.

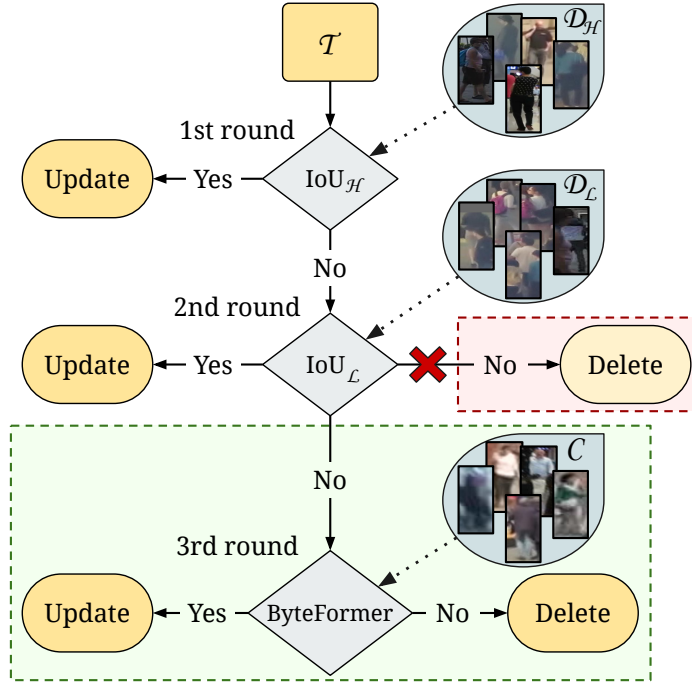


Figure 4.3: Algorithm for updating a set of tracks $\mathcal{T}$. BYTE [174] executes two rounds of associations with detector proposals. In our proposed architecture we modify the second round —highlighted in red— and introduce a third round of associations —highlighted in green— to mitigate false negatives by incorporating motion-based candidates into the tracking process.

While BYTE is able to recover numerous objects that otherwise would have been dismissed as false negatives, we believe there is still room for further improvement. Great emphasis is often placed on how to recognize as many identities as possible in a scene, but the decision on whether to terminate them is often overlooked and frequently reduced to a single hyperparameter. We argue that more care should be taken when eliminating tracks, avoiding premature terminations of objects that genuinely exist. To this end, we propose a third round of associations that does not rely on detections, as shown in the lower part of Figure 4.3.

Unsuccessfully matched tracks are evaluated on a new round of associations using our ByteFormer algorithm, which is further explained in Section 4.3.2. In this stage, we can no longer define criteria based on confidence scores since proposals are not detections. Furthermore, relying exclusively on motion-based cues or using just the latest detection for each track is ill-suited, as in crowded scenarios they will probably not be truly representative of the identity of the object.

Thus, in order to retain as much information as possible, tracks on this third round $\mathcal{T}_3$ are represented by a time-ordered set of their appearances and coordinates $\tau_i = \{\{a_{i_1}, b_{i_1}\}, \dots, \{a_{i_Z}, b_{i_Z}\}\}$, where Z is the length of the track and the appearance is extracted using a CNN. Similarly, the new candidates $C = \{c_1, \dots, c_P\}$ also comprise appearance and coordinates pairs $c_k = \{a_k, b_k\}$, to better guide the association. These candidates are proposed according to a motion model as described in Section 4.3.4.

As such, our generalization of BYTE beyond detections enforces BYTE to reconsider tracks that would otherwise have been discarded. If ByteFormer determines the suitability of a candidate, the corresponding track will be updated, reducing the number of false negatives. Otherwise, the track will be deactivated for its potential rebirth [5]. Finally, one of the advantages of this method is that it does not require postprocessing —e.g. Hungarian algorithm [78]— since ByteFormer already generates one-to-one matches.

4.3.2 ByteFormer

Our association mechanism has to effectively associate tracks and candidates. It has to be flexible in terms of the number of candidates and the number of available samples per track. At the same time, it must be able to model complex short-and-long-term relationships between tracks and candidates. With these requirements in mind, we opted for a Transformer-based architecture as the basis of ByteFormer, whose overview can be found in Figure 4.4.

Specifically, drawing parallels with the natural language processing (NLP) field, our task can be posed as a multiple-choice question-answering type of problem [115]. Thus, given a question —the track— and given a set of possible options —the candidates—, the goal of the network is to find the correct answer —the matching candidate. However, in our particular case, each element of the input is represented by its appearance and coordinates. As this information is very different in nature, it is not sensible to directly concatenate it into the same vector. Thus, before entering the ByteFormer encoder, this multimodal input is fused using the spatiotemporal encoding detailed in Section 4.3.3.

In contrast to conventional multiple-choice question-answering systems, which duplicate the Transformer for each different answer, our approach processes all candidates simultaneously. This not only enhances efficiency but also enables more effective modeling of the relationships among scene elements. Consequently, the encoder’s cross-attention mechanisms are able to consider a multitude of factors, including appearance, motion patterns, and spatial relationships between all the objects.

We also include additional contextual tokens that correspond to nearby detections that have already been associated. Their primary purpose is to provide ByteFormer with more information about the scene and to help it deduce if it is dealing with a duplicate track that should not be kept alive, despite its similarities with a candidate. The latter is especially critical, as sometimes duplicate detections are not suppressed, leading to multiple identities for the same object. To alleviate this, we also include a learned classification token [CLS], which checks for inconsistencies in the track memory and encodes whether it is corrupted with detections belonging to other objects.

It is also possible that the track has left the scene or the proposed candidates are not good enough. Thus, an additional learned token [MISS] is also included to give the Transformer

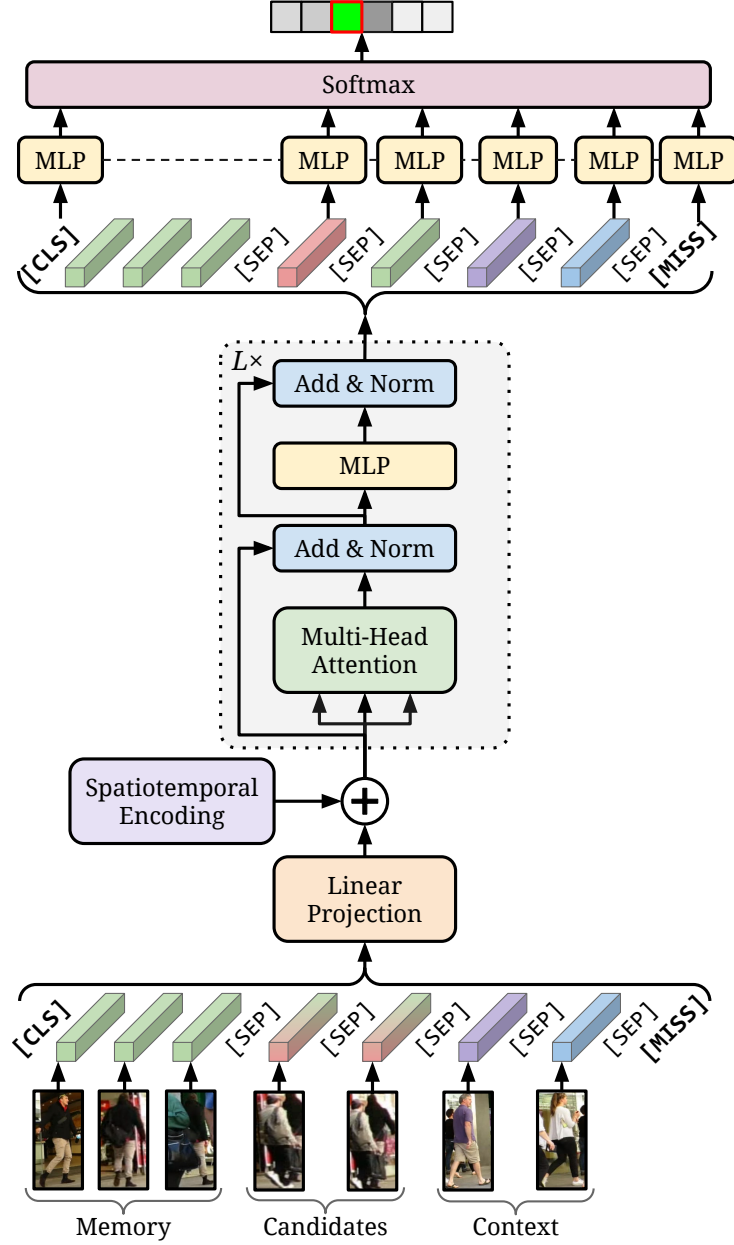


Figure 4.4: ByteFormer architecture. The input consists of the memory of a track, one or more candidates, and nearby contextual objects. Relationships between these elements are captured through L Transformer Encoder layers. The output determines whether the memory represents a valid object –indicated by the [CLS] token– and identifies which proposal belongs to the track, if any –the latter is represented by the [MISS] token.

the possibility to convey this information. A learned separator token [SEP] is also included to delimit the memory and the different candidates.

After cross-attending the different components of the input, the encoder yields a series of refined tokens, promoting the features of those candidates most closely related to the track. These refined elements are fed to a multi-layer perceptron (MLP) head that generates one logit per candidate. These logits are eventually merged through a Softmax which outputs the probability that each candidate belongs to the track. It should be noted that it is critical that the MLP head shares its weights, so as not to be restricted to a fixed input size.

4.3.3 Spatiotemporal Encoding

The representation of the objects has to capture both appearance and coordinates information, allowing ByteFormer to learn the nuances of object tracking effectively. To aggregate multi-modal inputs, the standard approach in NLP involves utilizing a positional encoding for each of the input tokens to represent their position in the sentence. However, here we want to represent more complex relationships between the tokens. Consequently, within ByteFormer, we adopt a spatiotemporal encoding that captures the three most relevant dimensions for motion-based tracking: time, size, and distance. These spatiotemporal cues are merged with the appearance features, assisting ByteFormer in accurately associating tracks and candidates under challenging circumstances.

The encodings for these three dimensions cannot be defined separately, as replicating the same sinusoidal function would generate potentially ambiguous information [154]. Therefore, it is necessary to define a joint spatiotemporal encoding STE by generalizing the function used in [146] to a 3-dimensional space:

$$\begin{aligned}
STE_{(E^t, E^s, E^d, 2i)} &= \sin\left(E^t / 10000^{6i/D_m}\right) \\
STE_{(E^t, E^s, E^d, 2i+1)} &= \cos\left(E^t / 10000^{6i/D_m}\right) \\
STE_{(E^t, E^s, E^d, 2j+D_m/3)} &= \sin\left(E^s / 10000^{6j/D_m}\right) \\
STE_{(E^t, E^s, E^d, 2j+1+D_m/3)} &= \cos\left(E^s / 10000^{6j/D_m}\right) \\
STE_{(E^t, E^s, E^d, 2k+2D_m/3)} &= \sin\left(E^d / 10000^{6k/D_m}\right) \\
STE_{(E^t, E^s, E^d, 2k+1+2D_m/3)} &= \cos\left(E^d / 10000^{6k/D_m}\right)
\end{aligned} \tag{4.1}$$

where (E^t, E^s, E^d) are the coordinates in the 3D space and $i, j, k \in [0, D_m/6]$ is the dimension within the D_m channels of the model. Thus, the spatiotemporal encoding for any candidate c can be obtained from its position $STE_{(E_c^t, E_c^s, E_c^d)}$.

We now need to define a method to distribute the different elements of the video in our 3D spatiotemporal encoding. It is desirable that the same interaction between a group of objects is always represented in the same manner, regardless of the moment of the sequence when it occurs or their absolute position in the frame. This is why we encode these dimensions relative to an anchor A_i , which in our case is the last bounding-box of the track τ_i at time t : $b_{i_t} = \{x_{i_t}, y_{i_t}, w_{i_t}, h_{i_t}\}$, where x and y are the coordinates of the center of the object, and w and h are its width and height, respectively. This boosts the generalization capacity of ByteFormer and improves convergence.

Therefore, for each candidate c , we have to find the coordinates for time, size, and distance (E_c^t, E_c^s, E_c^d) that will allow us to retrieve its spatiotemporal encoding. This process is illustrated in Figure 4.5, and each one of the dimensions will be encoded in a different manner.

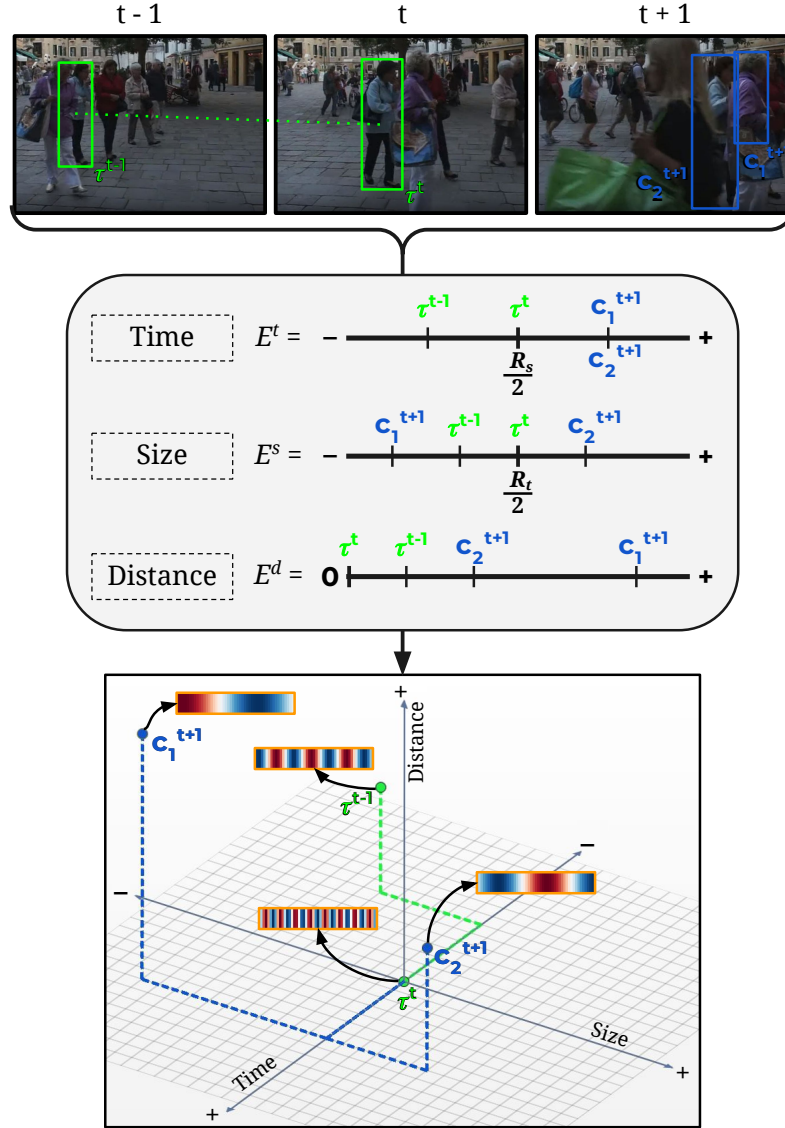


Figure 4.5: Generation of the spatiotemporal embedding —outlined in orange— for each element $\{\tau^{t-1}, \tau^t, c_1^{t+1}, c_2^{t+1}\}$ in the sequence. For each bounding box, its time E^t , size E^s , and distance E^d with respect to the anchor τ^t are projected into a 3D space. These coordinates are then used to compute the vector from STE .

Time Dimension. It is important to provide ByteFormer with information about the instant at which each bounding box was recorded. This allows it not only to sort the observations that comprise the track but also to determine whether it is appropriate to match a candidate after a large time gap. To this end, we consider our reference point as the instant t at which the last known position of the track was reported. Assuming that for the time dimension we have a representation range R_t , t will therefore be located at $R_t/2$. Consequently, older bounding boxes will take lower indices for the time encoding, and candidates c_k where $k > t$ will take higher values. The time index for a candidate c will therefore be computed as:

$$E_c^t = \rho^t (k - t) + \frac{R_t}{2} \quad (4.2)$$

with ρ^t as a scaling factor. This fosters a unique condition that does not occur with traditional encodings, as some tokens—the candidates—will have identical temporal encodings since they belong to the same moment in time. This property is extremely important for our architecture, as despite taking different positions in the input, we want ByteFormer to assume that all the candidates are at the same temporal distance.

Size Dimension. To improve decision-making, ByteFormer must also know the size of the objects to prioritize the associations with bounding boxes of similar size. Similarly to the time dimension, for a representation range R_s , A_i will be the origin point and will be located at $R_s/2$. We model this relation through a log-scale transformation, in a similar manner as [54] parametrizes width and height regressions. Thus, all the smaller bounding boxes will be located to the left, while those that are larger will be distributed in the opposite direction in the size representation space:

$$E_c^s = \rho^s \left(\log \left(\frac{w_c}{w_{A_i}} \right) + \log \left(\frac{h_c}{h_{A_i}} \right) \right) + \frac{R_s}{2} \quad (4.3)$$

with ρ^s as a scaling factor.

Distance Dimension. Finally, distance enables ByteFormer to easily rule out potential matches with extremely distant candidates. To this end, we encode the magnitude of the distance relative to the center of the object, defining the origin point at 0. However, we do not directly employ the Euclidean distance but weigh it by the size of the anchor. The coordinate for this encoding is computed as:

$$E_c^d = \rho^d \log \sqrt{\left(\frac{x_c - x_{A_i}}{w_{A_i}} \right)^2 + \left(\frac{y_c - y_{A_i}}{h_{A_i}} \right)^2} \quad (4.4)$$

with ρ^d as a scaling factor. Finally, for a candidate c , its spatiotemporal encoding will be the value of the coordinate (E_c^t, E_c^s, E_c^d) in *STE*.

Special Tokens. Learned tokens —i.e. [CLS], [MISS], [SEP]— also need an encoding. Intuitively, [SEP] is given the same encoding as the candidate it accompanies, which prevents all [SEP] tokens from having the same spatiotemporal encoding. [MISS] is given the same encoding as the anchor A_i , signaling that it is the termination of the track. And [CLS], which is used to indicate that the track contains corrupt information, is encoded by maximizing the distances in the representation space.

4.3.4 Proposal Generation

An important part of ByteFormer is the selection of the candidates and the context that will be fed as input to the encoder. We believe that simple-yet-effective solutions are the most versatile, and if a linear motion model is adequate to associate detections, it will also be well suited to propose new candidates. Thus, we use a Kalman filter [68] with camera motion compensation to predict the location of tracks in future frames and obtain new detection-independent candidates.

For context, we select tokens that enclose representative information about the environment that the track inhabits. Given the noisy information and severe occlusions often present in the third round of associations, it is necessary to leverage as much contextual information as possible. It is also desirable that they capture information similar to that of the track and serve as distractors, such that ByteFormer does not choose a “good enough” candidate, but rather picks the most representative element. This poses particularly beneficial for preventing the persistence of duplicate tracks. Thus, the context tokens will correspond to Q already matched first-and-second-round detections within the vicinity V_τ of the track. This vicinity is defined in terms of the Euclidean distance and is weighted by the size difference:

$$\text{Neigh}(\tau) = \{d \in \mathcal{D} \mid \text{Eucl}(\tau, d) * \text{Ratio}(\tau, d) < V_\tau\} \quad (4.5)$$

$$\text{Eucl}(\tau, d) = \sqrt{(x_\tau - x_d)^2 + (y_\tau - y_d)^2} \quad (4.6)$$

$$\text{Ratio}(\tau, d) = \max\left(\frac{\sqrt{w_\tau * h_\tau}}{\sqrt{w_d * h_d}}, \frac{\sqrt{w_d * h_d}}{\sqrt{w_\tau * h_\tau}}\right) \quad (4.7)$$

$$V_\tau^2 = (w_\tau + \zeta(w_\tau + h_\tau)) \times (h_\tau + \zeta(w_\tau + h_\tau)) \quad (4.8)$$

where ζ is a context factor. If there are more than Q elements that meet this condition, the closest ones will be chosen.

4.4 EXPERIMENTAL RESULTS

Experiments were conducted using PyTorch on a computing node with an Intel Xeon 4114 CPU, 96 GB of DDR4 RAM, and an NVIDIA Titan RTX GPU.

4.4.1 Implementation Details

For our detector, we use a YOLOX [51] network trained following [174], and set the detection confidence and overlap thresholds to $\sigma = 0.6$, $\epsilon_1 = 0.1$, and $\epsilon_2 = 0.5$. For the appearance extraction, we resize proposals to 128×384 px and extract 512-element embeddings using a ResNet-50 [59] with domain adaptation as described in [125]. Our encoder uses a 512 token size, with $L = 4$, 4-headed multiattention, $Q = 4$, $P = 1$, and 1024-size feedforward layers. To define the vicinity of a track we use $\zeta = 1.0$ and our spatiotemporal encoding has a maximum representation range of $R_t = 60$, $R_s = 210$, and $R_d = 210$, with $\rho^t = 2$, $\rho^s = 15$, and $\rho^d = 15$. When assembling the tracks that are fed to ByteFormer, in order to reduce drift we only use high-confidence detections and avoid updates when the area covered per object is bigger than 5% of the total frame. Due to hardware constraints, we are limited to a maximum memory size of $Z = 11$, so when a track has more detections we select equally-spaced samples that span its entire life.

	MOTA $\uparrow$	HOTA $\uparrow$	FN $\downarrow$	FP $\downarrow$	IDs $\downarrow$
BYTE	76.5 —	67.4 —	9120 —	3410 —	165 —
BYTE ($\epsilon_2 = 0.01$)	75.3 (−1.2)	65.6 (−1.8)	8854 (−266)	4196 (+786)	238 (+73)
BYTE + motion assoc.	75.4 (−1.1)	67.0 (−0.4)	7588 (−1532)	5493 (+2083)	162 (−3)
BYTE + appearance assoc.	76.6 (+0.1)	67.6 (+0.2)	8393 (−727)	4063 (+653)	162 (−3)
BYTE + ByteFormer	77.1 (+0.6)	67.6 (+0.2)	8326 (−794)	3889 (+479)	163 (−2)

Table 4.1: Ablation of different association approaches. The difference with the baseline is depicted next to each metric.

ByteFormer is trained only in MOTSynth [41] —no need for fine-tuning on specific training sets— via label-smoothed cross-entropy loss for 25 epochs, with a dropout regularization probability of 0.1 and using an AdamW optimizer [95] with a weight decay of 1×10^{-5} and an initial learning-rate of 2×10^{-5} that is decayed by 10 after 20 epochs. In a training sample, the positive candidate is the one that overlaps the most with the ground truth, as long as their IoU is higher than 0.7.

4.4.2 Ablation Study

In this section, we examine the effects of our proposals on tracking performance. For the evaluation, we employ the standard metrics used by the MOTChallenge [34]. Specifically, we report the MOTA score [69], which weights the number of false positives, false negatives, and identity switches, and the HOTA score [99] which balances object coverage and identity preservation [124]. For these experiments, ByteFormer was trained using just 100 MOTSynth [41] videos and validated on a subset of MOT17 [34]. Following the standard practice, we partitioned the MOT17 train sequences along the time axis and utilized the latter half as the evaluation set [182, 174, 125]. The remaining portion was not used for training.

Associations Beyond Detections. We first explore the benefits of generalizing BYTE and paying more attention to unmatched tracks —i.e. false negatives. Thus, using [174] as the baseline for this experiment, we compare our proposal to incorporate a third round with other approaches that reduce the number of false negatives (see Table 4.1). Specifically, we relax threshold ϵ_2 to associate a track during the second round and also explore different strategies to implement the additional round.

Decreasing the association threshold ϵ_2 leads to poor overall results. While BYTE recovers a significant number of false negatives, this strategy incurs a high cost by introducing +786 extra false positives, with the consequential decrease of −1.2 MOTA points. These results highlight that there are a high number of yet-to-be-discovered objects in the scene, while also indicating that naïve approaches are not adequate. Consequently, this suggests that the association needs to be performed beyond detections, as the majority of low-confidence detections prove to be unreliable. To this end, we implement an additional round of associations that leverages motion-based proposals.

CLS	MSS	STE	CTI	MOTA $\uparrow$	HOTA $\uparrow$	FN $\downarrow$	FP $\downarrow$	IDs $\downarrow$
				76.5 —	67.4 —	9120 —	3410 —	165 —
✓				75.0 (−1.5)	66.3 (−1.1)	8395 (−725)	4911 (+1501)	173 (+8)
	✓			76.4 (−0.1)	67.3 (−0.1)	8064 (−1056)	4513 (+1103)	150 (−15)
✓	✓			76.5 (0.0)	67.1 (−0.3)	8656 (−464)	3853 (+443)	164 (−1)
✓	✓	✓		76.7 (+0.2)	67.4 (0.0)	8528 (−592)	3851 (+441)	169 (+4)
✓	✓		✓	76.9 (+0.4)	67.6 (+0.2)	8387 (−733)	3884 (+474)	164 (−1)
✓	✓	✓	✓	77.1 (+0.6)	67.6 (+0.2)	8326 (−794)	3889 (+479)	163 (−2)

Table 4.2: Ablation of the different parts that integrate ByteFormer. CLS=[CLS] learned token, MSS=[MISS] learned token, STE=spatiotemporal encoding, CTI=contextual information. The difference with the baseline is depicted next to each metric.

In our third round of associations, we consider tracks without matching detections along with a number of motion-based candidates. One potential technique to assess the suitability of these newly introduced proposals consists in employing geometric-based cues as in [5] and associate a track and a candidate based on the overlap. Nonetheless, this simple method is prone to drift and produces more false positives (+2083) than the false negatives it is able to recover (+1532). Another possibility consists in leveraging visual features, as they prove useful in crowded scenarios where bounding boxes cannot be appropriately estimated (Section 4.2.2). Specifically, we follow [156] and compute the Euclidean distance between features extracted with a Re-ID network. We then associate a track with the candidate that exhibits the greatest similarity —i.e. least distance—, on the condition that a minimum threshold is met. This approach offers improvements, with MOTA and HOTA increases of +0.1 and +0.2, respectively. It confirms the benefit of employing visual cues to solve associations, but the proposal still introduces too many false positives to be considered reliable (+653).

Lastly, we utilize ByteFormer to tackle the associations between tracks and candidates. Notably, our method offers a robust solution by integrating motion cues and visual features spanning the entire track, leading to an additional reduction of −174 false positives compared to the alternative Re-ID appearance-based technique. As a result of our generalization of BYTE with the implementation of ByteFormer, we effectively attain an enhancement of +0.6 MOTA points over the baseline approach thanks to the recovery of −794 false negatives.

ByteFormer Components. ByteFormer relies on a series of design choices that ensure its proper operation and allow it to associate detections and tracks accurately. In Table 4.2, we analyze the impact of the learned [MISS] and [CLS] tokens, the spatiotemporal encoding, and the integration of contextual information. We use [174] as the baseline for this experiment.

To terminate a track, ByteFormer may indicate that the memory contains elements that do not belong to a single object by employing the [CLS] token, or that none of the candidates are suitable for association using the [MISS] token. Relying solely on the [CLS] token yields negative results, resulting in an additional +1501 false positives compared to the baseline. Conversely, if track termination is guided by the [MISS] token, the output remains marginally below the base-

line with a decrease of -0.1 points in MOTA. The integration of these two learned tokens leads to improved performance, as taking into account both conditions for whether to associate a track more accurately represents real-world situations. This decreases the number of false negatives by -464 compared to the baseline. Nevertheless, the improvement is not reflected in the MOTA score, since $+443$ additional false positives are concurrently introduced.

So far, ByteFormer has been guided solely by the appearance of the objects, without taking into account their locations within the frame. Incorporating positional information is crucial for distinguishing between visually similar objects and fostering deeper connections between appearance and motion. By adding our spatiotemporal encoding STE, the MOTA score is further increased by $+0.2$ points. Nonetheless, a high number of false negatives still persist due to the occasional generation of duplicate tracks. These negatively impact the system when kept active, and so far ByteFormer has had no way of identifying them. To address this issue, we integrate contextual information from nearby objects, successfully reducing false negatives by -269 and resulting in a MOTA increase of $+0.4$ points. The best results are achieved when combining the contextual information, the spatiotemporal encoding, and the learned tokens, resulting in an algorithm that outperforms the baseline by $+0.6$ points. This improvement is attributed to the successful recovery of -794 false negatives that would have otherwise been discarded.

4.4.3 Comparison to State of the Art

We compare ByteFormer to current state-of-the-art trackers on the test set of MOT17 [34] using private detections, which is the most widely adopted multi-object tracking benchmark. Results are shown in Table 4.3, with **bold** representing the best result, followed by **red** and **blue**, respectively. The comparison encompasses the online trackers that currently constitute the state of the art in multi-object tracking. Again, it's worth noting that we do not perform any fine-tuning on the training set of the benchmark as we only use MOTSynth [41] for training.

The comparative analysis features ByteTrack [174] in both versions: **offline** and **online**. This approach warrants special consideration as it is the work that introduces the BYTE association mechanism, which serves as the foundation for our proposal. In its original study, ByteTrack presents the results displayed in **gray**, attained via an offline interpolation procedure and leveraging distinct custom thresholds for each sequence in the test set. However, this strategy does not accurately represent real-world scenarios. Consequently, it is paramount to also provide ByteTrack's performance in its online setting and without the employment of tailored thresholds, denoted in **black**. It is important to mention that despite utilizing BYTE as the basis for our technique, we refrain from integrating any of the supplementary modifications employed in ByteTrack. Instead, we propose a fully-online method that does not require per-sequence tuning.

In the MOT Challenge [34], our ByteFormer tracker outperforms the current state-of-the-art in online multi-object tracking by a fair margin of $+0.4$ MOTA points. Our straightforward yet

	MOTA ↑	HOTA ↑	IDF1 ↑	IDs ↓
TubeTK [110]	63.0	48.0	58.6	5727
CTracker [114]	66.6	49.0	57.4	5529
CenterTrack [182]	67.8	52.2	64.7	3039
QDTrack [111]	68.7	53.9	66.3	3378
FairMOT [175]	73.7	59.3	72.3	3303
TraDeS [157]	69.1	52.7	63.9	3555
CrowdTrack [128]	75.6	60.3	73.6	2544
CorrTracker [151]	76.5	60.7	73.6	3369
MeMOT [15]	69.0	56.9	72.5	2724
MOTR [173]	73.4	57.8	68.6	2439
OUTrack [92]	73.5	58.7	70.2	4122
GTR [184]	75.3	59.1	71.5	2859
Unicorn [162]	77.2	61.7	75.5	5379
MTracker [176]	77.3	—	75.9	3255
MO3TR [187]	77.6	60.3	72.9	2847
GHOST [125]	78.7	62.8	77.1	2325
ByteTrack* [174]	78.9	62.8	77.1	2363
ByteTrack* [174]	80.3	63.1	77.3	2196
ByteFormer (ours)	79.3	63.1	77.7	2358

Table 4.3: Comparison to state-of-the-art for private detections on MOT17 test set. *since ByteTrack applies interpolation ---which makes it offline--- and uses different thresholds for each video we also show its re-computed results by [125] (online black / offline gray).

effective approach not only increases object coverage by retrieving false negatives that would otherwise have been discarded but also enhances identity preservation. By bridging the gaps where tracks lack proper matches, we manage to sustain their identities for a more extended period of time and prevent the assignment of incorrect identities, leading to a +0.6 point increase in IDF1 score. Collectively, these enhancements render our proposal highly competitive in terms of HOTA score, where it also attains state-of-the-art performance.

Our results reinforce the idea that attaining state-of-the-art performance does not necessarily rely on the pursuit of increasingly intricate end-to-end tracking pipelines. Instead, by focusing our efforts on thoroughly exploiting all the information available during tracking —e.g., unmatched trajectories— we can successfully recover a considerable number of objects that might have otherwise been dismissed by the detector.

4.5 CONCLUSIONS AND FUTURE WORK

In this work, we have demonstrated that it is not necessary to re-think intricate detection pipelines or to employ offline methods to increase object recall. Instead, we argue that if a basic motion model can associate bounding boxes, it can also be utilized to recover missing detections and prevent premature track termination. In order to achieve this, we present an extension of the BYTE algorithm that takes into account motion-based proposals in the absence of available detections. To handle the matching between tracks and new candidates, we introduce ByteFormer, our inno-

vative Transformer-based architecture that efficiently exploits both visual and spatial information while learning their interrelationships. Our fully online ByteFormer system demonstrates solid performance, achieving state-of-the-art results with a MOTA of 79.3 and a HOTA of 63.1 on the MOT17 benchmark. This underlines the effectiveness and robustness of the proposed approach, substantiating its potential for a wide range of multi-object tracking applications.

In terms of future work, several promising avenues could further enhance the performance of ByteFormer. First, it could be worth investigating the integration of additional features, such as incorporating semantic information or depth data, as it may lead to improved tracking accuracy, particularly in more complex and challenging scenarios. These extra features could potentially provide a richer context for the model to better discern objects in difficult environments.

Following the current trend in natural language processing and computer vision, exploring the use of unsupervised or self-supervised learning methodologies for training ByteFormer could help reduce its dependency on labeled data. By leveraging these alternative training techniques, the model may become more versatile and capable of adapting to new and unseen tracking challenges with minimal need for annotation.

Researching and developing computationally efficient variants of the ByteFormer architecture would make it more suitable for real-time applications and deployment scenarios with limited resources. Optimizing the model’s computational demands could allow it to be used in situations where swift response times and low-latency processing are crucial.

Lastly, assessing the performance of ByteFormer in other tracking tasks, such as multi-object tracking in 3D spaces —e.g., autonomous vehicles—, and open sets of categories among different datasets could also be of great interest. Evaluating the tracker in these diverse contexts could reveal its strengths and limitations, and potentially inspire further refinements to the ByteFormer architecture to extend the applicability of the model.

Through these advancements, we believe that ByteFormer provides a promising foundation for the ongoing development and application of Transformer-based architectures in multi-object tracking, driving the field toward more robust and reliable solutions that do not depend on overly complex detection pipelines.

5 Conclusions

The prevailing approach in multi-object tracking currently relies on tracking by detection, causing systems to be significantly impacted by the performance of their employed detectors. Nevertheless, ensuring high-quality detections throughout all video frames is far from a guarantee, mainly due to factors such as crowded scenes, decreased visibility, or real-time processing constraints that preclude the utilization of sophisticated detectors. This doctoral thesis acknowledges and addresses these limitations by presenting novel solutions that not only circumvent the reliance on object detections but also advance the current state of the art in multi-object tracking.

Chapter 2 presents SiamMT, a sophisticated framework that capitalizes on cutting-edge single-object tracking techniques to follow multiple class-agnostic targets. It proves to be well-suited for real-time systems, as it is capable of tracking up to 100 concurrent objects in HD720-resolution videos at over 25 fps without relying on detections. This is made possible through the development of a novel crop-and-resize operator, the reutilization of feature computations for multiple objects, and the creation of a new computationally efficient pairwise similarity operator. These latter two advancements are particularly valuable for extending to other architectures, as they serve as the foundation for the method introduced in Chapter 3. Furthermore, SiamMT's benefits extend beyond its speed, as it achieves state-of-the-art accuracy and robustness results on five major real-time benchmarks —garnering an average increase of 9.5 points compared to the top-performing counterpart at 25 fps— and permits the reusability of weights learned by SiamFC, delivering highly competitive outcomes as well.

Chapter 3 presents SiamMOTION, an advanced real-time multi-object tracker that offers improved robustness towards distractors and objects of varying sizes. Drawing inspiration from SiamMT, SiamMOTION shares computations among objects and utilizes the pairwise cross-correlation outlined in Chapter 2 as the foundation for its highly efficient similarity operator. Building upon this, SiamMOTION integrates cutting-edge techniques inspired by object detection. Specifically, it incorporates a sophisticated-yet-efficient proposal engine that generates high-quality features and a comparison head that effectively yields high-quality distractor-aware predictions. SiamMOTION has been assessed on a variety of video databases, attaining a real-time performance that excels beyond the existing state of the art, achieving an increase of +2.9 points in both average accuracy and average robustness compared to the top-performing counterpart at 25 fps, and differences of more than +11.6 points when contrasted with other trackers.

Moreover, an in-depth analysis of each component reveals that all newly-introduced features contribute significantly to the enhancement of tracking quality, with the proposal engine accounting for 68% of the improvement relative to the baseline and the comparison head responsible for 32%. This positions SiamMOTION as one of the foremost state-of-the-art proposals for motion estimation without requiring detections.

In addition, we have also investigated situations in which detections are not available for some of the objects within the scene. Consequently, in Chapter 4, we suggest utilizing motion-based proposals to maintain track continuity without the necessity of matching detections. To accomplish this, we put forth a generalization of the BYTE algorithm and solve the additional round of associations through ByteFormer. ByteFormer is a Transformer-based fully-online algorithm inspired by multiple-choice question-answering systems that leverages both motion and appearance cues to match tracks and candidates. Its ability to incorporate contextual information from other objects in the scene eliminates the need for post-processing its outputs, allowing it to recover numerous false negatives that were dismissed by the detector. This concept demonstrates remarkable performance, achieving a MOTA of 79.3 points and a HOTA of 63.1 points, outperforming existing state-of-the-art approaches by a fair margin. This highlights the efficacy and resilience of the proposed methodology, validating its potential applicability across a diverse array of multi-object tracking tasks.

Overall, this thesis draws attention to the significance of detections in multi-object tracking and exposes the considerable limitations of existing techniques when detections are absent or insufficient. SiamMT and SiamMOTION outperform the state-of-the-art for real-time multi-object tracking in scenarios lacking frame-by-frame detections, and ByteFormer offers a means to enhance the detector by recovering missed objects utilizing motion-based proposals. Still, numerous avenues remain to be explored, including the incorporation of segmentation masks, optimization for edge devices, and tracking in the wild.

FUTURE WORK

With SiamMOTION and SiamMT, we have provided the foundations for real-time tracking systems that do not depend on detections. These systems hold substantial potential for edge computing devices, where power consumption and device capabilities are often limited. As a possible future extension, the design of optimized architectures tailored to these low-resource devices could be explored. This potential advancement would encompass a comprehensive multi-object tracking (MOT) system, covering aspects such as track initiation, maintenance, termination, and efficient result transmission, all specifically designed for the target hardware. To this end, this could enable the integration of a network of cameras with enhanced tracking capabilities. The implications of this development would not only benefit intelligent video surveillance systems [63] but would also pave the way for in-device multi-camera tracking and multi-view 3D tracking solutions [40].

While bounding boxes continue to be highly relevant in many scenarios, certain situations demand more refined tracking outputs, such as crowded environments, high-accuracy applications, or unusually-shaped objects. Consequently, there has been a recent shift from bounding-box-based tracking towards segmentation-based approaches [84], akin to the transition witnessed in object detection some years ago [106]. In light of this development, two primary research directions can be pursued: firstly, enhancing the accuracy of mask-based multi-object tracking systems, and secondly, addressing the computational challenges associated with these systems to enable real-time performance. The integration of segmentation masks offers the potential to not only better represent object shapes but also benefit the algorithms themselves by facilitating more sophisticated handling of occlusions and background clutter [113].

Addressing occlusions is also a very pertinent topic in multi-object tracking, as they represent one of the most important challenges faced by algorithms such as ByteFormer. One potential solution may involve transforming the scene into a 3D space, which can be achieved by incorporating multimodal data such as depth [153] or generating synthetic scenes from alternative viewpoints using Neural Radiance Fields (NeRF) [105, 168]. The rationale behind this approach is that occlusions are a phenomenon resulting from projecting 3D environments onto 2D images, and by expanding tracking to the third dimension, occlusion-related issues could be mitigated. This integration of an additional dimension has the potential to facilitate the development of more powerful yet simpler tracking algorithms.

However, in our current discussion, we have assumed that the categories of interest for tracking are predefined and that extensive labeled datasets are available for training. However, this may not always be the case in real-world scenarios. One possible option to address this issue is to focus on generating synthetic databases [133] and developing algorithms to effectively bridge the domain gap. Nonetheless, in numerous practical applications, trackers must operate in unconstrained environments with unknown object categories. Consequently, it is essential to explore algorithms that can adapt to these open-set problems using approaches such as few-shot learning or meta-learning techniques [73] to enable robust and efficient tracking in the wild [94].

In connection to the aforementioned point, self-supervised learning stands out as a particularly potent approach, especially when combined with Transformer-based architectures. However, its applications should not be confined to a specific domain. The emergence of foundation models [138, 152] will undoubtedly revolutionize artificial intelligence and interconnect its various fields, including multi-object tracking. Drawing from well-established principles in transfer learning, in addition to the latest progress in both deep learning and large-scale computing systems, these models exhibit remarkably unexpected capabilities and significantly boost efficiency in numerous downstream tasks. Revolutionary contributions like Segment Anything (SAM) [75] are just the tip of the iceberg of the vast potential that artificial intelligence is poised to offer in the forthcoming years.

Bibliography

- [1] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *European Conf. Comput. Vis. (ECCV)*, volume 12356, pages 158–177, 2020.
- [2] Jérôme Berclaz, François Fleuret, Engin Türetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1806–1819, 2011.
- [3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 941–951, 2019.
- [4] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *European Conf. Comput. Vis. (ECCV) Workshops*, pages 850–865, 2016.
- [5] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 3464–3468, 2016.
- [6] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 6181–6190, 2019.
- [7] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *European Conf. Comput. Vis. (ECCV)*, pages 483–498, 2018.
- [8] R. J. Black, J. T. Whitney, Gerald M. Flachs, and Alton L. Gilbert. A pre-prototype real-time video tracking system. In *IEEE Nat. Aerospace Elec. Conf. (NAECON)*, pages 156–160, 1976.
- [9] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, pages 1–6, 2017.

- [10] Margaret A. Boden. *Mind as Machine: A History of Cognitive Science*. Oxford University Press, 2008.
- [11] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2544–2550, 2010.
- [12] Jean-Yves Bouguet et al. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.
- [13] Michael Brady. Computational approaches to image understanding. *ACM Comput. Surv.*, 14(1):3–71, 1982.
- [14] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6246–6256, 2020.
- [15] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8080–8090, 2022.
- [16] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conf. Comput. Vis. (ECCV)*, volume 12346, pages 213–229, 2020.
- [17] David Casasent. Unified synthetic discriminant function computational formulation. *Appl. Opt.*, 23(10):1620–1627, 1984.
- [18] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 22877–22887, 2023.
- [19] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qihong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *European Conf. Comput. Vis. (ECCV)*, volume 13682, pages 375–392, 2022.
- [20] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, Rongrong Ji, Zhenjun Tang, and Xianxian Li. Siamban: Target-aware tracking with siamese box adaptive network. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–17, 2022.
- [21] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

- [22] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [23] Zhen Cui, Shengtao Xiao, Jiashi Feng, and Shuicheng Yan. Recurrently target-attending tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1449–1458, 2016.
- [24] Yan Dai, Ziyu Hu, Shuqi Zhang, and Lianjun Liu. A survey of detection-based video multi-object tracking. *Displays*, 75:102317, 2022.
- [25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, volume 1, pages 886–893, 2005.
- [26] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4660–4669, 2019.
- [27] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6638–6646, 2017.
- [28] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, pages 58–66, 2015.
- [29] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1430–1438, 2016.
- [30] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conf. Comput. Vis. (ECCV)*, pages 472–488, 2016.
- [31] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. Adaptive color attributes for real-time visual tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1090–1097, 2014.
- [32] Stamatia Dasiopoulou, Vasileios Mezaris, Ioannis Kompatsiaris, V.-K. Papastathis, and Michael G. Strintzis. Knowledge-assisted semantic video object detection. *IEEE Trans. Circuits Syst. Video Technol.*, 15(10):1210–1224, 2005.
- [33] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. TAO: A large-scale benchmark for tracking any object. In *European Conf. Comput. Vis. (ECCV)*, volume 12350, pages 436–454, 2020.

- [34] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. MOTChallenge: A benchmark for single-camera multiple target tracking. *Int. J. Comput. Vis.*, 129(4):845–881, 2021.
- [35] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian D. Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *CoRR*, abs/2003.09003, 2020.
- [36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 248–255, 2009.
- [37] Xingping Dong, Jianbing Shen, Ling Shao, and Fatih Porikli. Clnet: A compact latent network for fast adjusting siamese trackers. In *European Conf. Comput. Vis. (ECCV)*, pages 378–395, 2020.
- [38] Xingping Dong, Jianbing Shen, Wenguan Wang, Ling Shao, Haibin Ling, and Fatih Porikli. Dynamical hyperparameter optimization via deep reinforcement learning in tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1515–1529, 2019.
- [39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Repr. (ICLR)*, 2021.
- [40] Martin Engilberge, Weizhe Liu, and Pascal Fua. Multi-view tracking using weakly supervised human motion prediction. In *IEEE Winter Conf. Appl. Comp. Vis. (WACV)*, pages 1582–1592, 2023.
- [41] Matteo Fabbri, Guillem Brasó, Gianluca Mageri, Aljoša Ošep, Riccardo Gasparini, Orkun Cetintas, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 10829–10839, 2021.
- [42] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conf. Comput. Vis. (ECCV)*, pages 450–466, 2018.
- [43] Baofu Fang, Gaoferi Mei, Xiaohui Yuan, Le Wang, Zaijun Wang, and Junyang Wang. Visual SLAM for robot navigation in healthcare facility. *Pattern Recognit.*, 113:107822, 2021.

- [44] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3038–3046, 2017.
- [45] Weitao Feng, Baopu Li, and Wanli Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. In *Int. Conf. Digital Image Comput. Tech. Appl. (DICTA)*, pages 1–10, 2022.
- [46] M. Fernández-Sanjurjo, M. Mucientes, and V. Brea. Real-time multiple object visual tracking for embedded GPU systems. *IEEE Internet Things J.*, 8:9177–9188, 2021.
- [47] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Eng.*, 8(3):173–184, 1983.
- [48] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13774–13783, 2021.
- [49] Abdalla Gad, Tasnim Basmaji, Maha Yaghi, Huda Alheeh, Mohammad Alkhedher, and Mohammed Ghazal. Multiple object tracking in robotic applications: Trends and challenges. *Appl. Sci.*, 12(19), 2022.
- [50] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 18686–18695, 2023.
- [51] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021.
- [52] Alton L. Gilbert and Gerald M. Flachs. A new target tracking algorithm for the automatic programmable film reader. *White Sands Missile Range Tech. Rep.*, 1974.
- [53] Ross B. Girshick. Fast R-CNN. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1440–1448, 2015.
- [54] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 580–587, 2014.
- [55] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6268–6276, 2020.

- [56] Shuman Guo, Shichang Wang, Zhenzhong Yang, Lijun Wang, Huawei Zhang, Pengyan Guo, Yuguo Gao, and Junkai Guo. A review of deep learning-based visual multi-object tracking algorithms for autonomous driving. *Appl. Sci.*, 12(21), 2022.
- [57] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5299–5309, 2021.
- [58] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2980–2988, 2017.
- [59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 770–778, 2016.
- [60] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conf. Comput. Vis. (ECCV)*, pages 749–765, 2016.
- [61] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conf. Comput. Vis. (ECCV)*, pages 702–715, 2012.
- [62] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2014.
- [63] Fabian Herzog, Junpeng Chen, Torben Teepe, Johannes Gilg, Stefan Hörmann, and Gerhard Rigoll. Synthechile: Multi-vehicle multi-camera tracking in virtual cities. In *IEEE Winter Conf. Appl. Comp. Vis. (WACV) Workshops*, pages 1–11, 2023.
- [64] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981.
- [65] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3588–3597, 2018.
- [66] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2019.
- [67] Yani Ioannou, Duncan P. Robertson, Roberto Cipolla, and Antonio Criminisi. Deep roots: Improving CNN efficiency with hierarchical filter groups. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5977–5986, 2017.

- [68] Rudolf E Kalman. A new approach to linear filtering and prediction theory. *J. Fluids. Eng.*, 82(1):35–45, 1960.
- [69] Rangachar Kasturi, Dmitry B. Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John S. Garofolo, Rachel Bowers, Matthew Boonstra, Valentina N. Korzhova, and Jing Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):319–336, 2009.
- [70] Abdul Hannan Khan, Mohsin Munir, Ludger van Elst, and Andreas Dengel. F2dnet: Fast focal detection network for pedestrian detection. In *IEEE Int. Conf. Pattern Recognit. (ICPR)*, pages 4658–4664, 2022.
- [71] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Correlation filters with limited boundaries. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4630–4638, 2015.
- [72] Chanh Kim, Fuxin Li, Mazen Alotaibi, and James M. Rehg. Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 9553–9562, 2021.
- [73] Sangwon Kim, Jimi Lee, and ByoungChul Ko. SSL-MOT: self-supervised learning based multi-object tracking. *Appl. Intell.*, 53(1):930–940, 2023.
- [74] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Repr. (ICLR)*, pages 1–15, 2015.
- [75] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *CoRR*, abs/2304.02643, 2023.
- [76] Matej Kristan et al. The seventh visual object tracking VOT2019 challenge results. In *IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, pages 2206–2241, 2019.
- [77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1106–1114, 2012.
- [78] Harold W. Kuhn. The hungarian method for the assignment problem. *Nav. Res. Logist.*, 52(1-2):83–97, 1955.
- [79] Ratnesh Kumar, Guillaume Charpiat, and Monique Thonnat. Multiple object tracking by efficient graph partitioning. In *Asian Conf. Comput. Vis. (ACCV)*, volume 9006, pages 445–460, 2014.

- [80] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese CNN for robust target association. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 418–425, 2016.
- [81] Laura Leal-Taixé, Anton Milan, Konrad Schindler, Daniel Cremers, Ian D. Reid, and Stefan Roth. Tracking the trackers: An analysis of the state of the art in multiple object tracking. *CoRR*, abs/1704.02781, 2017.
- [82] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4282–4291, 2019.
- [83] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8971–8980, 2018.
- [84] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 18825–18835, 2022.
- [85] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conf. Comput. Vis. (ECCV)*, pages 254–265, 2015.
- [86] Liting Lin, Heng Fan, Zhipeng Zhang, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. In *Neural Inf. Process. Syst. (NeurIPS)*, 2022.
- [87] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 936–944, 2017.
- [88] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2999–3007, 2017.
- [89] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42:318–327, 2020.
- [90] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conf. Comput. Vis. (ECCV) Workshops*, pages 740–755, 2014.

- [91] Mengyin Liu, Jie Jiang, Chao Zhu, and Xu-Cheng Yin. VLPD: Context-aware pedestrian detection via vision-language semantic self-supervision. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6662–6671, 2023.
- [92] Qiankun Liu, Dongdong Chen, Qi Chu, Lu Yuan, Bin Liu, Lei Zhang, and Nenghai Yu. Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing*, 483:333–347, 2022.
- [93] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *European Conf. Comput. Vis. (ECCV)*, volume 9905, pages 21–37, 2016.
- [94] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Deva Ramanan, Bastian Leibe, Aljosa Osep, and Laura Leal-Taixé. Opening up open world tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 19023–19033, 2022.
- [95] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Repr. (ICLR)*, pages 1–11, 2019.
- [96] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.
- [97] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *European Conf. Comput. Vis. (ECCV)*, pages 353–369, 2018.
- [98] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. Artif. Intell. IJCAI*, volume 2, pages 674–679, 1981.
- [99] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021.
- [100] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3074–3082, 2015.
- [101] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5388–5396, 2015.

- [102] Peter S. Maybeck. The kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*, pages 194–204. Springer New York, 1990.
- [103] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8834–8844, 2022.
- [104] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016.
- [105] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022.
- [106] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.
- [107] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating CNNs in a tree structure for visual tracking. *CoRR*, abs/1608.07242, 2016.
- [108] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4293–4302, 2016.
- [109] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. An adaptive color-based particle filter. *Image Vision Comput.*, 21(1):99–110, 2003.
- [110] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6307–6317, 2020.
- [111] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 164–173, 2021.
- [112] Seymour A. Papert. The summer vision project. *MIT AI Memos*, 1966.
- [113] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. Robust visual tracking by segmentation. In *European Conf. Comput. Vis. (ECCV)*, volume 13682, pages 571–588, 2022.

- [114] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conf. Comput. Vis. (ECCV)*, volume 12349, pages 145–161, 2020.
- [115] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Research*, pages 1–12, 2018.
- [116] Akshay Rangesh and Mohan Manubhai Trivedi. No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars. *IEEE Trans. Intell. Veh.*, 4(4):588–599, 2019.
- [117] J. Usha Rani and P. Raviraj. Real-time human detection for intelligent video surveillance: An empirical research and in-depth review of its applications. *SN Comput. Sci.*, 4(3):258, 2023.
- [118] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7464–7473, 2017.
- [119] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 779–788, 2016.
- [120] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6517–6525, 2017.
- [121] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.
- [122] Seyed Hamid Rezaatofghi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony R. Dick, and Ian D. Reid. Joint probabilistic matching using m-best solutions. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 136–145, 2016.
- [123] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015.

- [124] Fatemeh Sadat Saleh, Sadegh Aliakbarian, Hamid Rezatofighi, Mathieu Salzmann, and Stephen Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 14329–14339, 2021.
- [125] Jenny Seidenschwarz, Guillem Brasó, Ismail Elezi, and Laura Leal-Taixé. Simple cues lead to a strong multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13813–13823, 2023.
- [126] Bing Shuai, Andrew G. Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 12372–12382, 2021.
- [127] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3786–3795, 2017.
- [128] Daniel Stadler and Jürgen Beyerer. On the performance of crowd-specific detectors in multi-pedestrian tracking. In *IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, pages 1–12, 2021.
- [129] Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2246–2252, 1999.
- [130] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *CoRR*, abs/2012.15460, 2020.
- [131] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):104–119, 2021.
- [132] ShiJie Sun, Naveed Akhtar, Xiangyu Song, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Simultaneous detection and tracking with motion modelling for multiple object tracking. In *European Conf. Comput. Vis. (ECCV)*, volume 12369, pages 626–643, 2020.
- [133] Tao Sun, Mattia Segù, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 21339–21350, 2022.
- [134] Valtteri Takala and Matti Pietikäinen. Multi-object tracking using color, texture and motion. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2007.

- [135] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 10778–10787, 2020.
- [136] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3701–3710, 2017.
- [137] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1420–1429, 2016.
- [138] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.
- [139] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *Int. J. Comput. Vis.*, 104(2):154–171, 2013.
- [140] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Process.*, 4(11):1549–1560, 1995.
- [141] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2805–2813, 2017.
- [142] Lorenzo Vaquero, Víctor M. Brea, and Manuel Mucientes. Fast multi-object tracking with feature pyramid and region proposal networks. In *IEEE Int. Conf. Pattern Recognit. (ICPR)*, pages 3485–3491, 2022.
- [143] Lorenzo Vaquero, Víctor M. Brea, and Manuel Mucientes. Real-time siamese multiple object tracker with enhanced proposals. *Pattern Recognit.*, 135:109141, 2023.
- [144] Lorenzo Vaquero, Manuel Mucientes, and Víctor M. Brea. SiamMT: Real-time arbitrary multi-object tracking. In *IEEE Int. Conf. Pattern Recognit. (ICPR)*, pages 707–714, 2020.
- [145] Lorenzo Vaquero, Manuel Mucientes, and Víctor M. Brea. Tracking more than 100 arbitrary objects at 25 fps through deep learning. *Pattern Recognit.*, 121:108205, 2022.
- [146] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inf. Process. Syst. (NIPS)*, pages 5998–6008, 2017.

- [147] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 511–518, 2001.
- [148] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3119–3127, 2015.
- [149] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *CoRR*, abs/1501.04587, 2015.
- [150] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. *Adv. Neural Inf. Process. Sys. (NIPS)*, 26, 2013.
- [151] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3876–3886, 2021.
- [152] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 14408–14419, 2023.
- [153] Yong Wang, Xian Wei, Hao Shen, Lu Ding, and Jiuqing Wan. Robust fusion for RGB-D tracking using CNN features. *Appl. Soft Comput.*, 92:106302, 2020.
- [154] Zelun Wang and Jyh-Charn Liu. Translating math formula images to latex sequences using deep neural networks with sequence-level training. *Int. J. Document Anal. Recognit.*, 24(1):63–75, 2021.
- [155] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conf. Comput. Vis. (ECCV)*, volume 12356, pages 107–122, 2020.
- [156] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 3645–3649, 2017.
- [157] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 12352–12361, 2021.

- [158] Haisheng Xu and Youchan Zhu. Real-time object tracking based on improved fully-convolutional siamese network. *Comput. Electr. Eng.*, 86:106755, 2020.
- [159] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. An accelerated correlation filter tracker. *Pattern Recognit.*, 102:107172, 2020.
- [160] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6786–6795, 2020.
- [161] Yinda Xu, Zeyu Wang, Zuoxin Li, Yuan Ye, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI Conf. Artif. Intell. (AAAI)*, pages 12549–12556, 2020.
- [162] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *European Conf. Comput. Vis. (ECCV)*, volume 13681, pages 733–751, 2022.
- [163] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 10428–10437, 2021.
- [164] Bo Yang and Ram Nevatia. An online learned CRF model for multi-target tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2034–2041, 2012.
- [165] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conf. Comput. Vis. (ECCV)*, volume 13682, pages 341–357, 2022.
- [166] Junbo Yin, Wenguan Wang, Qinghao Meng, Ruigang Yang, and Jianbing Shen. A unified object motion and affinity model for online multi-object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6767–6776, 2020.
- [167] Yingjie Yin, De Xu, Xingang Wang, and Lei Zhang. Agunet: Annotation-guided u-net for fast one-shot video object segmentation. *Pattern Recognit.*, 110:107580, 2021.
- [168] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4578–4587, 2021.
- [169] Hongyang Yu, Guorong Li, Weigang Zhang, Qingming Huang, Dawei Du, Qi Tian, and Nicu Sebe. The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. *Int. J. Comput. Vis.*, 128(5):1141–1159, 2020.

- [170] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6727–6736, 2020.
- [171] Di Yuan, Xin Li, Zhenyu He, Qiao Liu, and Shuwei Lu. Visual object tracking with adaptive structural convolutional network. *Knowl. Based Syst.*, 194:105554, 2020.
- [172] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conf. Comput. Vis. (ECCV)*, volume 7573, pages 343–356, 2012.
- [173] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: end-to-end multiple-object tracking with transformer. In *European Conf. Comput. Vis. (ECCV)*, volume 13687, pages 659–675, 2022.
- [174] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European Conf. Comput. Vis. (ECCV)*, pages 1–21, 2022.
- [175] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129(11):3069–3087, 2021.
- [176] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Robust multi-object tracking by marginal inference. In *European Conf. Comput. Vis. (ECCV)*, volume 13682, pages 22–40, 2022.
- [177] Yucheng Zhang, Tian Wang, Kexin Liu, Baochang Zhang, and Lei Chen. Recent advances of single-object tracking methods: A brief survey. *Neurocomputing*, 455:1–11, 2021.
- [178] Yujia Zhang, Xiaodan Liang, Dingwen Zhang, Min Tan, and Eric P. Xing. Unsupervised object-level video summarization with online motion auto-encoder. *Pattern Recognit. Lett.*, 130:376–385, 2020.
- [179] Kaikai Zhao, Takashi Imaseki, Hiroshi Mouri, Einoshin Suzuki, and Tetsu Matsukawa. From certain to uncertain: Toward optimal solution for offline multiple object tracking. In *IEEE Int. Conf. Pattern Recognit. (ICPR)*, pages 2506–2513, 2020.
- [180] Linyu Zheng, Ming Tang, Yingying Chen, Guibo Zhu, Jinqiao Wang, and Hanqing Lu. Improving multiple object tracking with single object tracking. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2453–2462, 2021.

- [181] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod: End-to-end video object detection with spatial-temporal transformers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7853–7869, 2023.
- [182] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conf. Comput. Vis. (ECCV)*, volume 12349, pages 474–490, 2020.
- [183] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conf. Comput. Vis. (ECCV)*, pages 474–490, 2020.
- [184] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8761–8770, 2022.
- [185] Zongwei Zhou, Wenhan Luo, Qiang Wang, Junliang Xing, and Weiming Hu. Distractor-aware discrimination learning for online multiple object tracking. *Pattern Recognit.*, 107:107512, 2020.
- [186] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Qinghua Hu, and Haibin Ling. Vision meets drones: Past, present and future. *CoRR*, abs/2001.06303, 2020.
- [187] Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, and Hamid Rezatofighi. Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. *CoRR*, abs/2103.14829, 2022.
- [188] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *European Conf. Comput. Vis. (ECCV)*, pages 103–119, 2018.
- [189] Djemel Ziou, Salvatore Tabbone, et al. Edge detection techniques: An overview. *Pattern Recognit. Image Anal.*, 8:537–559, 1998.

A SiamMOTION Supplementary Material

This appendix contains the supplementary material of Chapter 3 relative to SiamMOTION. In particular, Figure A.1 shows some qualitative results generated by the algorithm on different multi-object databases of people and vehicles. Figure A.2 shows some failure cases mainly caused by situations involving rapid camera movements, severe occlusions, or a combination of both. Lastly, Figure A.3 illustrates the learning curves for the different configurations of the modules of SiamMOTION.



Figure A.1: Qualitative results of SiamMOTION.

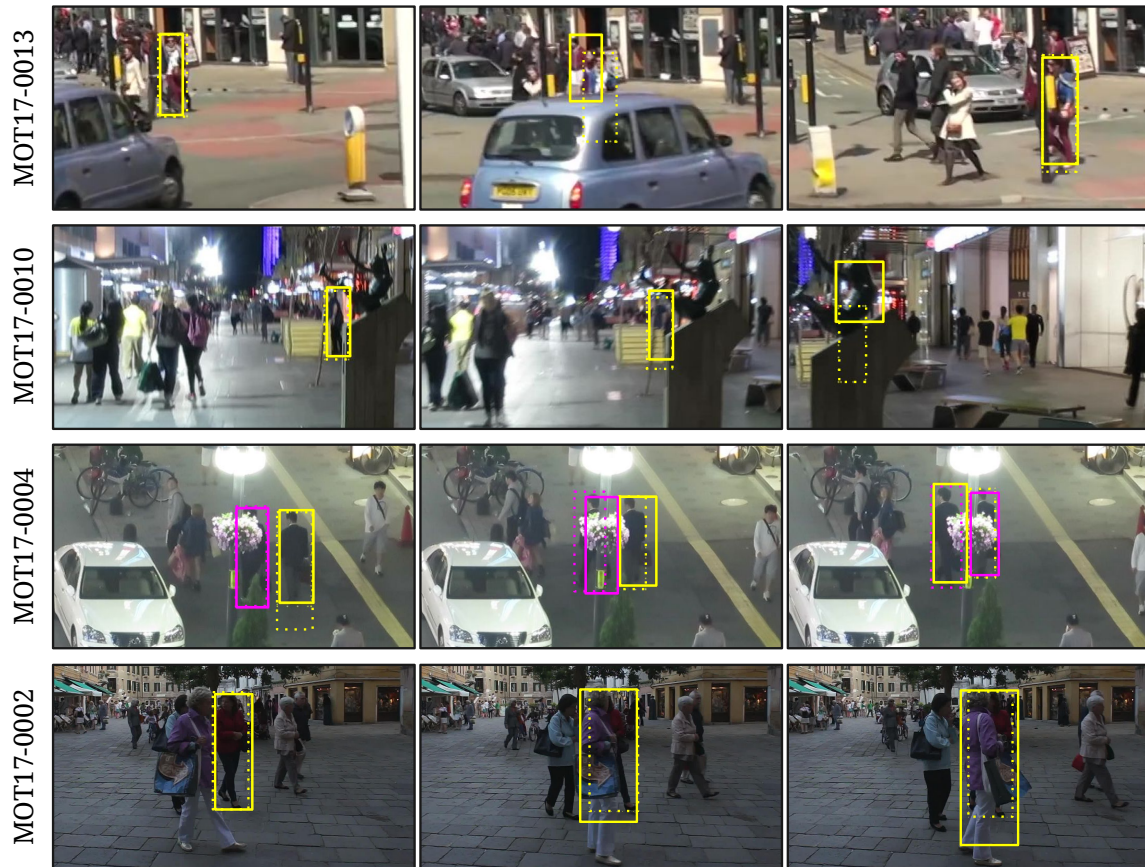


Figure A.2: Specific examples of failure cases for SiamMOTION. Predicted bounding boxes are shown with a solid line whereas the ground truth is shown with a dotted line. Boxes of other objects are not shown to avoid cluttering.

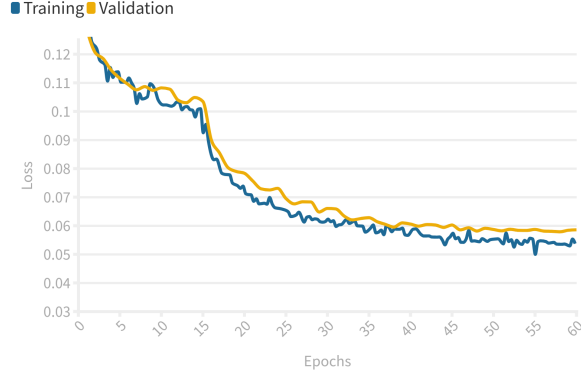
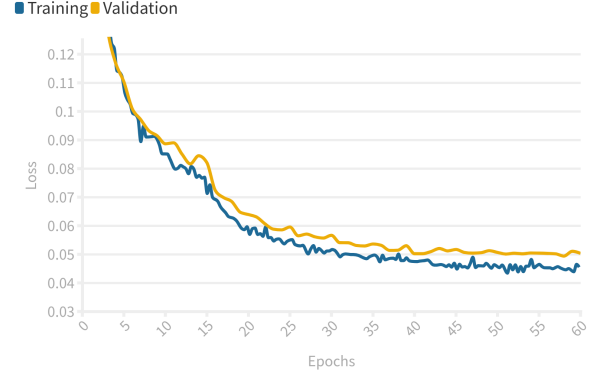
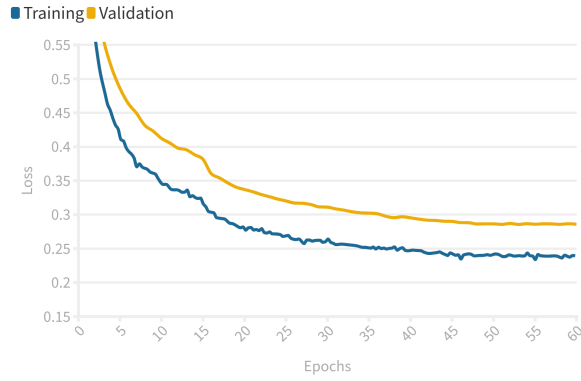
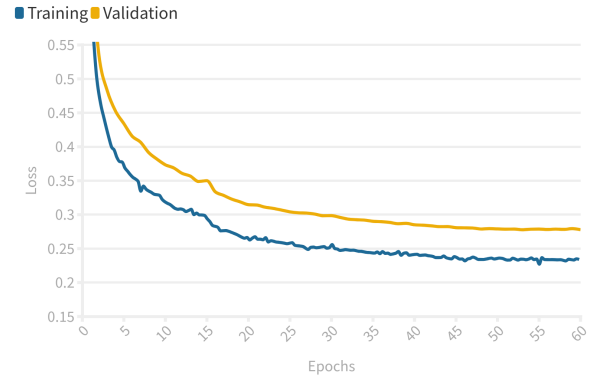
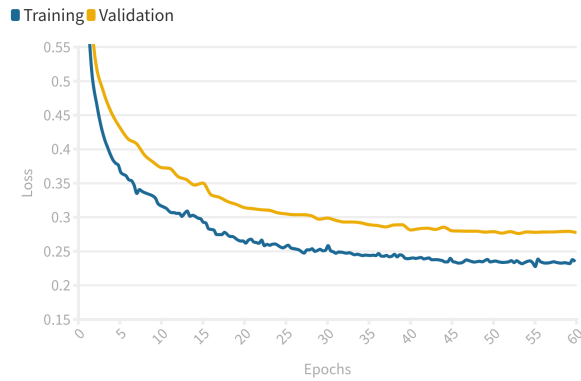
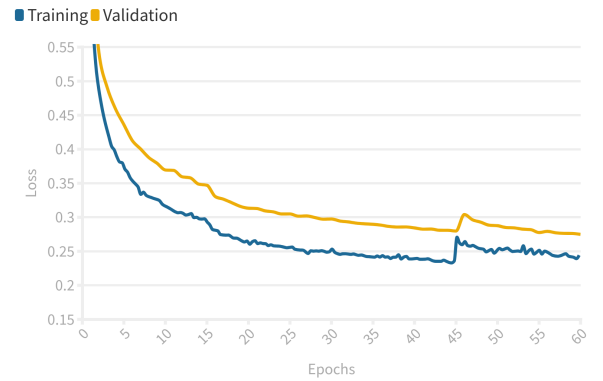
Baseline**B+F****B+F+R****B+F+R+A****B+F+R+A+P****B+F+R+A+P+I**

Figure A.3: Learning curves for different configurations of SiamMOTION. A version of SiamMT with a ResNet-18 as backbone is the baseline (B). F: Feature-Pyramid-based RoI extractor. R: PD-RPN similarity operation. A: Attention module. P: Multi-object penalization. I: Inertia module. The scales of the graphs are not comparable, as Baseline and B+F employ a different loss function and B+F+R+A+P+I includes the loss of the inertia module when it begins fitting from epoch 45 onward.

B Reproduction Rights

This appendix provides details pertaining to the licensing information, reproduction rights, and copyright restrictions of the original publications from which the content of this PhD thesis has been partially derived.

B.1 JOURNALS

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Tracking More Than 100 Arbitrary Objects at 25 FPS Through Deep Learning*. Pattern Recognition, Volume 121, pp. 108205, 2022. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2021.108205.

^a Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.



Tracking more than 100 arbitrary objects at 25 FPS through deep learning

Author: Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes

Publication: Pattern Recognition

Publisher: Elsevier

Date: January 2022

© 2021 The Authors. Published by Elsevier Ltd.

Creative Commons Attribution-NonCommercial-No Derivatives License (CC BY NC ND)

This article is published under the terms of the [Creative Commons Attribution-NonCommercial-No Derivatives License \(CC BY NC ND\)](#).

For non-commercial purposes you may copy and distribute the article, use portions or extracts from the article in other works, and text or data mine the article, provided you do not alter or modify the article without permission from Elsevier. You may also create adaptations of the article for your own personal use only, but not distribute these to others. You must give appropriate credit to the original work, together with a link to the formal publication through the relevant DOI, and a link to the Creative Commons user license above. If changes are permitted, you must indicate if any changes are made but not in any way that suggests the licensor endorses you or your use of the work.

Permission is not required for this non-commercial use. For commercial use please continue to request permission via RightsLink.

Figure B.1: License information of [145].

Quality Indicators: Journal Impact Factor: 8.518 (JCR 2021). Journal ranked in JCR 2021, in Computer Science, Artificial Intelligence (Q1, 22/145).

PhD Candidate Contribution (ANSI/NISO CRediT): Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft.

Reproduction Rights: Contents of Chapter 2 are extracted from this publication. Licensed under Creative Commons CC BY NC ND (Figure B.1).

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Real-Time Siamese Multiple Object Tracker With Enhanced Proposals*. Pattern Recognition, Volume 135, pp. 109141, 2023. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2022.109141.

^a Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.



Real-time siamese multiple object tracker with enhanced proposals

Author: Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes

Publication: Pattern Recognition

Publisher: Elsevier

Date: March 2023

© 2022 The Author(s). Published by Elsevier Ltd.

Creative Commons Attribution-NonCommercial-No Derivatives License (CC BY NC ND)

This article is published under the terms of the [Creative Commons Attribution-NonCommercial-No Derivatives License \(CC BY NC ND\)](#).

For non-commercial purposes you may copy and distribute the article, use portions or extracts from the article in other works, and text or data mine the article, provided you do not alter or modify the article without permission from Elsevier. You may also create adaptations of the article for your own personal use only, but not distribute these to others. You must give appropriate credit to the original work, together with a link to the formal publication through the relevant DOI, and a link to the Creative Commons user license above. If changes are permitted, you must indicate if any changes are made but not in any way that suggests the licensor endorses you or your use of the work.

Permission is not required for this non-commercial use. For commercial use please continue to request permission via RightsLink.

Figure B.2: License information of [143].

Quality Indicators: Journal Impact Factor: 8.518 (JCR 2021). Journal ranked in JCR 2021, in Computer Science, Artificial Intelligence (Q1, 22/145).

PhD Candidate Contribution (ANSI/NISO CRediT): Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft.

Reproduction Rights: Contents of Chapter 3 are extracted from this publication. Licensed under Creative Commons CC BY NC ND (Figure B.2).

B.2 CONFERENCES

Lorenzo Vaquero^a, Manuel Mucientes^a, Víctor M. Brea^a. *SiamMT: Real-Time Arbitrary Multi-Object Tracking*. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), pp. 707–714, Milano (Italy), 2020. ISSN: 1051-4651. DOI: 10.1109/ICPR48806.2021.9412625.

^aCentro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.



IEEE
Requesting
permission
to reuse
content from
an IEEE
publication

SiamMT: Real-Time Arbitrary Multi-Object Tracking

Conference Proceedings: 2020 25th International Conference on Pattern Recognition (ICPR)
Author: Lorenzo Vaquero
Publisher: IEEE
Date: 10 January 2021
Copyright © 2021, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Figure B.3: License information of [144].

Quality Indicators: Conference Ranking (GGS Rating): A-, Class 2.

PhD Candidate Contribution (ANSI/NISO CRediT): Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft.

Reproduction Rights: Contents of Chapter 2 are extracted from this publication. The permission granted by the publisher (IEEE) for the reproduction of this work is shown in Figure B.3

Lorenzo Vaquero^a, Víctor M. Brea^a, Manuel Mucientes^a. *Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks*. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), pp. 3485-3491, Montréal (Canada), 2022. ISSN: 2831-7475. DOI: 10.1109/ICPR56361.2022.9956673.

^aCentro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) and Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain.



Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks

Conference Proceedings: 2022 26th International Conference on Pattern Recognition (ICPR)

Author: Lorenzo Vaquero

Publisher: IEEE

Date: 21 August 2022

Copyright © 2022, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Figure B.4: License information of [142].

Quality Indicators: Conference Ranking (GGs Rating): A-, Class 2.

PhD Candidate Contribution (ANSI/NISO CRediT): Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft.

Reproduction Rights: Contents of Chapter 3 are extracted from this publication. The permission granted by the publisher (IEEE) for the reproduction of this work is shown in Figure B.4

List of Figures

Fig. 1.1	Output of a visual object tracking system after processing a video sequence.	4
Fig. 2.1	Diagram of a typical MOT system.	21
Fig. 2.2	Different architectures that show the evolution from SiamFC to SiamMT.	22
Fig. 2.3	Qualitative examples of SiamMT on different sequences.	25
Fig. 2.4	SiamFC and SiamMT network architectures during the inference phase.	26
Fig. 2.5	Pooled pixels when applying $\tilde{\kappa}$, RoIPool, and RoIAlign over a 4×4 features tensor.	27
Fig. 2.6	Transformation between pixel coordinates and features coordinates for a backbone with padding using RoIAlign and a backbone without padding using $\tilde{\kappa}$. . .	28
Fig. 2.7	Numerical example for a 33×33 -pixel object across different architectures.	30
Fig. 2.8	SiamMT’s feature extractor.	34
Fig. 2.9	Speed results for MT-VGA and MT-HD benchmarks with respect to the number of tracked objects.	38
Fig. 2.10	Comparison of average times per stage in the MT-HD benchmark for SiamFC-MI and SiamMT, with respect to the number of followed objects.	39
Fig. 3.1	Comparison of a multi-object tracking system lacking motion estimation between detections, SiamMT, and our proposal (SiamMOTION).	45
Fig. 3.2	SiamMOTION’s architecture.	48
Fig. 3.3	Features extracted with a backbone with padding and a backbone without it.	49
Fig. 3.4	SiamMOTION’s inertia module.	51
Fig. 3.5	SiamMOTION’s attention module.	53

Fig. 3.6	Pairwise-Depthwise-RPN.	55
Fig. 3.7	SiamMOTION’s multi-object penalization module.	56
Fig. 3.8	Box plot of the amount of overlap per object for the analyzed databases.	62
Fig. 4.1	Impact on MOTA when successful motion-based predictions without an associated detection are reported.	68
Fig. 4.2	Comparison between motion-based and appearance-based associations.	71
Fig. 4.3	Algorithm for updating a set of tracks $\mathcal{T}$	73
Fig. 4.4	ByteFormer architecture.	75
Fig. 4.5	Generation of the spatiotemporal embedding for each element in the sequence.	77
Fig. A.1	Qualitative results of SiamMOTION.	107
Fig. A.2	Specific examples of failure cases for SiamMOTION.	108
Fig. A.3	Learning curves for different configurations of SiamMOTION.	109
Fig. B.1	License information of [145].	112
Fig. B.2	License information of [143].	113
Fig. B.3	License information of [144].	114
Fig. B.4	License information of [142].	115

List of Tables

Tab. 2.1	State-of-the-art approaches for MVOT.	24
Tab. 2.2	Quantitative results for tracking quality.	36
Tab. 3.1	Ablation study with VOT-RT metrics @20 fps.	60
Tab. 3.2	VOT-RT metrics results.	63
Tab. 4.1	Ablation of different association approaches.	80
Tab. 4.2	Ablation of the different parts that integrate ByteFormer.	81
Tab. 4.3	Comparison to state-of-the-art for private detections on MOT17 test set. . . .	83



This thesis presents novel deep-learning approaches for tracking multiple objects in videos. Traditional multi-object trackers, which rely on frame-by-frame detections and primarily geometric attributes, are ill-suited for real-time environments and open-set scenarios. To overcome these limitations, we introduce SiamMT, an architecture that adapts single-object tracking techniques for handling multiple arbitrary targets in real time. This approach is further refined with SiamMOTION, which effectively manages distractors and accommodates objects of varying sizes by extracting semantically-rich features and proposing more accurate search areas. Lastly, we present ByteFormer, a Transformer-based architecture that complements a detector by recovering missed objects for enhanced performance in multi-object tracking.