UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PABLO ROBERLAN MANKE BARCELLOS

Rastreamento de Objetos em Sequências de Vídeo Utilizando Múltiplos Filtros de Correlação

Tese apresentada como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação

Orientador: Prof. Dr. Jacob Scharcanski

Porto Alegre
2021

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Barcellos, Pablo Roberlan Manke

Rastreamento de Objetos em Sequências de Vídeo Utilizando Múltiplos Filtros de Correlação / Pablo Roberlan Manke Barcellos. – Porto Alegre: PPGC da UFRGS, 2021.

94 f.: il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2021. Orientador: Jacob Scharcanski.

1. Rastreamento de objetos. 2. Filtros de correlação. 3. Rastreamento em vídeo. I. Jacob Scharcanski, . II. Título.

AGRADECIMENTOS

Primeiramente, agradeço a toda minha família e especialmente aos meus pais por todo o apoio, incentivo e motivação, que foram fundamentais para que eu chegasse até aqui. Agradeço meu orientador, Prof. Jacob Scharcanski, pelo apoio, dedicação e todo o auxílio recebido durante o curso, essenciais para a realização deste trabalho. Agradeço aos colegas do PPGC e do PPGEE pela convivência e por todos os momentos passados durante esse período.

Gostaria também de agradecer à Digicon S.A. e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio. Agradeço também o apoio da NVIDIA Corporation com a doação da GPU Titan Xp usada durante a pesquisa.

Agradeço também ao PPGC, pela oportunidade de realizar este trabalho, aos professores da UFRGS pelo esforço, dedicação e conhecimento partilhado ao longo do curso, e a todos que, de uma forma ou de outra, contribuíram para a minha formação.

RESUMO

O rastreamento de objetos em sequências de vídeos é um tema fundamental na área de processamento de imagens e visão computacional, sendo necessário nas mais diversas situações. Apesar dos diversos avanços na nessa área, o rastreamento de objetos continua sendo um problema desafiador, especialmente devido aos diversos fatores que podem afetar os resultados do rastreamento, como mudanças de iluminação e deformações não rígidas. Esta tese introduz um *framework* baseado em filtros de correlação capaz de rastrear objetos nas mais diversas situações. O método proposto utiliza um esquema colaborativo, combinando o uso de um filtro de correlação global com o uso de filtros de correlação locais para melhorar o processo de rastreamento. Ainda, o método utiliza feições extraídas usando Redes Neurais Convolucionais (CNN), e também utiliza uma estratégia para avaliar se os resultados estimados pelos filtros de correlação são confiáveis. Resultados experimentais realizados em *benchmarks* públicos mostram que o método proposto consegue obter bons resultados, sendo superior aos métodos comparativos do estado da arte.

Palavras-chave: Rastreamento de objetos. filtros de correlação. rastreamento em vídeo.

Object tracking in video sequences using multiple correlation filters.

ABSTRACT

The tracking of objects in video sequences is a fundamental subject in the field of image processing and computer vision, and it is required in a variety of situations. Despite several advances in this area, object tracking remains a challenging problem, especially due to the many factors that can affect the tracking results, such as illumination variations and non-rigid deformations. This thesis introduces a framework based on correlation filters capable of tracking objects in the most diverse situations. The proposed method uses a collaborative scheme, combining the use of a global correlation filter with the use of local correlation filters to improve the tracking process. Furthermore, the method uses features extracted using Convolutional Neural Networks (CNN), and also uses a strategy to evaluate if the results estimated by the correlation filters are reliable. Experimental results in public benchmarks show that the proposed method achieves good results, being superior to the state-of-the-art comparative methods.

Keywords: object tracking, correlation filters, video tracking.

LISTA DE ABREVIATURAS E SIGLAS

AUC	Area Under the Curve
CN	*Color Names*
CNN	Convolutional Neural Networks
DCF	Discriminative Correlation Filter
DFT	Transformada Discreta de Fourier
FFT	*Fast Fourier Transform*
HoG	Histograms of Oriented Gradients
HoG	*Histogram of Oriented Gradients*
IFFT	*Inverse Fast Fourier Transform*
KCF	Kernelized Correlation Filter
MACE	*Minimum Average Correlation Energy*
MEEM	Multiple Expert Entropy Minimization
MIL	Multiple Instance Learning
MOSSE	Minimum Output Sum of Squared Error
OPE	One-Pass Evaluation
PSR	*Peak to Sidelobe Ratio*
RGB	Espaço de cores *Red* (Vermelho), *Green* (Verde), *Blue* (Azul)
SIFT	*Scale Invariant Feature Transform*
SVM	Support Vector Machine
UMACE	*Unconstrained MACE*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O rastreamento de objetos é um problema fundamental na área de processamento de sinais e visão computacional, sendo crucial em diversas aplicações, incluindo processamento de vídeo, vigilância, controle de tráfego e veículos aéreos não tripulados (SMEULDERS et al., 2014; BARCELLOS et al., 2015; SUN et al., 2019). O problema consiste em, dado a posição inicial de um objeto (alvo) em uma sequência de vídeo, estimar a localização do objeto nos quadros subsequentes. Apesar dos avanços significativos nos últimos anos, o rastreamento de objetos utilizando imagens continua a ser um problema desafiador devido aos vários fatores que podem afetar o processo de rastreamento, como variações de escala, variações de iluminação, oclusões e deformações de objetos.

Os métodos de rastreamento podem ser categorizados em discriminativos e generativos (HU et al., 2017; NG; JORDAN, 2002). Os métodos generativos modelam uma distribuição conjunta e estimam as probabilidades de um conjunto de regiões/janelas candidatas com a finalidade de determinar qual delas melhor corresponde ao modelo do alvo. Os métodos discriminativos empregam uma distribuição condicional para treinar um classificador que seja capaz de separar o objeto rastreado do plano de fundo da cena (*background*) (LIU; LI; FANG, 2015). Recentemente, abordagens baseadas em filtro de correlação discriminativo (DCF - *Discriminative Correlation Filter*) obtiveram ótimos resultados no rastreamento de objetos em *benchmarks* (WU; LIM; YANG, 2013; WU; LIM; YANG, 2015; CHEN; HONG; TAO, 2015b; BOLME et al., 2010; DANELLJAN et al., 2014b; HENRIQUES et al., 2015). No primeiro quadro de uma sequência de imagens, uma janela de pixels centrada no objeto é usada para aprender um filtro de correlação para o alvo a ser rastreado. O alvo é rastreado nos quadros subsequentes correlacionando esse filtro aprendido com uma janela de busca, onde a posição com o valor máximo de resposta de correlação (pico) indica a localização do centro do alvo rastreado. Em seguida, o filtro é atualizado usando um esquema *on-line*, que leva em conta a nova aparência e localização do alvo.

Inicialmente, os filtros de correlação recorreram a feições simples, geralmente apenas a intensidade dos pixels (BOLME et al., 2010). Posteriormente, feições mais robustas foram propostas, incluindo representações de feições utilizando múltiplos canais, como *Color Names* (DANELLJAN et al., 2014b), histogramas de gradientes orientados (HoG - *Histograms of Oriented Gradients*) (HENRIQUES et al., 2015), e feições extraídas usando Redes Neurais Convolucionais (CNN - *Convolutional Neural Networks*) (MA

et al., 2015). Além disso, várias abordagens surgiram para melhorar o desempenho dos filtros de correlação, incluindo a introdução de *kernels* não lineares (HENRIQUES et al., 2015), filtros de correlação baseados em partes (LI; ZHU; HOI, 2015; LIU; WANG; YANG, 2015; AKIN et al., 2016) e regularização espacial (DANELLJAN et al., 2015b).

Neste trabalho, é proposto um método eficiente de rastreamento de objetos baseado em filtros de correlação. O *framework* proposto faz uso de múltiplos filtros de correlação locais, onde cada filtro é aprendido a partir de uma parte do objeto (filtros locais), e também de um filtro de correlação aprendido usando a região inteira do objeto (filtro global). Todos os filtros locais trabalham em associação com o filtro global para melhorar o rastreamento do objetos. Os filtros locais são usados para rastrear partes individuais do objeto e as posições dos filtros correlação locais são combinadas para determinar a posição do objeto rastreado na cena. Enquanto isso, o filtro global faz o rastreamento do objeto como um todo, e é usado em situações que os filtros locais falham e também para indicar a região de busca para os filtros locais nos quadros subsequentes. Além disso, o uso de filtros locais possibilita lidar com condições desafiadoras de maneira mais robusta, como deformações e variações de escala. O modelo da aparência do objeto utilizado pelos filtros de correlação do método proposto é baseado em feições extraídas de camadas hierárquicas de uma CNN (MA et al., 2015).

1.1 Contribuições

As principais contribuições do *framework* de rastreamento de objetos proposto podem ser resumidas da seguinte forma:

- introduzimos um *framework* de rastreamento de objetos baseado em partes utilizando filtros de correlação;
- propomos um *framework* colaborativo combinando filtros locais e um filtro global;
- propomos um esquema que é eficiente para lidar com variações de escala, oclusões parciais, rotações de objetos e problemas causados por movimentos rápidos dos objetos e desfoque de movimento;
- apresentamos um avanço nos resultados de rastreamento em comparação com métodos representantes do estado da arte;

1.2 Organização da Tese

O restante desta proposta está organizado da seguinte forma:

A Seção 2 apresenta os trabalhos relacionados mais relevantes na área de rastreamento de objetos. Na seção 3 temos alguns conceitos sobre rastreamento de objetos através de imagens/vídeos e os fundamentos teóricos envolvidos no estudo e desenvolvimento do método de rastreamento proposto. A Seção 4 descreve o método proposto, e a seguir, na Seção 5, são discutidos os experimentos e resultados obtidos. Por fim, as conclusões são apresentadas na Seção 6.

2 TRABALHOS RELACIONADOS

Esta seção apresenta um resumo das principais técnicas relacionadas com o rastreamento de objetos utilizando sequências de imagens de vídeo. Primeiramente será apresentada uma visão geral sobre rastreamento de objetos e as categorias de métodos existentes. Em seguida, será apresentado uma breve descrição dos métodos mais utilizados e de novas abordagens propostas recentemente na área.

O rastreamento de objetos através de vídeos é uma das tarefas mais desafiadoras no campo da visão computacional, estando relacionado a diversos tipos de aplicações, como processamento de vídeo, vigilância, controle de tráfego e robótica (SMEULDERS et al., 2014; BOUVIE et al., 2013). A proliferação de computadores potentes, a disponibilidade de câmeras de vídeo de baixo custo e de alta qualidade e a crescente necessidade de análise automatizada de vídeo gerou um grande interesse em algoritmos de rastreamento de objetos. De forma simples, o rastreamento pode ser definido como o problema de estimar a trajetória de um objeto no plano da imagem conforme ele se move ao redor de uma cena (YILMAZ; JAVED; SHAH, 2006).

De acordo com os métodos de modelagem de aparência do objeto, algoritmos de rastreamento geralmente podem ser divididos em duas categorias: métodos generativos e discriminativos. Rastreadores generativos realizam rastreamento através da busca de uma janela na imagem (*patch*) que apresente a melhor correspondência, já os métodos discriminativos usam técnicas para aprender a distinguir o alvo do plano de fundo da imagem (HU et al., 2017; CHEN; HONG; TAO, 2015a).

2.1 Métodos Generativos

Métodos de rastreamento generativos geralmente procuram pela região da imagem mais provável de corresponder ao objeto alvo de acordo com um modelo pré-treinado para minimizar o erro de reconstrução. Os modelos comumente usados incluem o uso de subespaço e templates. Ross et al. (ROSS et al., 2007) propôs o rastreador *Incremental Visual Tracker* (IVT), que é um método baseado em aparência que aprende incrementalmente um modelo de aparência de baixa dimensão de maneira *on-line* para se adaptar as mudanças de aparência do objeto. Adam et al. (ADAM; RIVLIN; SHIMSHONI, 2006) explorou a estrutura de dados do histograma integral e construiu o método *Frag-Track* (Frag), que não utiliza um modelos, funcionando apenas através da combinação do mapa

Tabela 2.1: Características e problemas tratados por diferentes métodos. Comparativo das diferentes feições e problemas que alguns dos métodos conseguem lidar.

Método	Feições	Baseados em Partes	Tratamento para Escala	Tratamento para Oclusão	Tempo Real (CPU)
Struck (HARE et al., 2016)	Haar	Não	Sim	Não	Sim
TLD (KALAL; MIKOLAJCZYK; MATAS, 2012)	Intensidade dos Pixels	Não	Sim	Sim	Sim
MIL (BABENKO; Ming-Hsuan Yang; BELONGIE, 2009)	Haar-like	Sim	Não	Sim	Sim
MEEM (ZHANG; MA; SCLAROFF, 2014)	Intensidade LAB + Local Rank Transform	Não	Não	Sim	10 FPS
MOSSE (BOLME et al., 2010)	Intensidade dos Pixels	Não	Não	Não	Sim
KCF (HENRIQUES et al., 2015)	HoG	Não	Não	Não	Sim
Danelljan et al. (DANELLJAN et al., 2014a)	HoG	Não	Sim	Não	Sim
SAMF (LI; ZHU, 2014)	HoG + Color Names	Não	Sim	Não	7 FPS
CFIT (HU et al., 2017)	HoG + Color Names + Local Rank Transform	Não	Sim	Sim	Sim
CF2 (MA et al., 2015)	Feições CNN	Não	Não	Não	10 FPS
Fu et al. (FU et al., 2020)	HoG + Color Names	Não	Não	Sim	Sim
Feng et al. (FENG et al., 2019)	HoG	Não	Sim	Sim	6 FPS
Liu et al. (LIU; WANG; YANG, 2015)	HoG	Sim	Sim	Sim	30 FPS
Li et al. (LI; ZHU; HOI, 2015)	HoG	Sim	Sim	Sim	4 FPS
DPCF (AKIN et al., 2016)	HoG + Color Names	Sim	Sim	Sim	20 FPS
Liang et al. (LIANG et al., 2019)	HoG	Não	Sim	Sim	Sim
CLIP (LIU et al., 2020)	HoG + Color Names + Histogramas de cor	Não	Sim	Sim	30 FPS
Método Proposto	Feições CNN	Sim	Sim	Sim	1 FPS

de votos de vários *patches*. O método *circulant sparse tracker* (CST) (ZHANG; BIBI; GHANEM, 2016) foi o primeiro a usar estrutura circulante para representação esparsa em *frameworks* baseados em filtro de partículas. Através do refinamento das partículas amostradas de forma eficiente, ele consegue reduzir drasticamente o número de partículas necessárias para o rastreamento.

A maior desvantagem dos métodos generativos é a necessidade de um grande número de amostras necessárias para treinar um modelo confiável, e também não utilizarem completamente a informação do fundo da cena. Esses fatores fazem com que o uso desses métodos generativos sejam limitados.

2.2 Métodos Discriminativos

Recentemente, dentre os vários métodos de rastreamento de objetos, os métodos discriminativos têm se destacado em *benchmarks* de rastreamento (WU; LIM; YANG, 2013; WU; LIM; YANG, 2015). Os métodos de rastreamento discriminativos consideram o rastreamento de objetos como um problema de classificação binária, empregando técnicas para aprender um classificador binário que consiga determinar o limiar ótimo para discriminar o objeto alvo do fundo da cena.

A Tabela 2.1 mostra uma comparação entre alguns dos diferentes métodos de rastreamento e os tipos de feições utilizadas por cada um, assim como outras diferenças, como se são métodos baseados em partes, se possuem tratamento para lidar com variação de escala e oclusão, e se funcionam em tempo real.

Duas abordagens representativas dos métodos discriminativos são rastreamento

por detecção (*Tracking-by-Detection*) e filtros de correlação (*Correlation Filters*):

- Métodos baseados em rastreamento por detecção (*Tracking-by-Detection*):
  Nesta abordagem um classificador *on-line* é treinado usando um algoritmo estatístico de aprendizado de máquina, então esse classificador é usado para dar um *score* para regiões candidatas em um novo quadro do vídeo, sendo a região (*patch*) com a pontuação mais alta considerada como a resposta de saída do rastreamento. O método Struck (HARE et al., 2016) apresenta o problema de rastreamento como uma predição de saída estruturada. Ele usa uma máquina de vetores de suporte (SVM - *Support Vector Machine*) de saída estruturada para aprender uma função de predição para estimar diretamente as transformações de um objeto em quadros consecutivos, integrando aprendizado e rastreamento para evitar estratégias de atualização *ad-hoc*. A abordagem do método TLD (*Tracking–learning–detection*) (KALAL; MIKOLAJCZYK; MATAS, 2012) propõe um *framework* de rastreamento de longo prazo que decompõe o processo de rastreamento em três subtarefas: rastreamento, aprendizado e detecção. Essas sub-tarefas são tratadas por componentes únicos que operam simultaneamente. Assim, eles podem compensar um ao outro e, por exemplo, o componente rastreador pode fornecer dados de treinamento com marcações não muito precisas para o detector e melhorá-lo durante a execução, ou ainda, o detector pode reiniciar o rastreador para minimizar erros de rastreamento caso necessário. Ao mesmo tempo, o componente de aprendizado estima o erro de detecção e então atualiza o modelo. Um sistema de rastreamento que usa um algoritmo de aprendizado de múltipla instância *on-line* (MIL - *Multiple Instance Learning*) é proposto em Babenko, Ming-Hsuan Yang and Belongie (2009). O *framework* contém três componentes: uma representação da imagem, um modelo de aparência e um modelo de movimento. A representação da imagem usa um conjunto de feições *Haar-like* (VIOLA; JONES, 2001) computados para cada *patch* da imagem, e um classificador discriminativo compõe o modelo de aparência, que é atualizado usando instâncias positivas e negativas a partir de um conjunto de amostras. O método MEEM (*Multiple Expert Entropy Minimization*) (ZHANG; MA; SCLAROFF, 2014) apresenta um esquema de restauração para corrigir atualizações indesejáveis no modelo. Um rastreador e o histórico de rastreamento constituem um conjunto de especialistas, e caso necessário, o melhor especialista é selecionado para restaurar o rastreador atual com base em um critério de entropia mínima. O rastreador base utiliza um algoritmo SVM *on-line* e uma técnica para mapeamento explícito das

feições para realizar a atualização do modelo. Dessa forma, o problema de desvio do modelo no rastreamento *on-line* pode ser resolvido de forma eficaz. Embora esses métodos possam geralmente alcançar resultados promissores, a estratégia de amostragem aleatória utilizadas por eles produz um conjunto limitado de amostras de treinamento positivas e negativas em torno do objeto. Consequentemente, esses métodos são limitados pelo alto custo de tempo e por amostras de treinamento inadequadas quando consideramos uma grande área de busca e grandes objetos.

- Métodos baseados em filtros de correlação:

  Para resolver essas limitações de alto custo de tempo e necessidade de um grande número de amostras de treinamento dos métodos baseados em rastreamento por detecção problemas, abordagens baseadas em filtros de correlação discriminativos (DCF) têm sido utilizadas e atraído a atenção nos últimos anos. O DCF é uma técnica supervisionada para aprender um classificador linear ou um regressor linear. Ele alcança alta eficiência computacional com o uso da Transformada Rápida de Fourier, e explora as propriedades da correlação circular para treinamento e detecção eficientes, implicitamente utilizando todas as versões deslocadas de uma amostra base para para produzir amostras de treino suficientes (DANELLJAN et al., 2015b).

  Os métodos baseados em filtros de correlação modelam a aparência dos objetos alvo usando filtros treinados em imagens de exemplo. Depois de usar um filtro de correlação para aprender um modelo de aparência para o alvo rastreado, o alvo é rastreado através da correlação do filtro com uma janela de busca no quadro subsequente, produzindo um mapa de resposta, cujo máximo valor de resposta de correlação indica a localização estimado do alvo. Bolme et al. (BOLME et al., 2010) introduziu o filtro MOSSE (*Minimum Output Sum of Squared Error*), que usa um único canal como feição de entrada (intensidade dos pixels), conseguindo aprender filtros de correlação estáveis inicializados apenas com um único quadro. O método proposto por Bolme et al. (BOLME et al., 2010) foi um dos primeiros a sugerir o uso de um esquema adaptativo para o treinamento, tornando o uso de filtros de correlação uma opção viável e eficiente para o rastreamento e objetos. O método proposto por eles treina um filtro de correlação em um conjunto de amostras de treinamento em níveis de cinza e consegue produzir uma saída estável com velocidades acima de 600 quadros por segundo. Apesar de possuir boa performance, a sua capacidade de discriminação é limitada devido ao uso de somente um único

canal da imagem como feições de entrada.

Henriques et al. (HENRIQUES et al., 2015; HENRIQUES et al., 2012) melhoraram o filtro MOSSE estendendo o uso de um único canal de feições para múltiplos canais e e introduziram métodos de *kernel* para lidar com problemas de não linearidade. O KCF (*Kernelized Correlation Filter*) é baseado no *kernel Ridge Regression* e no uso de matriz circulante, e em vez de utilizar um único canal da imagem como feição, introduz o uso de feições multicanais baseadas em histogramas de gradientes orientados para melhorar o desempenho do filtro. Devido ao seu desempenho e simplicidade, o método KCF se tornou bastante conhecido e utilizado com base para outros métodos baseados em filtros de correlação. Ainda assim, não é robusto o suficiente para lidar com situações mais desafiadoras, como rotações e deformações, e também não consegue lidar com variações de escala.

Danelljan et al. (DANELLJAN et al., 2014b) estenderam o método baseado em filtros de correlação através da utilização de atributos de cores, e também introduziram um método adaptativo para reduzir a dimensionalidade das feições. O trabalho proposto por Danelljan et al. (DANELLJAN et al., 2015b) apresenta uma regularização espacial para mitigar o problema criado pela convolução circular nos filtros de correlação durante o treinamento. Esse problema, conhecido como *boundary effect*, é causado ao se assumir que a imagem é periódica, o que causa descontinuidades nas bordas da imagem que será usada para aprender o filtro de correlação. A regularização proposta utiliza pesos para penalizar os coeficientes do filtro de correlação durante a etapa de aprendizado, com os pesos sendo usados para determinar a importância de cada coeficiente do filtro, conforme a sua localização espacial.

Para lidar com varições de escala, Li e Zhu (LI; ZHU, 2014) propuseram um esquema adaptativo para encontrar a escala ótima que compara amostras multiresolução obtidas do objeto em diversas escalas predefinidas. Além disso, utilizaram três feições complementares para o método KCF obtendo resultados promissores. O método proposto por Zhang et al. (ZHANG et al., 2016) modela simultaneamente os deslocamentos espaciais, variações de escala e transformações de rotação do objeto em uma estrutura de correlação unificada. Os templates alvos são transformados do sistema de coordenadas cartesiano para o sistema de coordenadas Log-Polar, permitindo que o método modele as rotações do objeto no mesmo *framework* que os deslocamentos espaciais e mudanças de escala. Huang et al. (HUANG et al., 2017) integra ao filtro de correlação uma abordagem baseada em um método de

detection proposal, que é amplamente adotado na área de detecção de objetos. para obter adaptabilidade de escala e *aspect ratio*.

O método CFIT (HU et al., 2017) combina filtros de correlação 2D e 1D para estimar a localização e a escala do alvo. Adicionalmente, eles combinam feições complementares para melhorar a discriminação do método, incluindo *HoG*, *Color Names* (DANELLJAN et al., 2014b; WEIJER et al., 2009) e histogramas locais. A integração de diferentes feições e a utilização de filtros em diferentes escalas ajudam a melhorar a performance geral do método, mas como desvantagem podemos citar a necessidade de uma estratégia melhor para realizar a atualização dos filtros e também os tipos de feições utilizadas, que apesar de funcionarem bem não são tão robustas quanto as baseadas em redes neurais, por exemplo. Apesar de diversos métodos adaptativos para escala terem sido propostos, muitos tem uma eficiência baixa e computacionalmente pesados, sendo ainda um problema desafiador estimar variações de escala de maneira rápida e precisa (HU et al., 2017).

O trabalho em Ma et al. (MA et al., 2015) explora o uso de feições extraídas de redes neurais convolucionais profundas. O uso dessas feições proporciona uma melhor discriminação dos objetos na cena, permitindo lidar melhor com objetos em diferentes resoluções e também com rotações. Apesar disso, ainda existe certa limitação em situações envolvendo oclusões, deformações e variações de escala.

Fu et al. (FU et al., 2020) também apresenta uma nova estratégia para regularização espacial, aplicando três tipos de penalidades aos coeficientes do filtro. Na região considerada com confiável, a penalidade é a mesma e baixa para que o alvo seja aprendido corretamente. Na região não confiável dentro do *boundig box* atual, as penalidades dos coeficientes dependem de suas posições espaciais. Na região não confiável fora do *boundig box*, a penalidade é a mesma para todos os coeficientes e alta, de forma a impedir que o filtro aprenda o *background*. Feng et al. (FENG et al., 2019) propõem um esquema para estender métodos baseados em filtros de correlação com regularização espacial para incorporar a saliência do alvo e fazer o mapa de pesos de regularização variar dinamicamente quadro a quadro, capturando todas as variações de forma do alvo. A saliência do objeto e a confiabilidade do rastreamento no domínio do espaço-tempo são indicadas por uma função de energia, que é usada para controlar o processo de atualização online do mapa de pesos, através de um método eficiente baseado em *level-sets*.

Outra abordagem comum é o uso de métodos de rastreamento baseados em partes

para rastrear um objeto (LI; ZHU; HOI, 2015; LIU; WANG; YANG, 2015; AKIN et al., 2016). Em vez de aprender um modelo para todo o alvo, o alvo é rastreado utilizando várias partes locais, tornando possível continuar rastreando o objeto mesmo quando ocorrer uma oclusão parcial, uma vez que partes do objeto permanecem visíveis. Liu et al. (LIU; WANG; YANG, 2015) combina um rastreador baseado em filtro de correlação com um *framework* de inferência Bayesiana para construir um rastreador baseado em partes. A abordagem proposta por eles utiliza um método de seleção de partes discriminativo para selecionar as partes mais robustas dentre um conjunto de amostras e, em seguida, combina os mapas de resposta individuais de cada parte usando um esquema adaptativo para atribuir pesos para cada uma das partes que são utilizadas para representar o objeto inteiro. De maneira similar, Li et al. (LI; ZHU; HOI, 2015) apresentam um *framework* para identificar *patches* confiáveis baseados em um método de filtro de partículas.

Akin et al. (AKIN et al., 2016) introduz um modelo colaborativo local-global, que combina filtros de correlação e um rastreador deformável baseado em partes para melhorar o desempenho do rastreamento. O objeto é representado por filtros locais que fornecem uma estimativa inicial da posição do objeto, que é utilizada por um filtro global para determinar a posição final. Apesar de rastrear partes do objeto separadamente, são utilizados dois filtros para cada objeto e apenas em uma direção (vertical ou horizontal), fazendo com que ele não seja tão robusto em casos de oclusão, dependendo de qual parte do objeto está sendo ocluída. Apesar de possuir um bom desempenho para detectar variações de escala, em algumas situações não é capaz de determinar corretamente a posição e escala do objeto devido às feições utilizadas e também ao número limitado de filtros.

Yuan et al. (YUAN et al., 2020) propôs uma rede adaptativa de filtros convolucionais estruturais para rastrear objetos utilizando um conjunto de filtros locais, que são gerados para capturar as estruturas locais do objeto alvo consideradas mais discriminativas. Em seguida, cada um dos mapas de respostas dos filtros locais são integrados em um mapa de resposta único, que é então combinado com o filtro base para produzir a localização final do alvo. Dessa forma, eles conseguem bom desempenho em situações com movimento muito rápidos e também com variações de escala, tendo como ponto fraco situações que apresentam plano de fundo muito similar com os objetos ou quando o alvo fica fora da cena por alguns instantes.

O trabalho de Liang et al. (LIANG et al., 2019) também propõe uma abordagem

para melhorar o rastreamento de objetos baseados em filtros de correlação em situações de oclusão e variações de escala. É proposto um método mais rápido e eficiente para criar uma representação em pirâmide para representar as feições *hog* em múltiplas escalas e também uma modificação na função de correlação para permitir que mais informação do *background* ao redor do objeto seja aprendida com a finalidade de melhorar o rastreamento quando ocorrer uma situação de oclusão. Através de um cálculo de regressão realizado nessa pirâmide multiescala utilizando filtros de correlação multiescala é estimada a melhor escala e melhor posição para o objeto rastreado. Apesar de ser um método bastante rápido e resolver o problema de variação de escala em alguns casos, o uso de feições simples e da pirâmide multiescala não são suficientes para melhorar significativamente os resultados, fazendo com que ele tenha um desempenho abaixo de métodos mais sofisticados em situações de oclusão e variação de escala mais complexas.

Liu et al. (LIU et al., 2020) propõem um método de rastreamento baseado em filtros de correlação composto por três módulos principais. O primeiro módulo é um filtro para a translação, que combina feições complementares para melhor lidar com variações de aparências do objeto. Este filtro é combinado com um segundo módulo, que é um filtro utilizado para estimar as variações de escala. Por fim, é utilizado também um módulo para correções de erros, baseado em *Region Proposal Generation* (ZITNICK; DOLLÁR, 2014), usado para produzir regiões candidatas usadas na detecção do objeto em casos que o rastreamento é perdido devido a problemas com oclusões e deformações. Para conseguir um filtro de correlação que fosse rápido no rastreamento, os autores optaram por usar feições mais simples, baseadas em *HOG*, *Color Names*, e histogramas de cor, que podem não ser tão robustas quanto aquelas baseadas em CNNs, por exemplo, mas são processadas mais rapidamente.

Durante a elaboração dessa tese foram realizados diversos trabalhos envolvendo rastreamento de objetos (BARCELLOS; SCHARCANSKI, 2020) e, também rastreamento de veículos (BARCELLOS; GOMES; SCHARCANSKI, 2016; BARCELLOS et al., 2015; GOMES; BARCELLOS; SCHARCANSKI, 2017; GOMES; BARCELLOS; SCHARCANSKI, 2018). O rastreamento de veículos é uma aplicação do rastreamento de objetos bastante desafiadora e de grande importância, pois permite obtermos informações sobre as condições de tráfego, que podem ser utilizadas para tarefas de gestão do tráfego, como sincronizar semáforos, auxiliar os motoristas na seleção de rotas, e auxiliar os governos no planejamento da expansão do sistema de trânsito e na construção de novas estradas.

Os métodos de rastreamento de veículos desenvolvidos apresentaram algumas limitações. Em virtude do ângulo de captura de alguns vídeos, ocorrem algumas situações de oclusões quando os veículos estão muito próximos, geralmente quando estão parados nos sinais de trânsito, ou quando existem veículos grandes, como ônibus, nas pistas mais próximas da câmera. Outro problema é a variação de escala dos objetos conforme eles se aproximam ou se afastam da câmera.

Para melhorar essa etapa de rastreamento, decidimos desenvolver um método mais robusto de rastreamento, que tivesse um bom desempenho não apenas para o rastreamento de veículos mas também para o rastreamento de outros objetos em geral. Como visto anteriormente, os métodos baseados em filtros de correlação apresentam uma alta eficiência e robustez, impulsionando significativamente o desenvolvimento de métodos de rastreamento. Por esse motivo, a escolha pelo desenvolvimento de um método de rastreamento que fosse baseado em filtros de correlação se mostrou a melhor alternativa. Muitos dos métodos baseados em filtros de correlação existentes usam um único tipo de feição ou feições que são insuficientes para detectar as diversas variações de aparência dos objetos rastreados. Outra deficiência é a forma como os modelos são atualizados, que muitas vezes acaba degradando o modelo e levando a resultados incorretos. Consequentemente, alguns métodos funcionam bem para estimar a forma e as deformações do objeto, e outros funcionam bem para estimar a localização do alvo, mas não ambos ao mesmo tempo. Neste trabalho, desenvolvemos um novo método com novas técnicas para aprimorar os métodos baseados em filtros de correlação e lidar adequadamente com essas duas questões, e também seja robusto a outras dificuldades encontradas ao longo do projeto, como distorções na lente da câmera, variações de iluminação, sombras, oclusões, e desfoques causados pelo movimento dos objetos.

3 FUNDAMENTOS TEÓRICOS

Nesta seção serão apresentados alguns conceitos teóricos utilizados na área de rastreamento de objetos em com uso de sequências de imagens e visão computacional. Serão apresentadas as definições teóricas e uma breve introdução das técnicas e métodos que foram utilizados na elaboração do método proposto, assim como outros fundamentos relevantes na área.

3.1 Tipos de Feições para Rastreamento de Objetos

A seleção das feições corretas desempenha um papel crítico no rastreamento. Em geral, a propriedade mais desejável de um recurso visual é sua singularidade, de modo que os objetos podem ser facilmente distinguidos no espaço de feições. A seleção de feições está diretamente relacionada à representação do objeto. Por exemplo, a cor é usada como um feição para representações de aparência baseadas em histogramas, enquanto para representação baseada em contorno, as bordas do objeto são geralmente usadas como feições (YILMAZ; JAVED; SHAH, 2006). Em geral, muitos algoritmos de rastreamento usam uma combinação desses recursos. A seguir são listados algumas feições utilizadas em métodos baseados em filtros de correlação.

3.1.1 Níveis de Cinza

A feição mais básica comumente utilizada é simplesmente o utilizar valores de intensidade (grayscale) obtidos a partir de uma imagem contendo apenas um canal com os níveis de intensidades de cada pixel. Um classificador pode ser treinado usando um único *patch* de imagem em tons de cinza que é centralizado em torno do alvo, e os valores de intensidade dos pixels são usados diretamente para encontrar o melhor correspondente para esse *patch* em outra imagem ou sequência do vídeo.

3.1.2 Cores

Entre todos as feições, a cor é uma das mais amplamente usadas para rastreamento. Apesar de sua popularidade, a maioria das faixas de cores são sensíveis à variação de

iluminação(YILMAZ; JAVED; SHAH, 2006). Portanto, em cenários onde esse efeito é inevitável, outras feições são incorporados à aparência do objeto do modelo.

O espaço de cores RGB (vermelho, verde, azul) é geralmente usado para representar a cor. No entanto, o espaço RGB não é um espaço de cores perceptualmente uniforme, ou seja, as diferenças entre as cores no espaço RGB não correspondem às diferenças de cores percebidas pelos humanos(YILMAZ; JAVED; SHAH, 2006). Além disso, as dimensões RGB são altamente correlacionadas. Para lidar com essas situações, podem ser utilizados o L*u*v e o L*a*b, que são espaços de cores perceptualmente uniformes, ou ainda o HSV (Matiz, Saturação, Valor), que é um espaço de cores aproximadamente uniforme.

3.1.3 Histogramas de Cor

Histogramas de cores também podem ser utilizados como feições para o rastreamento de objetos. Segundo Gonzalez and Woods (2002), dada uma imagem digital com níveis de intensidades no intervalo [0, L-1], o histograma dessa imagem é uma função discreta $h(r_k) = n_k$, onde r_k é o k-ésimo valor de intensidade e n_k é o número de pixels na imagem com intensidade r_k. É comum utilizar o histograma normalizado da imagem, que é obtido dividindo cada um de seus componentes pelo número total de pixels na imagem, denotado por $n = MN$, onde M e N são as dimensões das linhas e colunas da imagem, respectivamente. Assim, o histograma normalizado é uma estimativa da probabilidade de ocorrência do nível de intensidade r_k na imagem, e é dado por:

$$p_r(r_k) = \frac{n_k}{n} \tag{3.1}$$

onde:

$0 \leq r_k \leq 1$

$k = 0, 1, \ldots, L-1$, onde L é o número de níveis de intensidade na imagem.

$p_r(r_k)$ = probabilidade do k-ésimo valor de intensidade.

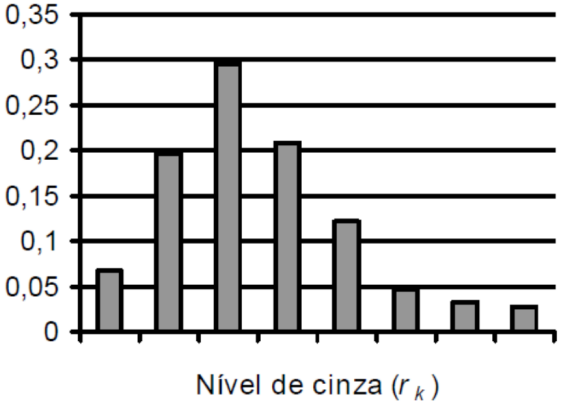
n_k = número de pixels cujo nível de intensidade corresponde a k.

A Tabela 3.1 apresenta os dados que correspondem a uma imagem de 128×128 pixels, com 8 níveis de cinza. O numero de pixels que corresponde a cada um dos níveis de cinza é indicado pela segunda coluna, e as respectivas probabilidades aparecem na terceira coluna. A representação gráfica deste histograma pode ser observada na Figura 3.1.

Tabela 3.1: Exemplo de histograma para imagem de 128x128 pixels e 8 níveis de cinza.

Nível de cinza (rk)	nk	pr(rk)
0	1120	0.068
1/7	3214	0.196
2/7	4850	0.296
3/7	3425	0.209
4/7	1995	0.122
5/7	784	0.048
6/7	541	0.033
1	455	0.028
Total	16384	1

Figura 3.1: Exemplo histograma da imagem com 8 níveis de cinza.



3.1.4 *Color Names*

Atributos de cores, ou *Color Names* (CN), são rótulos de cores linguísticos atribuídos por humanos para representar as cores do mundo. Em um estudo linguístico, concluiu-se que a língua inglesa contém onze termos básicos para cores: *black, blue, brown, grey, green, orange, pink, purple, red, white and yellow* (DANELLJAN et al., 2014b). Com base nesse estudo, foi desenvolvido uma operação que associa observações de cores RGB a esses rótulos de cores linguísticos.

O método proposto por Weijer et al. (WEIJER et al., 2009) é bastante utilizado para realizar esse mapeamento de cores para rótulos de cores. O modelo desenvolvido por eles para fazer esse mapeamento é aprendido automaticamente a partir de imagens recuperadas com a pesquisa de imagens do Google, e mapeia os valores RGB para uma representação de cor probabilística de 11 dimensões, que correspondem a cada uma das cores (DANELLJAN et al., 2014b).

3.1.5 *Scale Invariant Feature Transform* - SIFT

Um descritor SIFT *Scale Invariant Feature Transform* é construído a partir de gradientes da imagem, usando a magnitude e também a orientação. O descritor é um conjunto de histogramas de gradientes da imagem que são então normalizados para suprimir os efeitos de variação na intensidade da iluminação. Esses histogramas expõem tendências espaciais gerais nos gradientes da imagem, mas suprimem detalhes (FORSYTH; PONCE, 2002). Ele é obtido dividindo primeiro um *patch* da imagem em uma grade de $n \times n$. Em seguida, subdividimos cada elemento da grade em uma subgrade de $m \times m$ subcélulas. No centro de cada subcélula, calculamos uma estimativa do gradiente. A estimativa do gradiente é obtida como uma média ponderada dos gradientes ao redor do centro da célula, ponderando cada uma por $(1 - d_x/s_x)(1 - d_y/s_y)/N$, onde d_x (respectivamente d_y) é a distância em x (respectivamente y) do gradiente ao centro da subcélula, e s_x (respectivamente s_y) é o espaçamento x (respectivamente y) entre os centros da subcélula. Assim, os gradientes contribuem para mais de uma subcélula, de modo que um pequeno erro na localização do centro do *patch* leva a uma pequena mudança no descritor.

Essas estimativas de gradiente são então utilizadas para produzir histogramas. Cada elemento da grade possui um histograma de orientação de q células. A magnitude de cada gradiente é acumulada na célula do histograma correspondente à sua orientação, e a magnitude é ponderada por uma gaussiana considerando a distância do centro do *patch*, usando um desvio padrão correspondente a metade do tamanho do *patch*. Cada histograma é concatenado em um vetor $n \times n \times q$, que é então normalizado para ter comprimento unitário (FORSYTH; PONCE, 2002).

3.1.6 *Histogram of Oriented Gradients* - HoG

A feição HoG (*Histogram of Oriented Gradients*) (DALAL; TRIGGS, 2005) é um descritor que computa das orientações dos gradientes em uma imagem, produzindo ao final um vetor com os histogramas extraídos da imagem, e pode ser considerado uma variante do SIFT (FORSYTH; PONCE, 2002). Nesse caso também é computado o histograma das orientações do gradiente em células, mas agora o processo tenta identificar bordas de alto contraste. Podemos obter informações de contraste contando as orientações do gradiente com pesos que refletem o quão significativo um gradiente é em comparação com outros gradientes na mesma célula. Assim, em vez de normalizar as contribuições

de gradiente sobre toda a vizinhança, normalizamos apenas em relação aos gradientes próximos.

Seja $\|\nabla I_x\|$ a magnitude do gradiente no ponto x de uma imagem, C a célula cujo histograma desejamos calcular e $W_{x,C}$ o peso que usaremos para a orientação em x desta célula. Este peso pode ser definido como:

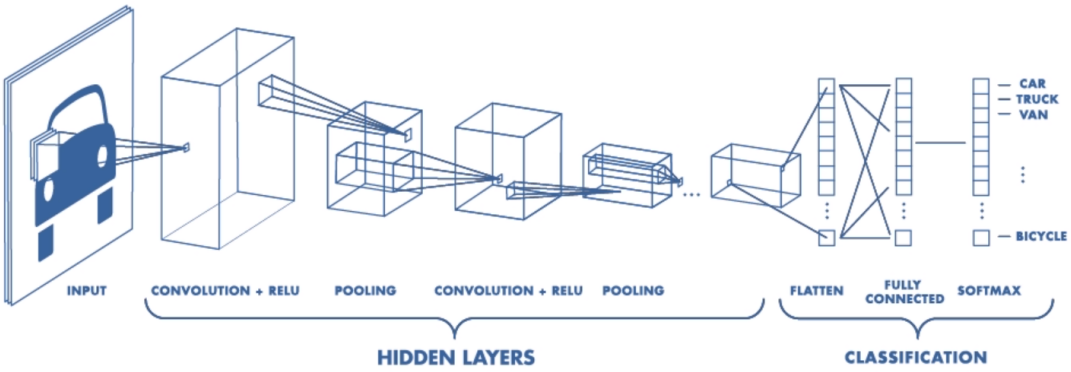
$$W_{x,C} = \frac{\|\nabla I_x\|}{\sum_{u \in C} \|\nabla I_u\|}. \tag{3.2}$$

Essa equação compara a magnitude do gradiente com os outros na célula, de modo que gradientes que são grandes em comparação com seus vizinhos tenham um peso grande.

3.1.7 Extração de Feições Utilizando Redes Neurais Convolucionais

Algoritmos baseados em redes neurais convolucionais (CNNs - *Convolutional Neural Networks*) apresentam uma abordagem muito poderosa devido ao fato de conseguirem criar algumas abstrações de baixo nível das imagens, como linhas, círculos, bordas, e então combiná-los iterativamente em algum objeto que queremos detectar.

Figura 3.2: Exemplo de uma rede neural de classificação.



Fonte: (PATEL; PINGEL, 2017)

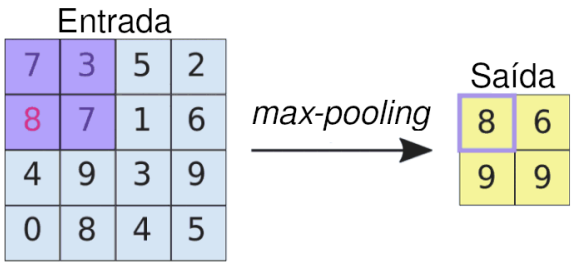
Na Figura 3.2 podemos ver uma ilustração de uma rede neural de classificação. Estamos mais interessados na parte das camadas ocultas (*hidden layers*). Como podemos ver, a rede tem várias combinações de convoluções seguidas por uma camada de *pooling*.

Usualmente, as entradas de uma rede neural são matrizes tridimensionais, i.e., imagens com três canais RGB, com valores que correspondem aos valores de cada pixel. As convoluções funcionam como filtros (referenciados também como neurônios ou kernels) que processam pequenas janelas de pixels de cada vez, chamadas campos receptivos

ou *receptive fields*, até se deslocar por toda a imagem e capturar os detalhes mais relevantes. Esse filtro é composto por uma matriz de números, chamados de pesos ou parâmetros, que são atualizados a cada nova entrada com o processo de *backpropagation*. Depois de deslizar o filtro sobre todos os locais, obtemos uma matriz de números conhecida como mapa de ativação ou mapa de feições (NIELSEN, 2015).

Além das camadas de convolução, a as CNNs também possuem camadas chamadas de *pooling*, que são geralmente usadas imediatamente após as camadas de convolução. A camada de *pooling* pega a saída de cada mapa de feições da camada convolucional e cria uma sub-amostragem ou versão reduzida do mapa de feições (NIELSEN, 2015). Cada unidade na camada de *pooling* pode resumir a informação de uma região de, por exemplo, 2×2 neurônios na camada anterior. Um procedimento comum para isso é conhecido como *max-pooling*, que passará para a saída apenas o valor máximo de cada uma das regiões da camada anterior, conforme ilustrado na Figura 3.3. Um grande benefício é que esse procedimento reduz o número de feições e, portanto, reduz o número de pesos necessários nas camadas posteriores (custo computacional), e também ajuda a controlar o *overfitting*.

Figura 3.3: Exemplo do procedimento de *max-pooling*.



A camada final de conexões na rede é uma camada totalmente conectada. Ou seja, essa camada conecta todas as saídas a partir da última camada de *poolling* até um vetor com N dimensões para realizar o processo de classificação, onde N é o número de classes que foram usadas no treinamento do modelo. Cada número neste vetor N-dimensional representa a probabilidade de uma determinada classe.

Existem três formas de usar CNNs para analises de imagens (PATEL; PINGEL, 2017):

- Treinar o modelo do zero: é a forma que apresenta melhores resultados, mas também é a que mais exige em termos de recursos computacionais e também de dados para treinamento;
- *Transfer learning*: é usado um modelo treinado em um conjunto de dados substan-

cialmente grande e depois utilizar os pesos aprendidos por esse modelo pré-treinado para resolver um outro problema, utilizando um conjunto de dados diferente.

- Usar a CNN pré-treinada para extrair feições: a ideia é utilizar as feições detectadas pelas diversas camadas de uma CNN para treinar um modelo baseado em aprendizado de máquina. Esta é forma que exige menos recursos computacionais, e as feições extraídas possibilitam atingir melhores resultados em algumas tarefas do que seria possível com a utilização de outras feições convencionais, como as apresentadas nas seções anteriores.

As CNNs podem conter centenas de camadas ocultas, e cada uma delas aprende a detectar diferentes feições em uma imagem. Cada camada aumenta a complexidade da feição aprendida na imagem. A Figura 3.4 ilustra exemplos de mapas de feições extraídos de diferentes camadas de uma CNN. A primeira camada, por exemplo, aprende a detectar feições mais simples, como bordas, já a última aprende a detectar formas mais complexas (PATEL; PINGEL, 2017).

Figura 3.4: Exemplo de feições extraídas de diferentes camadas de uma CNN.

Fonte: (PATEL; PINGEL, 2017)

3.2 Filtros de Correlação

O *framework* geral dos métodos de rastreamento baseados em filtro de correlação pode ser definido da seguinte maneira: Inicialmente, o filtro de correlação é treinado com um *patch* da imagem recortado de uma determinada posição do alvo no primeiro quadro do vídeo. A seguir, em cada quadro, o *patch* na posição predita anteriormente é recortado para realizar a detecção, e feições podem ser extraídas dos dados de entrada brutos. Uma função de janelamento utilizando uma janela do cosseno é geralmente aplicada ao *patch* para remover descontinuidades nas bordas da imagem (BOLME et al., 2010; GONZALEZ; WOODS, 2002).

Subsequentemente, operações de correlação são realizadas substituindo as convoluções por multiplicações elemento a elemento usando a Transformada Discreta de

Fourier (DFT). Na prática, a *DFT* de um vetor é calculada pelo eficiente algoritmo *Fast Fourier Transform* (*FFT*). Como resultado, um mapa de confiança espacial, ou mapa de resposta, pode ser obtido usando a Transformada Inversa de Fourier (*Inverse Fast Fourier Transform - IFFT*). A posição com um valor máximo neste mapa (pico) é referido como a nova posição do alvo. Em seguida, a aparência do alvo na posição estimada é extraída para treinamento e atualização do filtro de correlação (CHEN; HONG; TAO, 2015a). A Figura 3.5 exibe uma visão geral de um método baseado em filtros de correlação.

Descrevendo matematicamente, seja x a entrada do estágio de detecção e h o filtro de correlação. Na prática, x pode ser simplesmente um *patch* da imagem ou feições extraídas da imagem. Considerando que o símbolo $\hat{}$ representa a transformada de Fourier de um vetor, conforme o Teorema da Convolução, a convolução é igual à multiplicação elemento a elemento no domínio da frequência

$$x \otimes h = \mathscr{F}^{-1}(\hat{x} \odot \hat{h^*}), \tag{3.3}$$

onde $\mathscr{F}^{-1}$ representa a operação de transformada inversa de Fourier, $\odot$ denota a multiplicação elemento elemento e $*$ indica o complexo conjugado. O resultado de saída é a correlação esperada entre x e h, em forma de um mapa de confiança.

Para treinar o filtro, devemos definir uma saída de correlação desejada. Usando a nova instância x' do alvo, o filtro de correlação h deve satisfazer a seguinte equação:

$$y = \mathscr{F}^{-1}(\hat{x'} \odot \hat{h^*}) \tag{3.4}$$

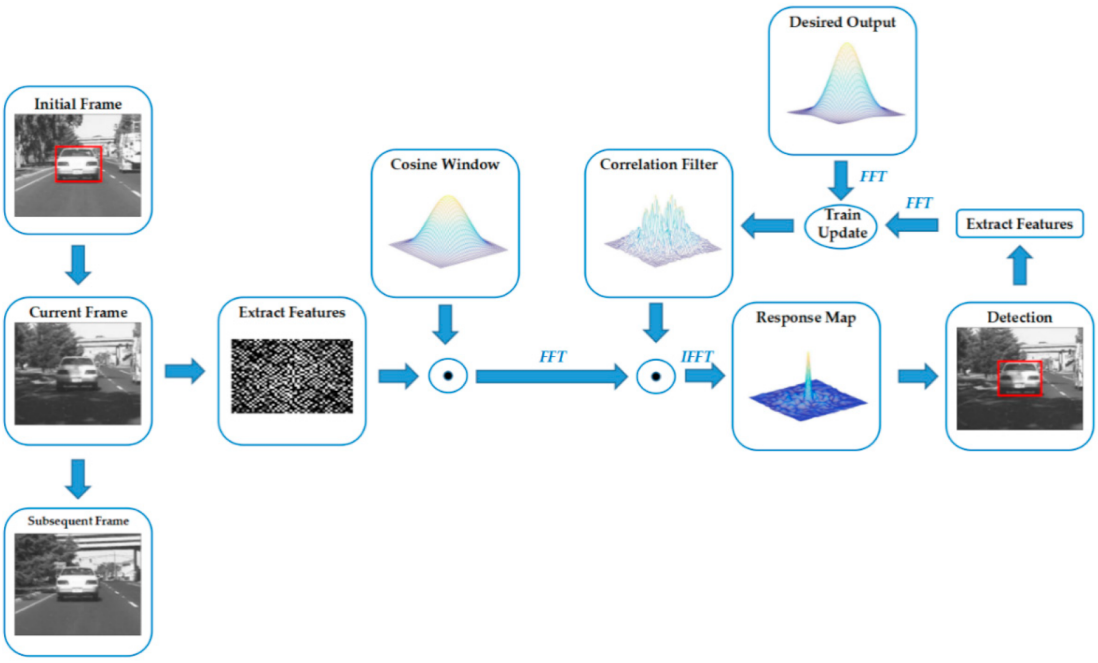
e assim:

$$\hat{h^*} = \frac{\hat{y}}{\hat{x^*}} \tag{3.5}$$

onde $\hat{y}$ é a DFT de y e a divisão é computada elemento a elemento.

Em termos computacionais, a complexidade da convolução para uma imagem de tamanho $n \times n$ é $O(n^4)$, enquanto que a multiplicação elemento a elemento usando a FFT somente requer $O(n^2 \log n)$ (CHEN; HONG; TAO, 2015a). Assim, o uso da FFT proporciona uma aceleração significante ao método.

Figura 3.5: Visão geral de um método baseado em filtros de correlação. Inicialmente, o filtro de correlação é treinado pelo primeiro quadro com uma *bounding box* inicial e a saída desejada. Em seguida, feições são extraídas do quadro atual e multiplicadas por uma janela de cosseno para enfatizar a região central e eliminar as descontinuidades nas bordas da imagem. As feições são então transformadas para o domínio da frequência, utilizando a *FFT*. Um mapa de resposta é obtido multiplicando o filtro de correlação e as feições extraídas. O mapa de resposta é então transformado para o domínio do tempo aplicando a *IFFT*. Finalmente, a posição do valor máximo do mapa de resposta é considerada como a posição central do alvo no quadro atual. Novas feições são então extraídas do resultado detectado para treinar e atualizar o filtro de correlação.



Fonte: Yang et al. (2019)

3.2.1 Deslocamentos Cíclicos

Para simplificar as notações, será utilizado um exemplo de sinal de apenas uma dimensão, mas os conceitos apresentados aqui podem ser generalizados para imagens 2-D e de múltiplos canais.

Considere um vetor $n \times 1$ representando um *patch* com o objeto de interesse, denotado x. Ele será referido como a amostra base. O objetivo é treinar um classificador com a amostra base (um exemplo positivo) e várias amostras virtuais obtidas por meio de sua translação (que servem como exemplos negativos). Podemos modelar translações unidimensionais deste vetor por um operador de deslocamento cíclico, que é a matriz de permutação:

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{3.6}$$

O produto $Px = [x_n, x_1, x_2, ..., x_{n-1}]^T$ muda x por um elemento, modelando uma pequena translação. Podemos encadear u deslocamentos para obter uma translação maior usando a potência da matriz $P^u x$. Um u negativo fará o deslocamento na direção reversa. Um exemplo para uma imagem 2-D é mostrado na Figura 3.6. A formulação no domínio de Fourier permite treinar um método de rastreamento com todos os deslocamentos cíclicos possíveis de uma amostra base, tanto vertical quanto horizontal, sem iterá-los explicitamente (HENRIQUES et al., 2015). Podemos ver que o último elemento é deslocado para o início, induzindo alguma distorção em relação a uma translação verdadeira. Esses artefatos causados pela descontinuidade das bordas podem ser vistos (parte superior da imagem mais à esquerda), mas são atenuados pela aplicação de uma janela do cosseno ao *patch*. Um sinal 1-D deslocado horizontalmente com este modelo é ilustrado na Figura 3.7.

O fato de que uma grande porcentagem dos elementos de um sinal ainda são modelados corretamente, mesmo para deslocamentos relativamente grandes, explica o motivo dos deslocamentos cíclicos funcionarem bem na prática. Devido à propriedade cíclica, obtemos o mesmo sinal x periodicamente a cada n deslocamentos. Isso significa que o

Figura 3.6: Exemplos de deslocamentos cíclicos verticais de uma amostra de base.

Fonte: Henriques et al. (2015)

conjunto completo de sinais deslocados é obtido com:

$$\{P^u x | u = 0, ...n-1\} \tag{3.7}$$

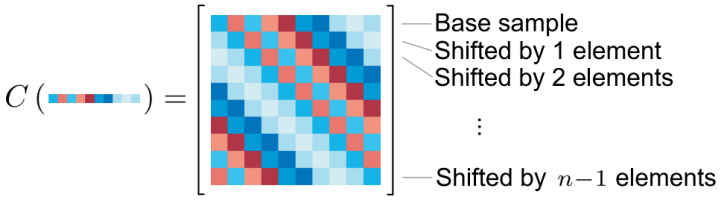
Também devido à propriedade cíclica, podemos ver a primeira metade deste conjunto como deslocamentos na direção positiva, e a segunda metade como deslocamentos na direção negativa.

3.2.2 Matriz Circulante

Para calcular uma regressão com amostras deslocadas, podemos usar o conjunto da Equação 3.7 como as linhas de uma matriz de dados X:

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_n - 1 \\ x_n - 1 & x_n & x_1 & \cdots & x_n - 2 \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}. \tag{3.8}$$

Figura 3.7: Ilustração de uma matriz circulante, onde as linhas são deslocamentos cíclicos de uma imagem vetorizada ou suas translações em 1-D. As mesmas propriedades podem ser aplicadas para matrizes circulantes contendo imagens 2-D



Fonte: Henriques et al. (2015)

Uma ilustração do padrão resultante é dado na Figura 3.7. Podemos observar que o padrão é determinístico e totalmente especificado pelo vetor gerador x, que é a

primeira linha da matriz. O que talvez seja a propriedade mais importante é o fato de que todas as matrizes circulantes são diagonais pela Transformada Discreta de Fourier, independentemente do vetor gerador x.

3.2.3 Pré-processamento

Como visto anteriormente, um problema com o algoritmo de convolução FFT é que a imagem e o filtro são mapeados para a estrutura topológica de um toro. Desta forma, ele conecta a borda esquerda da imagem à borda direita e a parte superior à parte inferior. Durante a convolução, as imagens giram através do espaço toroidal em vez de se transladar como fariam no domínio espacial. Conectar artificialmente os limites da imagem introduz um artefato que afeta a resposta de correlação. Para reduzir esse efeito, a imagem o dado ou imagem de entrada é multiplicado por uma janela do cosseno, que reduz gradualmente os valores dos pixels próximos às bordas para zero. Essa operação também tem o benefício de dar mais ênfase ao centro do objeto alvo (BOLME et al., 2010).

3.2.4 Esquemas de Treinamento

Ao longo dos anos, diversos estudos tem sido propostos para treinar um filtro de correlação *on-line* (CHEN; HONG; TAO, 2015a). Dependendo do tipo de método usado para treinamento podemos obter resultados e comportamento distintos para o filtro de correlação. A seguir são apresentados algumas dessas abordagens.

3.2.4.1 Treinamento Tradicional

No caso mais simples, templates recortados de uma imagem podem ser usados diretamente para produzir picos para um dado alvo em um mapa de correlação. A desvantagem dessa abordagem é que a resposta de correlação para o *background* da cena também será alta. Para resolver essa questão, uma variedade de filtros de correlação foram treinados para suprimir respostas para amostras de treinamento negativas, mantendo uma alta resposta para o alvo. A principal diferença entre esses filtros é o método como eles usam as amostras de treinamento coletadas para serem construídos. O método *Minimum Average Correlation Energy* (MACE) (MAHALANOBIS; Vijaya Kumar; CASA-

SENT, 1987) é treinado com restrições rígidas impostas para que os picos sejam sempre produzidos na mesma altura. Outros métodos consideram que as restrições rígidas sejam desnecessárias, como o *Unconstrained MACE* (UMACE) (MAHALANOBIS et al., 1994), que é treinado relaxando essas restrições.

3.2.4.2 Filtros de Correlação Adaptativos

Bolme et al. (BOLME et al., 2010) desenvolveram um novo filtro, chamado *Minimum Output Sum of Squared Error* (MOSSE), para treinar filtros de correlação de maneira mais eficiente. Um filtro simples pode ser obtido com a amostra x e a correspondente saída desejada y. No entanto, mais amostras são necessárias para melhorar a robustez dos filtros de correlação. Para mapear adequadamente as amostras de entrada para as saídas desejadas, o filtro MOSSE encontra um filtro h minimizando a soma do erro quadrático entre as saídas de correlação obtidas e as saídas de correlação desejadas. Esta minimização pode ser computada no domínio da frequência como:

$$\min_{\hat{h^*}} \sum_i \left\| (\hat{x_i} \otimes \hat{h^*} - \hat{y_i} \right\|^2, \tag{3.9}$$

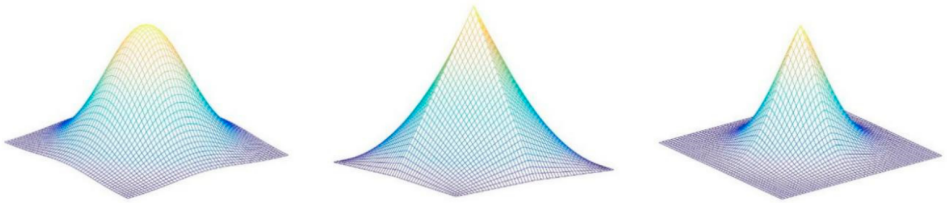
onde i indexa cada imagem de treino. Assim, a solução de $\hat{h*}$ é dada por:

$$\hat{h*} = \frac{\sum_i \hat{y_i} \odot \hat{x_i^*}}{\sum_i \hat{x_i} \odot \hat{x_i^*}} \tag{3.10}$$

A derivação detalhada dessa solução pode ser encontrada no trabalho de Bolme et al. (2010).

A resposta de saída desejada y pode assumir qualquer forma, mas no filtro MOSSE e em geral ela é gerada a partir do *groundtruth* com uma distribuição em forma de Gaussiana 2-D cujo pico está no centro. Se uma função delta de Kronecker for usada para definir y, cujo valor no centro do alvo é um (amostras positivas) e os valores em outros lugares são zero (amostras negativas), o filtro resultante é teoricamente um filtro UMACE mencionado anteriormente. Portanto, UMACE é um caso especial do filtro MOSSE. A Figura 3.8 ilustra alguns exemplos de distribuições que podem ser utilizadas para gerar a resposta de saída desejada y.

Figura 3.8: Exemplos de diferentes distribuições usadas para gerar a resposta de correlação desejada. (a) distribuição Gaussiana 2-D; (b) distribuição triangular 2-D; (c) distribuição 2-D combinada de Gaussiana e triangular.



(a) (b) (c)

Fonte: Yang et al. (2019)

3.2.4.3 Kernelized Correlation Filters

Os métodos de rastreamento baseados em filtros de correlação se mostraram eficientes e robustos para o rastreamento de objetos. Entretanto, a performance de métodos como o MOSSE pode apresentar limitações, uma vez que podem ser vistos como simples classificadores lineares. Tirando vantagem do *kernel trick*, os filtros de correlação podem ser mais eficazes (CHEN; HONG; TAO, 2015a).

Os métodos de kernel devem seu nome ao uso de funções de kernel, que os permitem operar em um espaço de feições implícito de alta dimensão sem nunca computar as coordenadas dos dados nesse espaço, mas simplesmente computar os produtos internos entre os conjuntos imagens de todos os pares de dados no espaço de feições. Essa operação costuma ser computacionalmente mais eficiente do que o cálculo explícito das coordenadas. Essa abordagem é chamada de *kernel trick*, também conhecido como substituição de kernel. A ideia geral é que, se tivermos um algoritmo formulado de forma que o vetor de entrada entre apenas na forma de produtos escalares, esse produto escalar pode ser substituído por alguma outra opção de kernel (BISHOP, 2006).

Henriques et al. (HENRIQUES et al., 2012; HENRIQUES et al., 2015) propôs que os filtros de correlação podem ser efetivamente "kernelizado"tratando o problema de regressão linear como sendo uma regressão *Ridge Regression* e utilizando matriz circulante.

- *Ridge Regression*

  Considerando os filtros de correlação como classificadores, eles podem ser treinados para encontrar a relação entre a i-ésima entrada x_i e seu rótulo y_i a partir de um conjunto de treinamento. Assumindo que a relação tenha a forma $f(x_i) = y_i$, o

problema do treinamento pode ser visto como a minimização da seguinte função:

$$\min_{w} \sum_{i} (f(x_i) - y_i)^2 + \lambda \left\| w \right\|^2 , \tag{3.11}$$

onde w denota os parâmetros e λ é um parâmetro de regularização para evitar *over-fitting*. Resolvendo a Equação 3.11, o parâmetro w pode ser dado por uma forma fechada (RIFKIN; YEO; POGGIO, 2003):

$$w = (X^T X + \lambda I)^{-1} X^T y, \tag{3.12}$$

onde X é uma matriz onde cada coluna é uma amostra de treinamento, y é o vetor com os rótulos correspondentes e I é a matriz identidade. Caso o cálculo seja realizado no domínio de Fourier, X^T deve ser substituído pelo Hermitiano transposto de X:

$$w = (X^H X + \lambda I)^{-1} X^H y, \tag{3.13}$$

onde X^H é o Hermitiano transposto, i.e., $X^H = (X^*)^T$, e X^* é o complexo conjugado de X.

Para introduzir as funções do kernel para melhorar o desempenho, os dados de entrada x podem ser mapeados para um espaço de feições não linear com $\varphi(x)$, e w pode ser expresso por uma combinação linear das entradas $w = \sum_i \alpha_i \varphi(x_i)$. Assim $f(x_i)$ assume a forma:

$$f(x_i) = \sum_{j=1}^{n} \alpha_i k(x_i, x_j), \tag{3.14}$$

onde $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$ é a função kernel. Supondo que K seja a matriz kernel com seus elementos $K_{ij} = k(x_i, x_j)$, a solução para a Equação 3.11 usando funções de kernel pode ser dada por (RIFKIN; YEO; POGGIO, 2003):

$$\alpha = (K + \lambda I)^{-1}, \tag{3.15}$$

onde I é a matriz identidade e α é o vetor de coeficientes α_i, sendo α a variável a ser otimizada em vez de w. Nesta representação alternativa α é dito estar no espaço dual. Para evitar a dificuldade de computar a matriz inversa, o conceito de matriz circulante pode ser utilizado.

Como visto na Subseção 3.2.2, o padrão resultante de uma matriz circulante é é determinístico e totalmente especificado pelo vetor gerador x, que é a primeira linha da matriz, ilustrado na Figura 3.7. Uma das propriedade mais importantes é o fato de que todas as matrizes circulantes são diagonais pela Transformada Discreta de Fourier, independentemente do vetor gerador x. Isso pode ser descrito como:

$$X = Fdiag(\hat{x})F^H, \tag{3.16}$$

onde F é a matriz DFT, que é usada para computar a DFT de um vetor $\mathscr{F}(z) = \sqrt{n}Fz$. Desta forma, a solução de w pode ser descrita como:

$$w = Fdiag\left(\frac{\hat{x}}{\hat{x^*} \odot \hat{x} + \lambda}\right) F^H y, \tag{3.17}$$

que é equivalente a uma forma mais simples no domínio de Fourier:

$$\hat{w} = \frac{\hat{x^*} \odot \hat{y}}{\hat{x^*} \odot \hat{x} + \lambda}, \tag{3.18}$$

onde a divisão é realizada elemento a elemento. Da mesma forma, α também pode ser calculado de forma eficiente se a matriz de kernel K for circulante:

$$\alpha = F\left(diag(\hat{k} + \lambda)\right)^{-1} F^H y \tag{3.19}$$

onde k é o vetor base da matriz circulante K, e ainda:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k} + \lambda}, \tag{3.20}$$

onde a divisão é elemento a elemento.

Foi provado que a função kernel de uma matriz kernel circulante deve ser unitariamente invariante, i.e., não deve sofrer alteraçõe por transformações unitárias (CHEN; HONG; TAO, 2015a). Como produto escalar e funções de kernel de base radial satisfazem esta condição, os kernels polinomiais e os kernels gaussianos são normalmente aplicados.

Se o kernel k é computado entre x e x', um kernel polinomial $k^{xx'} = (x^T x' + a)^b$ pode ser descrito como:

$$k^{xx'} = (\mathscr{F}^{-1}(\hat{x^*} \odot \hat{x'}) + a)^b \tag{3.21}$$

e um kernel Gaussiano $k^{xx'} = exp\left(-\frac{1}{\sigma^2}(\|x - x'\|^2)\right)$ pode ser computado como:

$$k^{xx'} = exp\left(-\frac{1}{\sigma^2}(\|x\|^2 + \|x'\|^2 - 2\mathscr{F}^{-1}(\hat{x^*} \odot \hat{x'}))\right) \tag{3.22}$$

- Detecção

  Em um novo quadro, o alvo pode ser detectado pelo parâmetro treinado α e uma amostra de base x mantida. Considerando um nova amostra z, um mapa de resposta y pode ser obtido através da seguinte equação:

$$y = C(k^{xz})\alpha = \mathscr{F}^{-1}\left(\hat{k^{xz}} \odot \hat{\alpha}\right) \tag{3.23}$$

  De forma similar a outro métodos baseados em filtros de correlação, a posição com o valor máximo em y é predita como a nova posição do alvo rastreado.

3.3 Fatores Prejudiciais ao Rastreamento de Objetos

Nesta seção, serão apresentados brevemente os principais problemas e desafios que envolvem o rastreamento de objetos em cenários realistas, incluindo a localização de objetos que sofrem oclusão, variações de escala e a presença de sombras na cena.

3.3.1 Oclusão

De acordo com Yilmaz, Javed and Shah (2006), a oclusão pode ser classificada em três categorias: auto-oclusão, oclusão inter-objeto e oclusão pela estrutura do plano de fundo da cena. A auto-oclusão ocorre quando uma parte do objeto obstrui a outra. Esta situação surge com mais frequência durante o rastreamento de objetos articulados ou com partes móveis. A oclusão inter-objeto ocorre quando dois objetos sendo rastreados se obstruem. De forma similar, a oclusão pelo plano de fundo ocorre quando alguma estrutura no fundo da cena oclui os objetos rastreados. Geralmente, para oclusão inter-objeto, os métodos de rastreamento podem explorar o conhecimento da posição e da aparência dos objetos para detectar e resolver a oclusão. A oclusão parcial de um objeto por uma estrutura de cena é difícil de detectar, pois é difícil diferenciar entre o objeto mudando sua forma e o objeto sendo obstruído.

Quando ocorre oclusão, a similaridade da região alvo desejada se degrada, fazendo

o método indicar regiões equivocadas como sendo o alvo. Outro problema é a atualização do modelo com as informações erradas (MA et al., 2017). o que traz ainda modelos insatisfatórios para os quadros subsequentes. Assim, a oclusão deve ser dada atenção ao projetar algoritmos de rastreamento precisos e robustos. Para lidar com oclusões, a abordagem usada maioria dos métodos é primeiramente detectar o momento que ocorre a oclusão para, posteriormente, retomar o rastreamento (SCHARCANSKI et al., 2011; MA et al., 2017). Dessa forma, é evitado que regiões erradas sejam incorporadas ao modelo de aparência do objeto e degradem o rastreamento.

3.3.2 Sombras

A presença de sombras é outro fator que pode prejudicar o rastreamento de objetos. Quando os objetos de interesse têm uma forma bem definida, *template matching* ou classificadores mais sofisticados podem ser usados para detectar diretamente os objetos da imagem. Entretanto, com a presença de sombras essa tarefa passa a ter um desafio maior, pois as sombras passam a ser consideradas como parte do objeto, uma vez que compartilham os mesmos padrões de movimento e possuem mudanças de intensidade similar à dos objetos (SANIN; SANDERSON; LOVELL, 2012). Como exemplo de situações em que o desempenho de detecção e rastreamento são afetados pelas sombras, podemos citar objetos que são agrupados por causa de suas sombras projetadas, e a inclusão de pixels de sombra no modelo de aparência do objeto, diminuindo a sua confiabilidade e aumentando o probabilidade de perda do rastreamento.

Existem diversos métodos propostos para tratar o problema das sombras. De maneira geral, os métodos de detecção de sombras podem ser divididos em quatro categorias, que são baseadas no tipo de feições usadas pelos métodos. De acordo com essa classificação, podemos citar os métodos baseados em cromaticidade. Os métodos de detecção de sombra nesta categoria são baseados na noção de que as cromaticidades das regiões de sombra e das regiões do fundo da cena tendem a ser semelhantes e as cores diferindo principalmente em termos de intensidade. Os métodos baseados nessa abordagem geralmente utilizam um espaço de cor que fornece uma melhor separabilidade entre cromaticidade e intensidade do que o espaço de cores RGB. Podemos citar também os métodos baseados em propriedades físicas, que tentam modelar a aparência dos pixels de sombra considerando a iluminação ambiente e outras fontes de luz. Esses métodos costumam estimar os componentes de iluminação e reflexão com base na intensidade de um pixel e de sua

vizinhança (HUANG; CHEN, 2009; GOLCHIN et al., 2013).

Temos ainda os métodos baseados em geometria, onde as sombras são detectadas com base nas características geométricas das regiões de sombra, como sua forma, tamanho e direção do movimento, usando o conhecimento prévio da iluminação e dos objetos na cena. Esses métodos costumam usar informações geométricas, como a localização da câmera, a direção da iluminação da fonte de luz, a superfície do solo e as geometrias do objeto alvo (GOLCHIN et al., 2013) Por fim, temos os métodos baseados em textura. Os métodos nesta categoria são baseados em evidências experimentais de que as texturas dos objetos geralmente são diferentes das texturas do fundo da cena, e as texturas das regiões de sombra e das regiões do fundo tendem a ser semelhantes. Os métodos baseados em textura costumam usar técnicas de correlação para comparar a textura de uma região da imagem com a textura da região correspondente em uma imagem de referência.

A detecção de sombra pode ajudar a melhorar a confiabilidade do rastreamento de objetos, mas a escolha do tipo de método de sombra depende das especificações do sistema (por exemplo, erro tolerado), a disponibilidade de recursos computacionais e informações prévias sobre a cena e sua iluminação em condições normais de operação. Em nosso artigo apresentamos algumas dessas soluções de forma mais detalhada (BARCELLOS; GOMES; SCHARCANSKI, 2016).

3.3.3 Variação de Escala

Estimar a escala de maneira robusta é um problema desafiador no rastreamento de objetos. A maioria dos métodos existentes não consegue lidar bem com grandes variações de escala em sequências de imagens complexas (DANELLJAN et al., 2014a). Ele influencia o desempenho de rastreamento em dois aspectos. Em primeiro lugar, os recursos do alvo exibem uma diferença de vários níveis quando a escala do alvo muda. Ele aciona a imprecisão de rastreamento ao procurar o candidato em um espaço de escala fixa. Em segundo lugar, a escala fixa contribui ainda mais para a imprecisão da atualização do modelo (ou seja, se a caixa delimitadora de rastreamento for maior do que o alvo, as informações de fundo geralmente estão contidas; pelo contrário, se a caixa delimitadora de rastreamento for menor do que o alvo, ela sofre de perda de informações sobre o alvo). Assim, a variação de escala pode degradar a capacidade de representação do modelo. Essa variação influencia o desempenho do rastreamento em dois aspectos. Primeiramente, as feições do alvo exibem diferenças em múltiplos níveis quando a escala do alvo muda, le-

vando a imprecisão do rastreamento ao buscar o candidato com a melhor correspondência em um espaço de escala fixa. Em segundo lugar, a escala fixa contribui ainda mais para a imprecisão da atualização do modelo, pois se a *bounding box* de rastreamento for maior do que o alvo, as informações de fundo geralmente serão incorporadas à representação desse objeto; caso ocorra o contrário, e a *bounding box* de rastreamento for menor do que o alvo, teremos uma perda de informação sobre o alvo (MA et al., 2017).

Uma abordagem bastante utilizada é a utilização de uma representação em pirâmide de escala (LIANG et al., 2019; HU et al., 2017; LI; ZHU, 2014; DANELLJAN et al., 2014a), onde o objeto rastreado é comparado com modelos aprendidos em diferentes escalas, sendo a escala do modelo mais similar ao objeto utilizada como o valor de escala estimado para aquele objeto. Outra abordagem é o uso de rastreamento baseado em partes do objeto, onde cada parte do alvo é rastreada de forma independente, o que permite estimar a variação de tamanho do objeto ao longo do tempo (BARCELLOS; SCHARCANSKI, 2020; AKIN et al., 2016). Outra abordagem comum é o uso do sistema de coordenadas Log-Polar, que é largamente usada em processamento de sinais para estimar variações de escala e rotação. Métodos baseados nessa abordagem geralmente convertem as imagens para coordenadas polares logarítmicas, dessa forma, as mudanças de escala e rotação se transformam em deslocamentos, que podem ser estimados simplesmente computando a translação do objeto nesse sistema de coordenadas (LI et al., 2017)

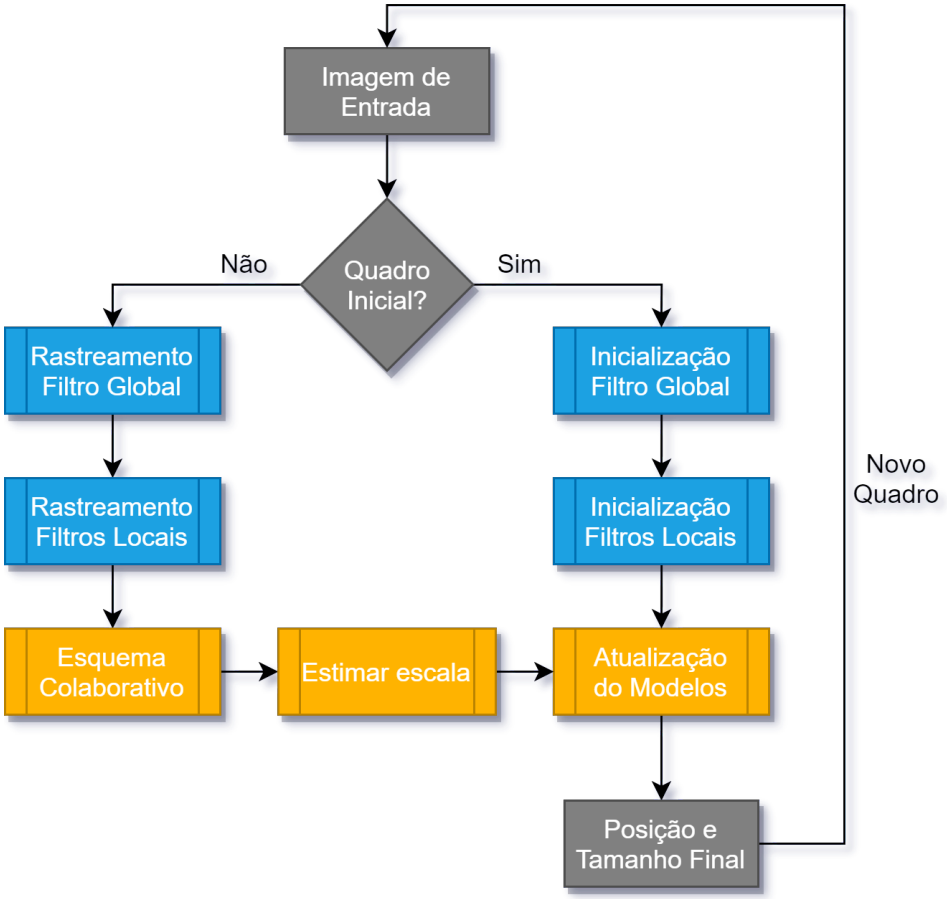
4 MÉTODO PROPOSTO

Neste trabalho, está sendo proposto um *framework* de rastreamento baseado em partes utilizando filtros de correlação. O método proposto combina múltiplos filtros de correlação baseados em partes locais de forma conjunta com um filtro de correlação global para melhorar o rastreamento de objetos. Os filtros locais rastreiam partes individuais do objeto e suas posições são combinadas para determinar o centro do objeto rastreado. O uso de filtros locais introduz robustez e flexibilidade ao processo de rastreamento, especialmente em situações em que o objeto sofre de oclusões parciais, deformações ou mudanças de aparência, uma vez que algumas partes individuais permanecem visíveis e preservam a aparência original. O filtro global é aprendido usando a região inteira do objeto e é usado para estimar a posição do objeto nos casos em que os filtros locais não conseguem fazer o rastreamento de uma parte local de forma confiável. Ainda, as regiões de busca para os filtros de correlação locais serão determinadas com base na posição indicada pelo filtro global como sendo a região do objeto.

A Figura 4.1 exibe um diagrama com as principais etapas do método proposto. Primeiramente, caso a imagem de entrada seja o primeiro quadro do vídeo, e realizado a inicialização do filtro global, onde feições serão extraídas da imagem para criar o modelo inicial do objeto. Após, com base na posição do filtro global, é realizado o mesmo processo de inicialização para a criação dos modelos de cada filtro local. Nos quadros subsequentes, primeiramente é realizado o rastreamento utilizando o filtro global para estimar a posição do objeto. Em seguida, é realizado o rastreamento de cada uma das partes rastreadas pelos filtros locais, baseado na posição estimada pelo filtro global, que irá determinar qual a área de busca para cada um dos filtros locais. Assim, o esquema colaborativo proposto (ver Seção 4.5) é utilizado para estimar a posição final do objeto rastreado. Após definir a posição do objeto, é estimado o valor para a escala atual, comparando a variação das posições dos filtros locais em relação aos quadros anteriores. Por fim, é realizado a atualização dos modelos (ver Seção 4.4, de acordo com as novas posições estimadas, e é retornado a posição e o tamanho estimado do objeto.

A seguir, apresentamos as feições convolucionais utilizadas no método proposto, os detalhes dos filtros de correlação global e locais e o esquema colaborativo proposto.

Figura 4.1: Ilustração método proposto.

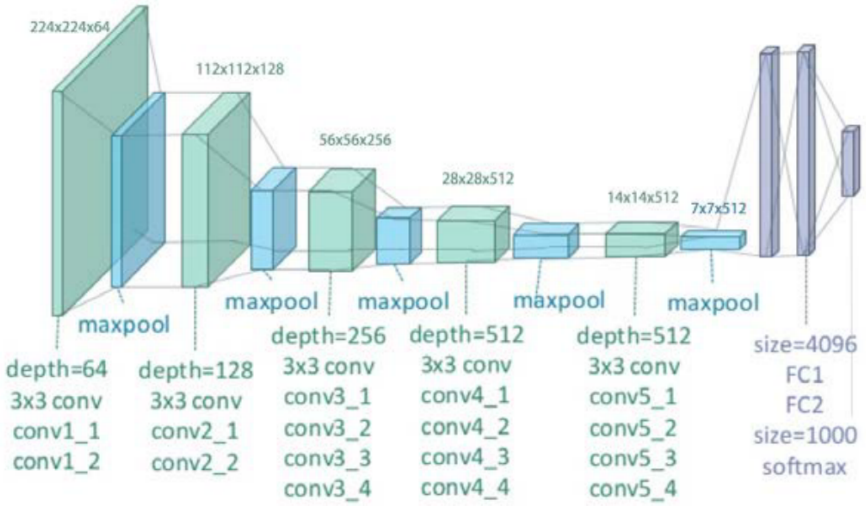


4.1 Extração de Feições

O uso de Redes Neurais Convolucionais profundas para detecção de objetos tem avançado significativamente nos últimos anos com o aumento do poder computacional e a disponibilidade de grandes bases de dados de imagens para treinamento, e o uso feições extraídas usando CNNs demonstrou ser extremamente robusto para tarefas de reconhecimento visual (DANELLJAN et al., 2015a). Geralmente, as feições são extraídas das camadas de redes pré-treinadas, por exemplo, AlexNet (KRIZHEVSKY; HINTON, 2012) e VGG-NET (SIMONYAN; ZISSERMAN, 2014), que são treinadas utilizando a base de dados de larga escala ImageNet (Jia Deng et al., 2009). Neste trabalho utilizamos a rede VGG-19, que possui 19 camadas (16 camadas convolucionais, 3 camadas totalmente conectadas) que usa estritamente filtros 3×3, junto com camadas de *max-pooling* de 2×2. A Figura 4.2 exibe uma ilustração da arquitetura da CNN VGG-19.

A VGG-19 é treinada em mais de um milhão de imagens e pode classificar imagens em 1000 categorias de objetos. Como resultado, o modelo aprendeu representações

Figura 4.2: Ilustração arquitetura VGG-19.



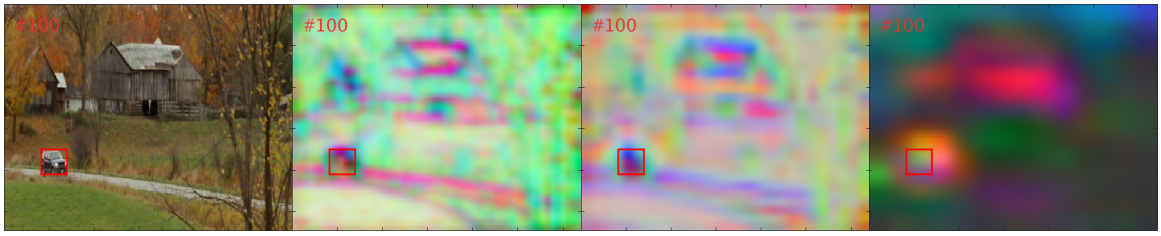
Fonte: (ZHENG; YANG; MERKULOV, 2018)

de feições bastantes ricas para uma ampla gama de imagens. Como a rede é treinada para classificação de imagens, as 3 ultimas camadas dessa rede atuam como classificador. Caso objetivo seja utilizar apenas as feições aprendidas, como neste trabalho, após a rede aprender os pesos do modelo as camadas finais podem ser removidas e as primeiras camadas podem ser usadas como um extrator de feições.

Como visto na Subseção 3.1.7, uma CNN típica recebe uma imagem RGB de tamanho fixo como entrada e realiza sequências de convoluções, normalizações locais e operações de *pooling* para cada camada, e as saídas das camadas convolucionais são usadas como feições (*deep features*). Assim, os mapas de feições gerados para cada camada apresentam diferentes feições, enfatizando diferentes características da imagem. Ma et al. (MA et al., 2015) observaram que as últimas camadas convolucionais fornecem mais informações semânticas do alvo, e também são robustas às variações de aparência, enquanto as camadas iniciais fornecem detalhes espaciais mais finos e são mais precisas para a localização do alvo. Assim, eles propõem o uso de múltiplos filtros de correlação de acordo com as camadas convolucionais hierárquicas. Portanto, optamos por usar essa abordagem em nosso trabalho para explorar as vantagens de feições que podem representar detalhes de mais alto nível e a semântica dos objetos, bem como as vantagens das feições de níveis mais baixos que representam detalhes mais finos e precisos. A Figura 4.3 exibe exemplos de feições extraídas de diferentes camadas convolucionais da rede VGG-19 (SIMONYAN; ZISSERMAN, 2014).

À medida que a profundidade das camadas convolucionais aumenta, a resolução espacial do mapa de feições é reduzida devido às operações de *pooling*. Por esse motivo,

Figura 4.3: Exemplos de feições extraídas de diferentes camadas convolucionais.
(a) Entrada (b) *conv3-4* (c) *conv4-4* (d) *conv5-4*



Neste exemplo, temos a imagem de entrada (a) e as saídas de diferentes camadas convolucionais da rede VGG-NET. As figuras (b)-(d) exibem as saídas das camadas convolucionais *conv3-4*, *conv4-4* e *conv5-4*, respectivamente.

as camadas totalmente conectadas não são usadas para extração de feições, já que a resolução espacial é de somente 1×1. Como as baixas resoluções espaciais das últimas camadas são insuficientes para localizar os alvos corretamente, devido às operações de *pooling*, uma interpolação bilinear é aplicada para redimensionar cada mapa de feições para um tamanho fixo e mitigar o problema (MA et al., 2015).

4.2 Filtro de Correlação Global

Filtro de Correlação é uma técnica utilizada para aprender um classificador discriminativo capaz de estimar o deslocamento de um objeto entre quadros consecutivos de uma sequência de vídeo. Amostras de treinamento são geradas em torno do alvo de maneira eficiente através do uso de propriedades da correlação circular, e as feições são extraídas dessas amostras para realizar o treinamento do filtro de correlação. Nos quadros subsequentes, é realizado a correlação do filtro aprendido com as diversas posições dentro de uma janela de busca, e o valor máximo de correlação no mapa de resposta gerado indica a localização do alvo. As amostras para o objeto alvo e para o fundo da cena são geradas usando deslocamentos cíclicos da imagem de entrada (*patch*), que é formado por uma área aumentada ao redor do objeto alvo. A Figura 4.4 mostra um exemplo de deslocamentos cíclicos para um *patch* de entrada, conforme apresentado na Subseção 3.2.1. Dado o *patch* de entrada $x = [x_1, x_2, \ldots, x_n]$, é obtido uma matriz circulante C de tamanho $n \times n$ concatenando todos os possíveis deslocamentos cíclicos de x (HENRIQUES

et al., 2012):

$$C = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_n - 1 \\ x_n - 1 & x_n & x_1 & \cdots & x_n - 2 \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}. \tag{4.1}$$

A matriz C denota a matriz com as amostras de treinamento, onde cada linha é uma amostra, conforme descrição mais detalhada presente na Seção 3.2. Além disso, C se torna diagonal no domínio de Fourier:

$$C = D\, diag(X)\, D^H \tag{4.2}$$

onde D é a matriz resultante da Transformada Discreta de Fourier (DFT), X é a DFT de x, $diag(X)$ é a matriz diagonal com os elementos não zeros de X e H denota o conjugado complexo transposto. Assim, a matriz de amostras de treinamento C pode ser eficientemente representada no domínio da frequência (SUI; WANG; ZHANG, 2017).

Figura 4.4: Exemplo de deslocamento cíclicos.
(a) Imagem original

(b) Deslocamentos cíclicos da imagem original em +20 e -20 pixels em ambas as direções

No método proposto, as saídas de cada camada convolucional são usadas como feições multicanais (MA et al., 2015). Portanto, x é o vetor de feições da camada l e tem tamanho $M \times N \times D$, onde M, N e D correspondem à largura, altura e ao número de

canais, respectivamente. Cada amostra $x_{m,n}, (m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$, possui uma função $2D$ responsável por gerar os rótulos das amostras seguindo uma distribuição Gaussiana da forma $y(m,n) = \exp \frac{-(m-M/2)^2+(n-N/2)^2}{2\sigma^2}$, onde σ é a largura do kernel. A vantagem de gerar os rótulos usando uma função gaussiana é evitar o uso de um limiar rígido. Assim, o valor do rótulo é de 1 próximo ao centro do alvo e decai para 0 à medida que a distância em relação ao centro aumenta. Um filtro de correlação w com o mesmo tamanho de x é aprendido minimizando o seguinte problema de regressão linear:

$$w* = \underset{w}{\operatorname{argmin}} \sum_{m,n} \|(w \otimes x_{m,n} - y_{m,n})\|^2 + \lambda \|w\|_2^2 \tag{4.3}$$

onde λ é um parâmetro de regularização ($\lambda \geq 0$), e $w \otimes x_{m,n}$ é a resposta de convolução do filtro desejado em uma amostra x de tamanho $M \times N$, dada por $\sum_{d=1}^{D} w_{m,n,d}^T x_{m,n,d}$. Esse problema de minimização pode ser resolvido eficientemente para cada canal de feições usando a Transformada de Fourier.

$$W^d = \frac{Y \odot \bar{X}^d}{\sum_{i=1}^{D} X^i \odot \bar{X}^i + \lambda} \tag{4.4}$$

onde W^d é o filtro aprendido no domínio de Fourier para o canal d ($d \in 1,\ldots,D$), $\odot$ denota o produto de Hadamard (multiplicação elemento a elemento), X e Y são as transformadas de Fourier de x e y, respectivamente, e barra significa o complexo conjugado.

Após o filtro de correlação ser aprendido, as mesmas propriedades são usadas nos quadros subsequentes para detectar o alvo. Um *patch* da imagem de entrada é obtido na posição anterior do alvo, e feições são extraídas usando a CNN para criar um vetor de feições z de tamanho $M \times N \times D$. O mapa de resposta do vetor de feições z na camada l, f^l, é computado como

$$f^l = \mathscr{F}^{-1}(\sum_{d=1}^{D} W^d \odot Z^d), \tag{4.5}$$

onde $\mathscr{F}^{-1}$ denota a transformada inversa de Fourier e Z é o vetor de feições z no domínio de Fourier.

Por fim, o local do alvo no mapa de correlação f^l, de tamanho $M \times N$, para a camada l é estimado localizando a posição que apresenta o valor de resposta máximo no mapa de correlação. Depois de gerar os mapas de correlação para cada uma das camadas, a correlação máxima pode estar em locais diferentes em cada camada l, pois as feições extraídas de diferentes camadas da CNN podem produzir diferentes mapas de res-

posta. Portanto, é essencial combinar esses mapas de resposta para garantir que o local do alvo seja estimado corretamente. Essa abordagem permite explorar os múltiplos níveis de abstração de cada camada de feições, ou seja, as feições nas últimas camadas convolucionais, que apresentam mais informações semânticas, e as feições nas camadas iniciais, que apresentam mais detalhes espaciais do objeto, são combinadas para gerar um estimativa melhor da posição do objeto rastreado. Uma soma ponderada dos mapas de resposta das diferentes camadas é computada para estimar a localização do alvo da seguinte maneira:

$$\arg\max_{m,n} = \sum_{l} \mu_l f^l(m,n) \tag{4.6}$$

onde μ_l é um parâmetro de peso, indicando a contribuição de cada camada para o resultado final.

A localização estimada do alvo é a posição (m, n) que maximiza essa soma dos mapas de respostas ponderados para as múltiplas camadas. A Figura 4.5 ilustra diferentes mapas de correlação gerados a partir de três diferentes camadas de uma CNN (Figura 4.5a, Figura 4.5b e Figura 4.5c). Na Figura 4.5d temos o mapa de resposta final utilizado para determinar a posição estimada do objeto rastreado, que é computado através da combinação dos diversos mapas de respostas, cada um com um peso diferente para indicar a contribuição de cada camada para o resultado final.

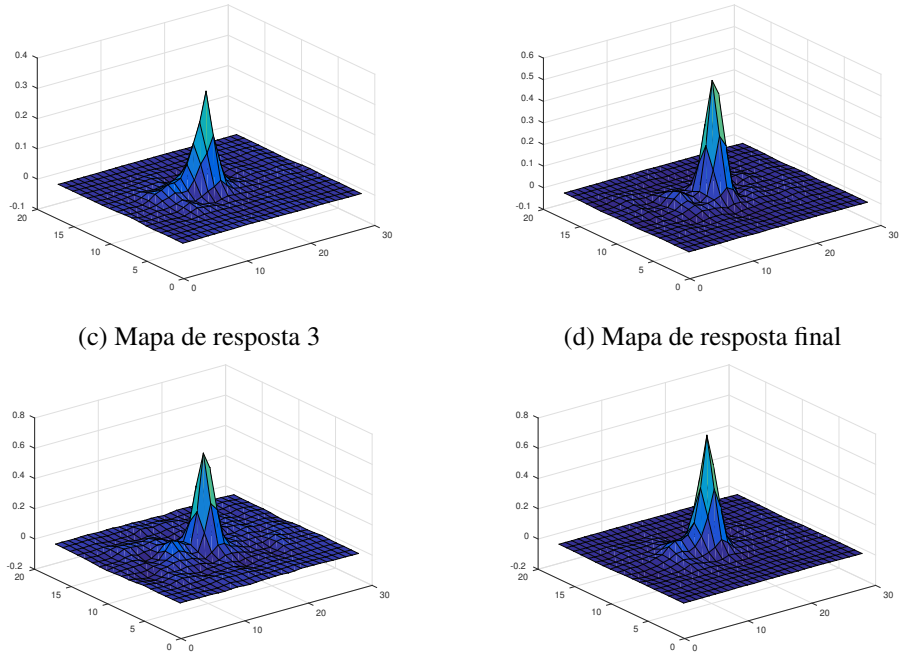
Neste trabalho utilizamos as camadas convolucionais *conv4-4*, *conv3-4* e *conv5-4*, e os parâmetros de peso 1, 0.5 e 0.25, respectivamente, conforme sugerido no estudo de Ma et al. (MA et al., 2015). Desta forma, a camada *conv4-4* apresenta um equilíbrio entre informação semântica e detalhes finos da imagem, e por isso utiliza um peso maior. Das três camadas utilizadas, a camada *conv3-4* é a que apresenta mais detalhes finos da cena, o que ajuda a estimar com mais precisão a posição do objeto. e por isso foi utilizado o segundo maior peso. Já a camada *conv5-4* é a que consegue discriminar o alvo mesmo com grandes variações no fundo da cena, mas não é tão precisa em extrair os detalhes finos da imagem, assim é necessário evitar atribuir um peso muito alto para essa camada, pois isso poderia levar a uma estimativa incorreta da posição do alvo.

4.3 Filtros de Correlação Locais

Inspirados pela eficiência e robustez dos métodos de filtros de correlação baseados em partes (LI; ZHU; HOI, 2015; LIU; WANG; YANG, 2015; AKIN et al., 2016), é

Figura 4.5: Exemplos de mapas de repostas gerados para diferentes camadas da CNN.

(a) Mapa de resposta 1 (b) Mapa de resposta 2



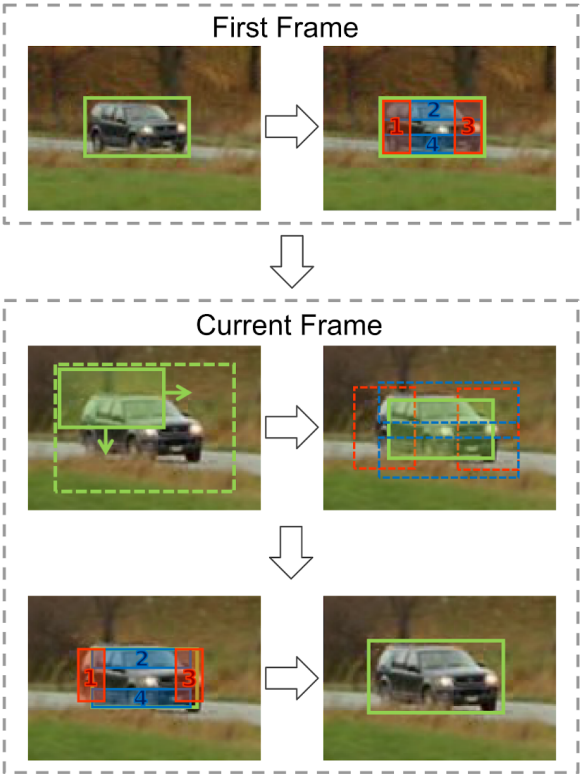
(c) Mapa de resposta 3 (d) Mapa de resposta final

proposto um *framework* que utiliza filtros de correlação locais em associação com um filtro de correlação global. Os filtros de correlação locais propostos funcionam da mesma maneira que o filtro global, mas em vez de usar a região inteira do objeto, apenas partes individuais do objeto são usadas.

A Figura 4.6 mostra uma visão geral do esquema proposto. No primeiro quadro, a *bounding box* do objeto alvo é usada para inicializar o modelo do filtro global. Em seguida, com base na localização do filtro global, os filtros de correlação baseados em partes são inicializados, gerando filtros de correlação locais para cada uma das partes. Nos quadros subsequentes, o alvo é rastreado correlacionando o filtro global em uma janela de busca e a localização estimada pelo filtro global é então usada para fornecer as posições iniciais das janelas de busca para os filtros locais. Depois que a posição de cada parte é determinada, elas são combinadas para fornecer a posição final e a forma do objeto rastreado.

A região do objeto rastreado é dividida em quatro partes e cada parte é rastreada de forma independente. A Figura 4.7 mostra um exemplo de um objeto e a região que será rastreada pelo filtro global (Figura 4.7a) e as quatro regiões que serão rastreadas por filtros de correlação locais. A região do objeto é dividida em duas partes horizontais, como mostrado na Figura 4.7b, e em duas partes verticais, como mostrado na Figura 4.7c. A Figura 4.7d ilustra as quatro regiões que serão rastreadas por cada um dos filtros locais.

Figura 4.6: Visão geral do método de rastreamento por partes proposto.



Em vez de usar cada uma das metades do objeto como as regiões a serem rastreadas, optamos por usar apenas as regiões próximas às extremidades do objeto. Assim, podemos obter um desempenho melhor em termos de processamento, uma vez que a área a ser processada será menor. Além disso, usar apenas as extremidades pode colaborar para que o método consiga discriminar melhor entre o objeto e o plano de fundo da cena, pois os filtros locais são centralizados nas regiões do objeto que possuem mais detalhes, que geralmente estão próximos aos extremos e contornos do objeto.

Dada uma *bounding box* BB centrada em (m_c, n_c) e tamanho $h_{bb} \times w_{bb}$ contendo o objeto a ser rastreado, os dois filtros de correlação horizontais são centralizados em Δ^H pixels a partir do centro do objeto em direção às bordas esquerda e direita, respectivamente, definindo regiões de tamanho $h_{bb} \times 2S$. Da mesma forma, os dois filtros de correlação verticais são centralizados em Δ^V pixels a partir do centro do objeto em direção às bordas superior e inferior, respectivamente, definindo regiões de tamanho $2S \times w_{bb}$. Formalmente, esses deslocamentos nas direções horizontal (Δ^H) e vertical (Δ^V) são definidos como:

$$\begin{aligned} \Delta^H &= (w_{bb}/2) - S \\ \Delta^V &= (h_{bb}/2) - S. \end{aligned} \tag{4.7}$$

Consequentemente, cada uma das partes horizontais, $L_1(m,n)$ e $L_2(m,n)$, e verticais, $L_3(m,n)$ e $L_4(m,n)$ são centralizadas nas posições definidas a seguir:

$$\begin{aligned} L_1(m_{h1}, n_{h1}) &= (m_c - \Delta^H, n_c) \\ L_2(m_{v2}, n_{v2}) &= (m_c, n_c - \Delta^V) \\ L_3(m_{h3}, n_{h3}) &= (m_c + \Delta^H, n_c) \\ L_4(m_{v4}, n_{v4}) &= (m_c, n_c + \Delta^V), \end{aligned} \tag{4.8}$$

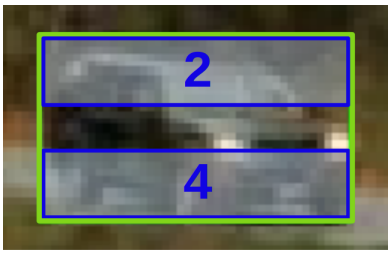
onde L_p ($p \in \{1,2,3,4\}$) representa a localização de cada uma das partes p. Amostras de treinamento para cada uma das partes a serem rastreadas são geradas usando deslocamentos cíclicos de um *patch* centrado em cada uma das posições L_p, de forma similar ao realizado para o filtro global. A Figura 4.8 ilustra as posições das regiões locais do objeto que serão rastreadas por cada um dos filtros locais.

Figura 4.7: Regiões rastreadas por cada um dos filtros.

(a) Região global

(b) Regiões horizontais

(c) Regiões verticais



(d) Regiões locais do alvo

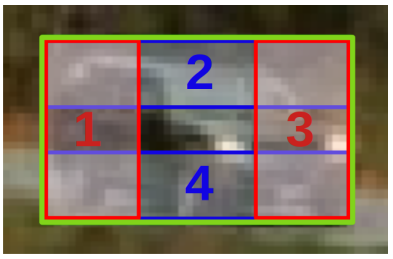
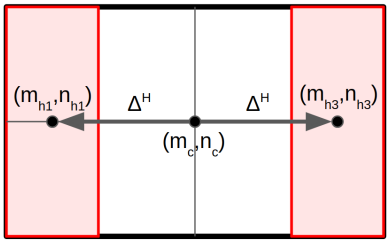
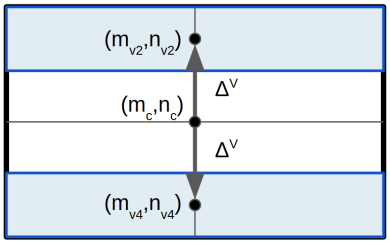


Figura 4.8: Posições dos filtros locais em relação ao centro do objeto.

(a) Posições dos filtros horizontais



(b) Posições dos filtros verticais



Modificando o tamanho de cada uma das partes irá influenciar no desempenho do método. Se aumentarmos muito a área utilizada teremos, primeiramente, um impacto no uso de recursos computacionais, uma vez que será processado uma região maior da imagem. Ainda, podemos perder desempenho em termos de rastreamento, pois quanto maior for a área utilizada, menor será a robustez à mudanças na aparência. Assim, caso a região escolhida seja muito grande, uma mudança em apenas parte dessa região irá afetar o rastreamento da parte inteira, podendo levar à perda do rastreamento do objeto, o que não aconteceria caso, por exemplo, a região que sofreu uma mudança na aparência estivesse de fora das partes escolhidas para serem rastreadas.

Nos casos em que a região escolhida for menor, a aparência do objeto fora da partes rastreadas pode variar bastante sem que isso afete o rastreamento, desde que as partes escolhidas não sofram modificações profundas de aparência. Vale notar que uma região muito pequena também afetará negativamente o método, pois as partes não terão informações o suficiente para fazer o rastreamento de maneira adequada. Por isso, é necessário avaliar cada situação para definir o melhor tamanho para essas regiões.

Para cada parte p é aprendido um filtro de correlação no domínio de Fourier da mesma forma que no filtro global, usando o seguinte forma fechada:

$$W_p^d = \frac{Y_p \odot \bar{X}_p^d}{\sum_{i=1}^{D} X_p^i \odot \bar{X}_p^i + \lambda} \tag{4.9}$$

onde W_p^d é o filtro aprendido para a parte p no canal de feição d ($d \in 1, \ldots, D$), $\odot$ denota o produto de Hadamard, X_p e Y_p são as transformadas de Fourier do vetor de feição (x_p) e do vetor de rótulos Gaussiano (y_p) para a parte p, respectivamente, e a barra significa complexo conjugado.

Após aprender os filtros de correlação para todas as partes do objeto, a seguinte equação é usada para estimar o mapa de resposta f_p^l para a parte p na camada l:

$$f_p^l = \mathscr{F}^{-1}(\sum_{d=1}^{D} W_p^d \odot Z_p^d), \tag{4.10}$$

onde Z_p de tamanho $M \times N \times D$ é o vetor de feições no domínio de Fourier da região de busca obtida na posição anterior da parte p.

Como no filtro de correlação global, a localização final L_p de cada parte p é estimada por uma média ponderada dos mapas de respostas das diferentes camadas, definido

como:

$$\arg\max_{m,n} = \sum_{l} \mu_l f_p^l(m,n) \tag{4.11}$$

onde μ_l é o mesmo parâmetro de peso usado para o filtro de correlação global, indicando a contribuição de cada camada para o resultado final. A localização estimada da parte p é a posição (m,n) que maximiza a soma sobre os mapas ponderados para as múltiplas camadas.

4.4 Esquema Proposto para a Atualização dos Filtros de Correlação

A aparência do objeto pode mudar durante o processo de rastreamento devido a mudanças em sua posição, oclusão, variações de iluminação, sombras e ou outras interferências do ambiente. Para lidar melhor com essas mudanças de aparência e minimizar esses efeitos, é empregado um esquema de atualização para atualizar os filtros ao longo do vídeo.

Atualizar os filtros com muita frequência pode introduzir erros no modelo, enquanto atualizar com pouca frequência pode não capturar mudanças bruscas do objeto. Consequentemente, é necessário encontrar um equilíbrio na taxa de atualização. Para resolver esse problema, usamos uma estratégia para verificar se o resultado da correlação de um filtro é confiável, ou seja, determinar se a posição estimada pelos filtros pode ser considerada como correta. É proposto uma abordagem que avalia os valores das respostas de correlação para determinar se o resultado do filtro é confiável. A Figura 4.9 ilustra uma sequência onde o alvo rastreado sofre oclusão. Nesta situação não queremos atualizar o filtro com uma representação incorreta do objeto rastreado. Se observarmos a resposta de correlação máxima de um filtro de correlação para cada quadro ao longo do vídeo, teremos os resultado ilustrado na Figura 4.10a. A figura mostra que os valores de resposta de correlação permanecem dentro de um intervalo de valores (geralmente alto) e só diminuem nos quadros em que o objeto não é visível devido à oclusão. Assim, a seguinte regra é usada para avaliar a confiabilidade do filtro:

$$R = \frac{\phi(t)}{média(\{\phi(t-1), \phi(t-2), \ldots, \phi(t-\tau)\})}, \tag{4.12}$$

onde τ é o número de quadros anteriores usados para estimar a confiabilidade do filtro e

$\phi(t)$ é o valor do PSR (*Peak to Sidelobe Ratio*) para o mapa de correlação, definido como:

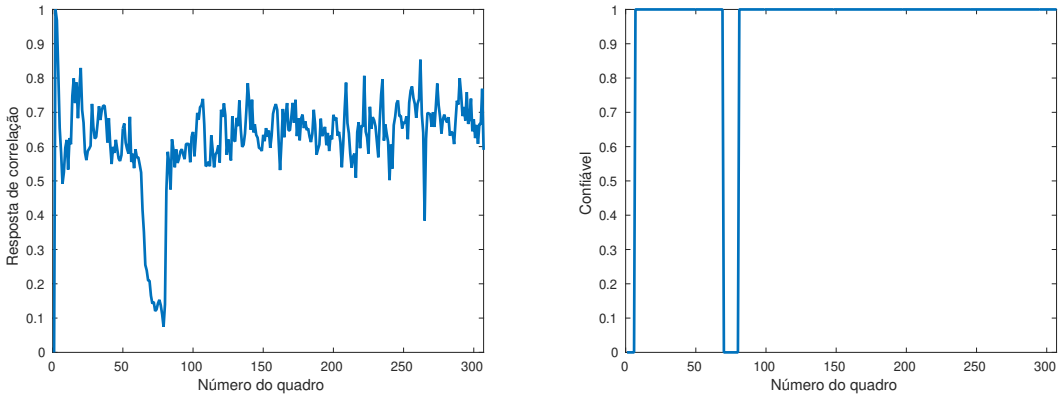
$$\phi(t) = \frac{max_M - med_M}{dp_M}, \tag{4.13}$$

onde max_M, med_M e dp_M são os valores máximo de resposta, a média e desvio padrão do mapa de correlação no quadro t, respectivamente.

Se $R_t > \Psi$, o resultado obtido pelo filtro é considerado como confiável no quadro t. A Figura 4.10b ilustra o resultado obtido usando essa abordagem, com a sequência de quadros onde ocorre a oclusão sinalizados como não confiáveis (valor 0), enquanto o restante foi considerado como confiável (valor 1).

Figura 4.9: Alvo rastreado sofre oclusão durante alguns quadros.

Figura 4.10: Resposta de correlação máxima para cada quadro ao longo do vídeo.



Um filtro ideal para a camada *l* pode ser obtido minimizando o erro de saída em todas as amostras de treinamento (MA et al., 2015). No entanto, isso tem um alto custo computacional, uma vez que é necessário resolver um sistema de equações lineares de $D \times D$ por pixel. Em vez disso, atualizamos o numerador (A) e o denominador (B) do

filtro de correlação W^d na equação 4.4 separadamente como segue:

$$A_t^d = \begin{cases} (1-\gamma)A_{t-1}^d + \gamma(Y \odot \bar{X}_t^d) & \text{if } R_t > \Psi \\ A_{t-1}^d & \text{caso contrário} \end{cases} \tag{4.14a}$$

$$B_t^d = \begin{cases} (1-\gamma)B_{t-1}^d + \gamma(\sum_{i=1}^{D} X_t^i \odot \bar{X}_t^i) & \text{if } R_t > \Psi \\ B_{t-1}^d & \text{caso contrário} \end{cases} \tag{4.14b}$$

$$W_t^d = \frac{A_t^d}{B_t^d + \lambda}, \tag{4.14c}$$

onde γ é corresponde a uma taxa de aprendizado.

Portanto, os filtros são atualizados através de uma interpolação linear entre o novo filtro estimado e o filtro do quadro anterior. A atualização ocorre somente se os valores de correlação de cada filtro forem considerados confiáveis. Quando o valor não é considerado confiável, o filtro não é atualizado, mantendo o filtro do quadro anterior.

Cada parte p dos filtros W_p^d na equação 4.9 é atualizada usando o mesmo esquema, atualizando apenas o filtro local p que é considerado confiável.

4.5 Esquema Colaborativo de Rastreamento

É proposto um esquema colaborativo eficiente que combina o filtro de correlação global e os filtros de correlação locais para aproveitar as vantagens das duas abordagens. Os filtros locais são usados para rastrear as partes individuais do objeto, e cada uma das partes mantém a informação de deslocamento em relação ao centro do objeto. Os dois filtros horizontais e os dois filtros verticais funcionam como pares, sendo os dois filtros horizontais responsáveis pela determinação da posição no eixo horizontal, e os dois verticais para determinar a posição no eixo vertical.

A média entre os dois filtros locais horizontais e os dois verticais são usados para determinar as posições do objeto rastreado da seguinte maneira:

$$L_G(m_g, n_g) = \left(\frac{m_{h1} + m_{h3}}{2}, \frac{n_{v2} + n_{v4}}{2} \right), \tag{4.15}$$

onde m_{hp} e n_{vp} são as posições horizontais e verticais da parte L_p, respectivamente.

Assim, essa posição $L_G(m_g, n_g)$, localizada entre os filtros locais, é usada como o centro estimado do objeto. Além disso, essa posição é usada como a nova posição para o filtro global, e novas feições são extraídas dessa posição e usadas para atualizar o modelo de aparência global de objeto rastreado. É importante notar que os filtros locais serão usado somente quando os resultados de correlação dos filtros forem considerados confiáveis, ou seja, usamos uma estratégia para avaliar se a posição estimada pelos filtros está correta, conforme descrito na Seção 4.4. Caso algum dos pares de filtros horizontais e verticais seja considerado como tendo um valor de correlação não confiável, o filtro global será usado para estimar a posição que será usada como a posição do centro do objeto. Ainda, a variação de distância entre os dois filtros horizontais e entre os dois filtros verticais fornecem uma informação conveniente. Como os filtros locais estão localizados nas extremidades do alvo, as mudanças de escala do alvo podem ser estimadas com base no aumento ou decremento da distância entre essas partes.

No primeiro quadro, a distância entre os filtros esquerdo e direito e os filtros superior e inferior são armazenados para serem usados para estimar as alterações de escala do objeto nos quadros seguintes. O fator de escala inicial ϵ_p é calculado no primeiro quadro da seguinte forma:

$$\epsilon_p = dist(L_p, L_{p+2}), p \in \{1, 2\} \tag{4.16}$$

onde $dist(a, b) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}$ representa a distância Euclidiana.

Para cada quadro t subsequente, o fator de escala é estimado como:

$$\epsilon_{est} = \frac{1}{2}\left(\sum_{p=1}^{2} \frac{dist(L_p, L_{p+2})}{\epsilon_p}\right) \tag{4.17}$$

Para evitar que erros na hora de estimar o fator de escala afetem o rastreamento, uma restrição é aplicada à ϵ_{est} para estimar o valor final do fator de escala (ϵ_t):

$$\epsilon_t = \begin{cases} \epsilon_{max}, & \text{if } \epsilon_{est} > \epsilon_{max}, \\ \epsilon_{min}, & \text{if } \epsilon_{est} < \epsilon_{min}, \\ \epsilon_{est}, & \text{caso contrário.} \end{cases} \tag{4.18}$$

Desta forma, caso o fator de escala seja estimado incorretamente para algum frame, ele ficará dentro de um intervalo de valores toleráveis, evitando que uma área

muito pequena ou muito grande seja estimada de forma incorreta e incorporada ao modelo do objeto. Assim, um fator de escala que foi estimado muito acima ou abaixo do valor real não é propagado para os quadros seguintes, permitindo que o método consiga recuperar o valor correto de escala nos quadros seguintes.

Assim, a posição $L_G(m_g, n_g)$ é utilizada como o centro estimado do objeto rastreado, e o tamanho estimado do objeto é dado pelo tamanho inicial do alvo, indicado pela *bounding box* inicial (BB), multiplicado pelo fator de escala atual (ϵ_t).

Adicionalmente, a posição do filtro global é usada para indicar a região de busca para cada filtro local no quadro subsequente. Assim, as posições iniciais de busca para os filtros locais são determinadas com base na posição do filtro global, e os filtros locais horizontais, $L_1(m_{h1}, n_{h1})$ e $L_2(m_{v2}, n_{v2})$, e verticais, $L_3(m_{h3}, n_{h3})$ e $L_4(m_{v4}, n_{v4})$ são centrados nas posições computadas como a seguir:

$$\begin{aligned} L_1(m_{h1}, n_{h1}) &= (m_g - \epsilon_t \Delta^H, n_g) \\ L_2(m_{v2}, n_{v2}) &= (m_g, n_g - \epsilon_t \Delta^V) \\ L_3(m_{h3}, n_{h3}) &= (m_g + \epsilon_t \Delta^H, n_g) \\ L_4(m_{v4}, n_{v4}) &= (m_g, n_g + \epsilon_t \Delta^V) \end{aligned} \tag{4.19}$$

onde $L_G(m_g, n_g)$ é a posição na qual o filtro global está centralizado. Portanto, o filtro global fornece a posição inicial para os filtros locais e os filtros locais são combinados para fornecer a posição final do alvo.

5 RESULTADOS EXPERIMENTAIS E DISCUSSÃO

Para avaliar o desempenho do método proposto, foram realizados diversos experimentos utilizando *benchmarks* públicos especialmente desenvolvidos para avaliar métodos de rastreamento. Neste capítulo são apresentados os resultados obtidos pelo método proposto, assim como comparações com outros métodos de rastreamento de objetos. Na Seção 5.1 são apresentados os detalhes de implementação, enquanto na Seção 5.2 são apresentados os detalhes das bases de dados utilizadas para os testes e também a metodologia de avaliação. A seguir, na Seção 5.3 são apresentado os resultados experimentais, contendo uma analise quantitativa e qualitativa do método proposto, e também uma comparação com outros métodos. Por fim, na Seção 5.6 é feito uma análise geral sobre os resultados obtidos.

5.1 Detalhes de Implementação

O método proposto foi implementado em Matlab em um PC com uma CPU Intel Core i7-4790@3.60GHz com 32 GB de RAM e uma GPU GeForce Titan Xp.

Para a extração de feições é usado a VGG-NET (SIMONYAN; ZISSERMAN, 2014) treinada na base de dados de grande escala ImageNet (Jia Deng et al., 2009). Descartamos as camadas totalmente conectadas da rede, uma vez que a resolução espacial é de 1×1 apenas, e usamos as saídas das camadas convolucionais *conv3-4*, *conv4-4* e *conv5-4* como feições (*deep features*). Dada uma janela de busca de tamanho $M \times N$, que é centralizada no objeto alvo e maior que o tamanho do alvo (e.g., 1,8 vezes o tamanho do alvo), redimensionamos os canais de feições de cada camada convolucional para $M/4 \times N/4$.

As extrações de feições e atualizações dos filtros seguem o mesmo procedimento durante todos os quadros, e os parâmetros adotados são os mesmos para todos os filtros. Para gerar os rótulos utilizando as funções gaussianas, o parâmetro de largura do kernel σ é definido como 0.1. Seguindo a recomendação em Henriques et al. (HENRIQUES et al., 2015), o parâmetro de regularização λ nas eqs. (4.4) and (4.9) é definido como 10^{-4}. Como usual em métodos que utilizam filtros de correlação, é aplicado uma função de janelamento utilizando uma janela do cosseno em todos os canais de feições extraídos de cada camada convolucional para remover descontinuidades nas bordas da imagem (BOLME et al., 2010; HENRIQUES et al., 2015; GONZALEZ; WOODS, 2002). Como os modelos

de aparência dos filtros de correlação de nosso método proposto são baseados em camadas hierárquicas de uma CNN, empregamos os parâmetros sugeridos em Ma et al. (MA et al., 2015) para o parâmetro de peso μ_l, ou seja, 1, 0.5 e 0.25 para o camadas convolucionais *conv4-4*, *conv3-4* e *conv5-4*, respectivamente. O parâmetro τ, o número de quadros anteriores usados para estimar a confiabilidade do filtro na eq. (4.12), é definido como 5 e o parâmetro de confiabilidade ψ e a taxa de aprendizado γ nas eqs. (4.14a) and (4.14b) são definidos experimentalmente como 0.9 e 0.03, respectivamente.

5.2 Critérios de Avaliação

Avaliamos nosso método proposto utilizando o *benchmark* de rastreamento de objetos OTB-2013 (WU; LIM; YANG, 2013), contendo 50 sequências de vídeo, e o *benchmark* OTB-2015 (WU; LIM; YANG, 2015), composto por 100 sequências de vídeo. Nos *benchmarks*, o *ground truth* para cada quadro das sequências de vídeo é constituído de uma *bounding box* retangular, que é comparada com a saída de nosso método proposto.

As sequências apresentam diferentes situações de desafiadoras, incluindo mudanças de iluminação, deformações não rígidas, variações de escala, e rotações dos objetos. Para facilitar a avaliação de desempenho de cada método de rastreamento, as sequências podem ser classificadas de acordo com 11 atributos diferentes, destacando os diferentes desafios presentes em cada cena (WU; LIM; YANG, 2013; WU; LIM; YANG, 2015). Cada sequência pode ser classificadas em um ou mais dos onze atributos a seguir:

1. Variação de iluminação - a iluminação da cena apresenta variações ao longo do vídeo, dificultando o correto rastreamento do objeto.
2. Variação de escala - o objeto se afasta ou aproxima da câmera, fazendo com que a razão entre a *bounding box* do quadro inicial e a do quadro atual esteja acima de um limiar τ_s, $\tau_s > 1$.
3. Oclusão - o objeto alvo sofre oclusão por outros objetos da cena, ficando parcialmente ou totalmente por alguns instantes, levando à mudanças na aparência do objeto rastreado.
4. Deformação - o objeto sofre deformações não rígidas em sua aparência, fazendo com que a sua forma mude ao longo do vídeo.
5. Desfoque de movimento - a região do alvo sofre desfoque devido ao movimento do objeto ou da câmera.

6. Movimento rápido - o objeto se move muito rapidamente entre os quadros do vídeo, i.e., a posição do objeto no *ground truth* entre um quadro e outro é maior que τ_m pixels ($\tau_m = 20$).
7. Rotação no plano - durante o rastreamento, o movimento do alvo causa uma rotação da aparência do objeto no plano da imagem.
8. Rotação fora do plano - devido ao movimento do objeto e a mudança de angulo de visão da câmera, o alvo sofre uma rotação fora do plano da imagem.
9. Fora de visão - alguma parte do alvo deixa de aparecer na cena por alguns instantes.
10. Plano de fundo confuso - a região de fundo da cena próxima ao alvo tem uma cor ou textura semelhante ao objeto rastreado.
11. Baixa resolução - o número de pixels dentro da *bounding box* no *ground truth* é menor que τ_r ($\tau_r = 400$).

Os resultados quantitativos são relatados utilizando taxa de precisão (*Precison Rate*), taxa de sucesso (*Success Rate*) e a área sob a curva (AUC - *Area Under the Curve*), seguindo o protocolo *One-Pass Evaluation* (OPE), onde o rastreador é executado uma vez desde o quadro inicial até o quadro final, para cada sequência (WU; LIM; YANG, 2015).

A taxa de precisão é definida como a porcentagem de quadros rastreados corretamente para diferentes valores de limiares de distância (em pixels). O objeto é considerado como corretamente rastreado se a distância Euclidiana entre a posição do centro do objeto indicado pelo *ground truth* e a posição estimada do centro do objeto estiver abaixo de um dado limiar. Os resultados relatados na literatura utilizam o valor de 20 pixels como limiar para reportar como sendo a taxa de precisão. Assim, a taxa reporta a porcentagem de quadros em que a posição estimada do objeto está até 20 pixels de distância da posição indicada no *ground truth*.

A taxa de sucesso de sobreposição é definida como a porcentagem de quadros em que a taxa de sobreposição entre a *bounding box* estimada (A_t) e a *bounding box* do *ground truth* (A_g) é maior que um determinado limiar. A taxa de sobreposição é a calculada como a área de intersecção sobre a área de união entre a *bounding box* estimada e a *bounding box* do *ground truth*, formalmente definida como $[\#(A_t \cap A_g)/(\#(A_t \cup A_g)]$, onde $\#(A_t)$ e $\#(A_g)$ representa o número de pixels na região da *bounding box* estimada e da *bounding box* do *ground truth*, respectivamente. Os gráficos que exibem as taxas de sucesso mostram a porcentagem de quadros que possuem taxa de sobreposição acima de determinado limiar, para diferentes valores de limiares de sobreposição, e os resultados

são relatados utilizando os valores da área sob a curva (AUC) (WU; LIM; YANG, 2015).

5.3 Resultados Experimentais

Nós comparamos nosso método com métodos de rastreamento clássicos e também com métodos atuais representativos do estado da arte que utilizam diversas abordagens, incluindo métodos que utilizam aprendizagem profunda - DLT (WANG; YEUNG, 2013), métodos que utilizam um ou mais classificadores discriminativos, - SCM (Wei Zhong; Huchuan Lu; Ming-Hsuan Yang, 2014), Struck (HARE et al., 2016), TGPR (GAO et al., 2014) e MEEM (ZHANG; MA; SCLAROFF, 2014), e outros métodos que também utilizam filtros de correlação, com diferentes abordagens, como multicanais e feições HoG - KCF (HENRIQUES et al., 2015), feições extraídas usando CNNs - CF2 (MA et al., 2015), filtros de correlação baseados em partes do objeto - DPCF (AKIN et al., 2016), uso de múltiplas feições- SAMF (LI; ZHU, 2014), Staple (BERTINETTO et al., 2016), CFIT (HU et al., 2017) e CLIP (LIU et al., 2020). Os resultados dos métodos comparativos foram obtidos diretamente através do código fornecido pelos autores, ou então a partir do resultado gerada pelos autores, que geralmente é um arquivo contendo as posições estimadas para os objetos em cada um dos quadros do vídeo. Para os método CLIP (LIU et al., 2020) não foi possível obter essas posições estimadas quadro a quadro, por isso, nos gráficos são exibidos apenas o valor do resultado final, obtido diretamente do artigo dos autores.

Nas seção a seguir são apresentados os resultados dos nossos testes para cada uma das bases de dados comparadas: OTB-2013 e OTB-2015.

5.3.1 Base de Dados OTB-2013

A Tabela 5.1 exibe as taxas de precisão para a base de testes OTB-2013 utilizando o método proposto e também os métodos comparativos. O melhor e o segundo melhor resultado estão destacados em verde e azul, respectivamente. Como observado na Tabela 5.1, o método proposto obteve o melhor resultado geral, 90.7%, que leva em conta todas as 50 sequências de vídeos, desconsiderando os atributos individuais. O segundo melhor resultado geral foi obtido pelo método CF2 (89.1%), seguido pelo método MEEM (83.0%), com o terceiro melhor resultado. Considerando cada um dos diferentes atributos

presentes nos vídeos, o método proposto obteve o melhor resultado em nove dos onze atributos, perdendo apenas nos atributos fora de visão e baixa resolução, mas ainda assim obtendo o segundo melhor resultado em ambos os atributos. O método proposto se destacou nas sequências que possuem desfoque de movimento e movimentos rápidos. Na avaliação da média dos atributos, o método proposto também obteve o melhor resultado, 86.9%, deixando o método CF2 com o segundo melhor resultado, 85.2%, e o método MEEM com o terceiro melhor, 79.1%.

Tabela 5.1: Taxa de precisão para a base de dados OTB-2013.

Atributos	Proposto	CF2	CFIT	DPCF	Staple	SAMF	MEEM	KCF	SCM	TGPR	TLD	Struck	DLT
Geral	**0.907**	**0.891**	**0.827**	**0.828**	**0.793**	**0.785**	**0.830**	**0.741**	**0.649**	**0.705**	**0.608**	**0.656**	**0.548**
Variação de iluminação	**0.868**	0.844	0.753	0.804	0.741	0.682	0.766	0.728	0.559	0.644	0.498	0.552	0.479
Variação de escala	**0.894**	0.880	0.782	0.753	0.733	0.723	0.785	0.679	0.672	0.620	0.606	0.639	0.606
Oclusão	**0.889**	0.877	0.867	0.862	0.787	0.839	0.799	0.749	0.642	0.680	0.537	0.588	0.517
Deformação	**0.892**	0.881	0.798	0.862	0.812	0.810	0.846	0.740	0.583	0.700	0.465	0.553	0.481
Desfoque de movimento	**0.873**	0.844	0.651	0.722	0.688	0.564	0.715	0.650	0.339	0.537	0.518	0.551	0.427
Movimento rápido	**0.818**	0.790	0.634	0.701	0.643	0.608	0.742	0.602	0.333	0.493	0.551	0.604	0.435
Rotação no plano	**0.885**	0.868	0.791	0.754	0.773	0.714	0.800	0.725	0.597	0.675	0.584	0.617	0.510
Rotação fora do plano	**0.886**	0.869	0.817	0.807	0.773	0.767	0.840	0.729	0.618	0.682	0.579	0.616	0.545
Fora de visão	0.727	0.695	0.629	0.689	0.679	0.636	**0.727**	0.650	0.429	0.505	0.576	0.539	0.505
Plano de fundo cofuso	**0.902**	0.885	0.732	0.811	0.753	0.676	0.797	0.752	0.578	0.717	0.428	0.585	0.440
Baixa resolução	0.923	**0.935**	0.747	0.567	0.695	0.709	0.888	0.630	0.661	0.438	0.566	0.550	0.554
Média	**0.869**	**0.852**	**0.746**	**0.757**	**0.734**	**0.703**	**0.791**	**0.694**	**0.546**	**0.608**	**0.537**	**0.581**	**0.500**

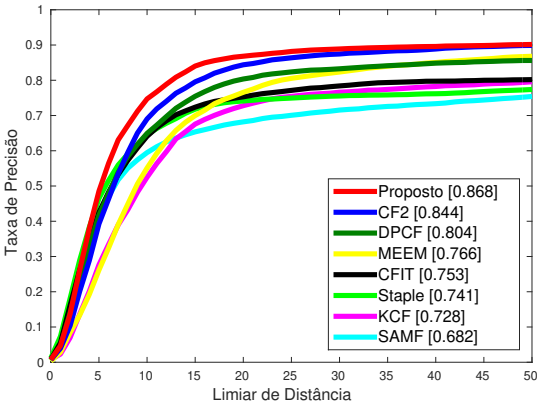
A Figura 5.1 e a Figura 5.2 exibem as taxas de precisão para diferentes valores de limiares de distância, variando de 0 até 50 pixel. Nos gráficos são exibidos apenas os oito melhores métodos para facilitar a visualização. Podemos notar que o método proposto apresenta um desempenho promissor em termos de taxa de precisão, sendo ligeiramente superior ao método CF2, que também faz uso de feições extraídas de Redes Neurais Convolucionais Profundas.

A Tabela 5.2 apresenta uma comparação em termos de taxa de sucesso. O melhor e o segundo melhor resultado estão destacados em verde e azul, respectivamente. Analisar apenas a taxa de precisão pode levar a interpretações incorretas sobre o desempenho dos métodos, pois ela leva em conta apenas a posição central do objeto, sem considerar mudanças de escala e outras deformações. Assim, um método pode indicar que esta rastreando corretamente o centro do objeto, mesmo que não esteja mais rastreando o objeto inteiro, e sim apenas uma parte dele. Portanto, a taxa de sucesso é de grande importância, pois nos permite analisar se o objeto como um todo está sendo corretamente rastreado, comparando a *bounding box* estimada do objeto com a região real do objeto, definida no *ground truth*.

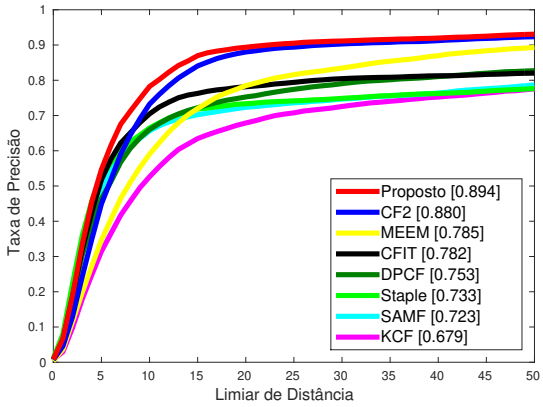
Na Tabela 5.2 podemos observar que o método proposto obteve o melhor resultado geral, 65.3%, e o segundo melhor método foi o CFIT, com 61.4%, enquanto os métodos DPCF e CF2 empataram com o terceiro melhor resultado geral, ambos com 60.5%. Ana-

Figura 5.1: Gráficos com as taxas de precisão OTB-2013 - parte 1.

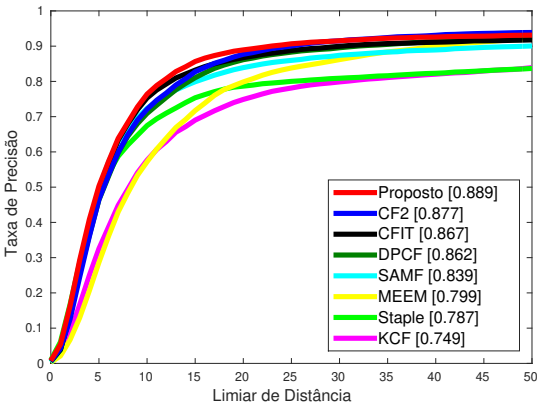
(a) Variação de iluminação



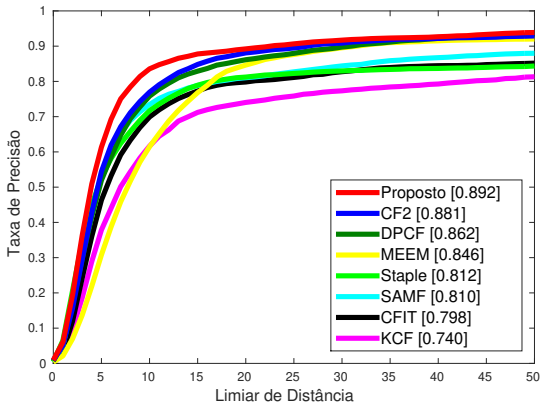
(b) Variação de escala



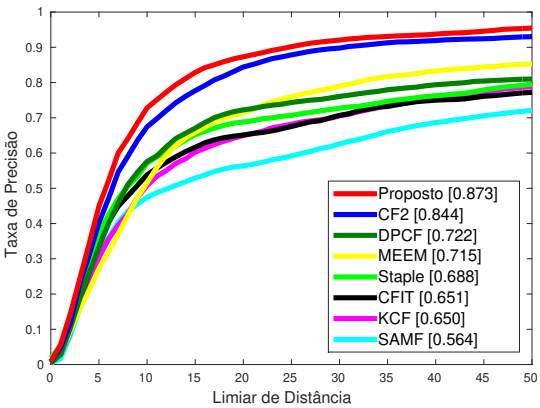
(c) Oclusão



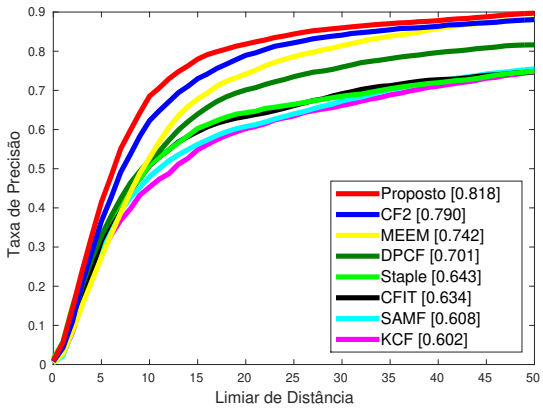
(d) Deformação



(e) Desfoque de movimento



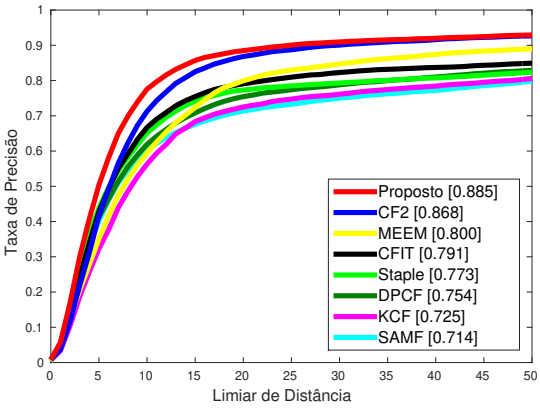
(f) Movimento rápido



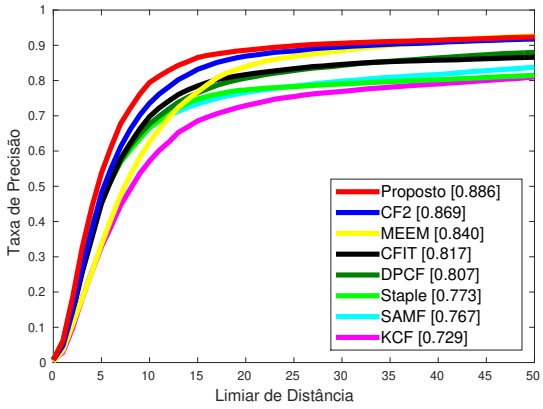
Gráficos com as taxas de precisão para as sequências contendo os atributos variação de iluminação, variação de escala, oclusão, deformação, desfoque de movimento, e movimento rápido para a base de dados OTB-2013.

Figura 5.2: Gráficos com as taxas de precisão OTB-2013 - parte 2.

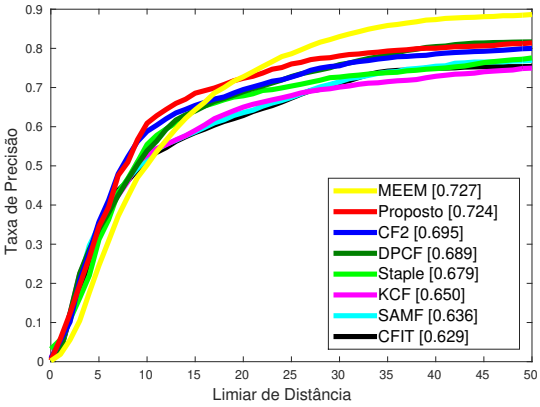
(a) Rotação no plano



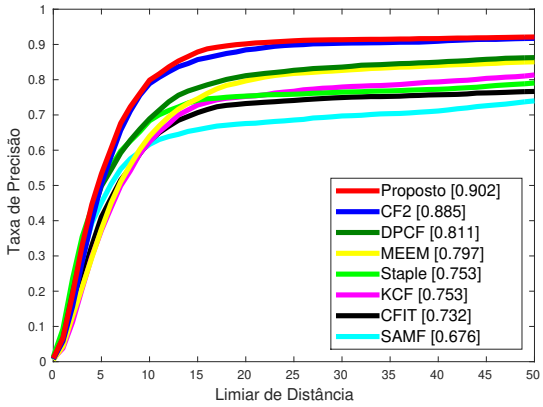
(b) Rotação fora do plano



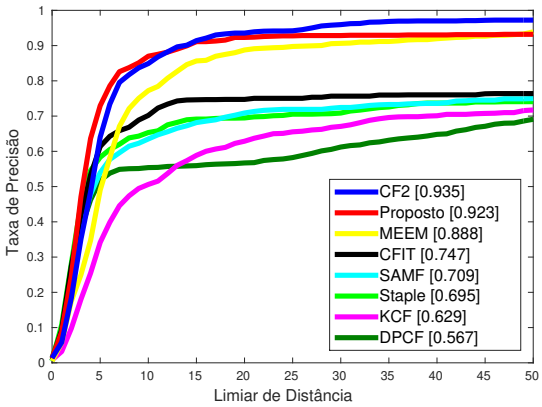
(c) Fora de visão



(d) Plano de fundo confuso



(e) Baixa resolução



Gráficos com as taxas de precisão para as sequências contendo os atributos rotação no plano, rotação fora do plano, fora de visão, plano de fundo confuso, e baixa resolução na base de dados OTB-2013.

lisando pelos atributos presentes em cada sequência, podemos ver que o método proposto foi melhor em nove atributos, obtendo o segundo melhor resultado nos outros dois atributos restantes. O método CFIT obteve o melhor resultado em um atributo, oclusão, e o método MEEM obteve o melhor resultado no outro atributo, fora de visão. O método proposto se destacou em situações que apresentam rotações no plano da imagem e também situações que apresentam baixa resolução, e obteve também o melhor resultado na média dos atributos.

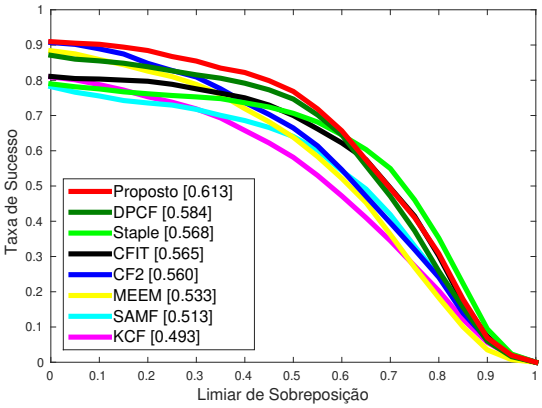
Como mencionado anteriormente, os resultados reportados na Tabela 5.2 utilizam os valores da área sob a curva (AUC) dos gráficos das taxas de sobreposição. Esses gráficos com as taxas de sobreposição podem ser observados na Figura 5.3 e na Figura 5.4, que ilustram as taxas de sucesso para diferentes valores de sobreposição, exibindo na legenda os valores ordenados em termos de área sob a curva para cada um dos oito melhores métodos. Assim, podemos observar melhor a variação de desempenho dos métodos ao longo das sequências de vídeos.

A Figura 5.5 exibe uma comparação visual entre o método proposto e os métodos comparativos em algumas situações desafiadoras. A sequencia *CarScale* apresenta situações de oclusão, variação de escala, rotação no plano da imagem e também rotação fora do plano da imagem. Na sequência *Skating1* temos a presença de variações de iluminação, mudanças de escala, deformações e rotações, enquanto na sequência *Jogging-1* temos a predominância de deformações e oclusões. A sequência *Skiing* apresenta mudanças de escala, rotações, oclusões, deformações,além de baixa resolução. Por fim, a sequência *Couple* apresenta deformações, movimentos rápidos e plano de fundo confuso. Podemos observar que apesar das dificuldades o método proposto foi efetivo no rastreamento, enquanto os métodos comparativos não conseguiram rastrear corretamente os objetos.

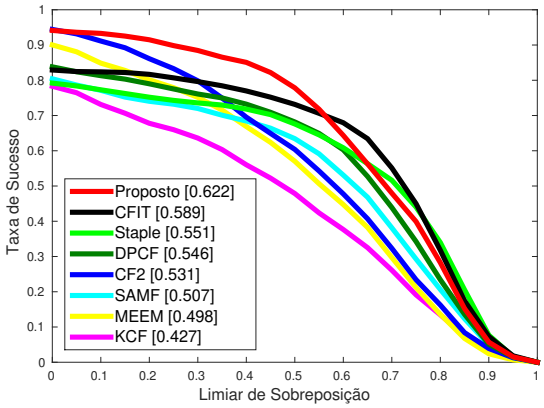
Analisando as duas métricas, podemos ver que o método proposto obteve um melhor desempenho geral, com destaque para situações em que ocorrem mudanças de iluminação, desfoque de movimento e movimentos rápidos, onde o uso das feições extraídas de Redes Neurais Convolucionais Profundas contribuem para uma correta identificação da região do objeto, e também para outras situações em que ocorrem deformações, mudanças de escala, rotações no plano e fora do plano da imagem, e semelhança entre o plano de fundo e o objeto, onde o uso de múltiplos filtros locais para realizar o rastreamento colabora para um melhor resultado.

Figura 5.3: Gráficos com as taxas de sucesso OTB-2013 - parte 1.

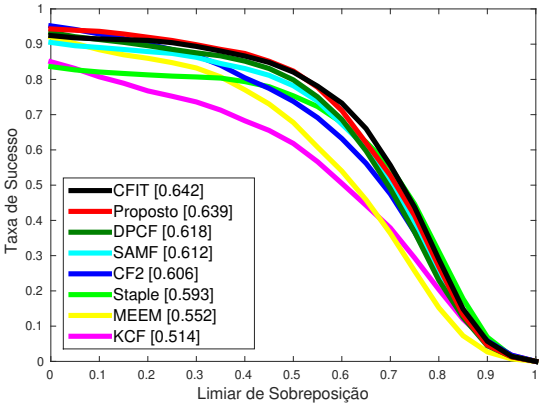
(a) Variação de iluminação



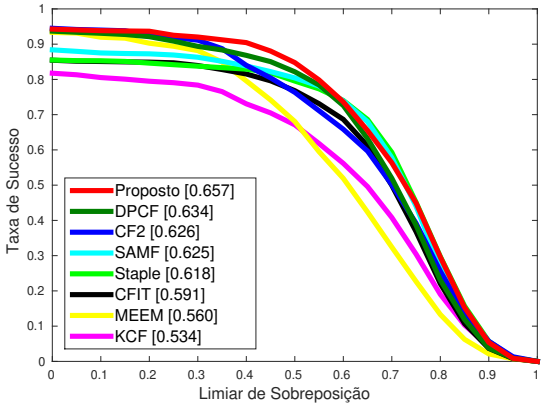
(b) Variação de escala



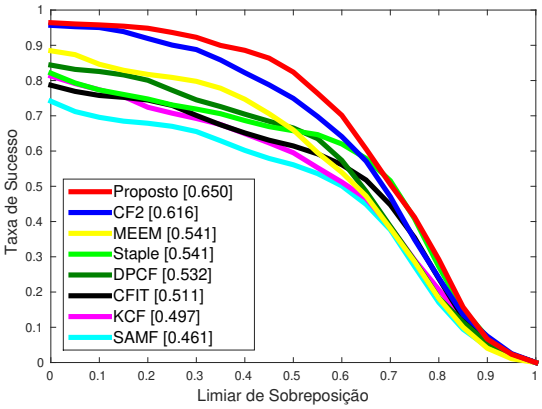
(c) Oclusão



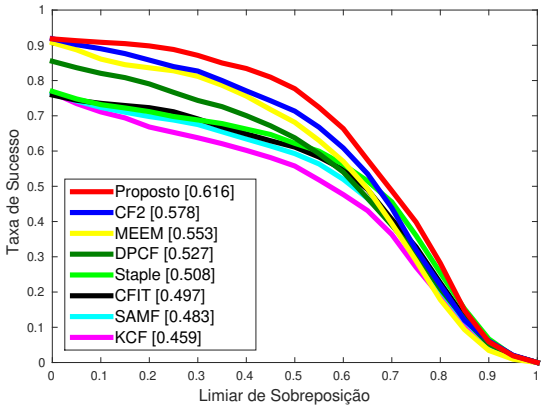
(d) Deformação



(e) Desfoque de movimento



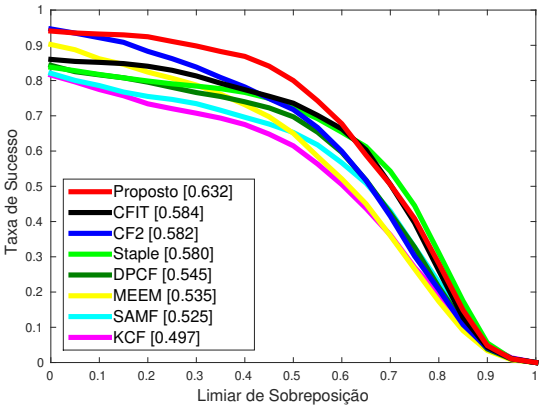
(f) Movimento rápido



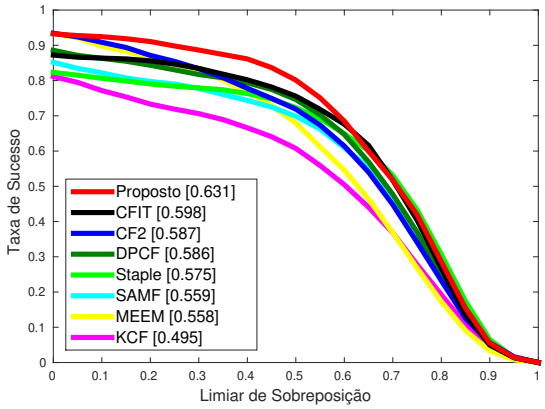
Gráficos com as taxas de sucesso para as sequências contendo os atributos variação de iluminação, variação de escala, oclusão, deformação, desfoque de movimento e movimento rápido na base de dados OTB-2013.

Figura 5.4: Gráficos com as taxas de sucesso OTB-2013 - parte 2.

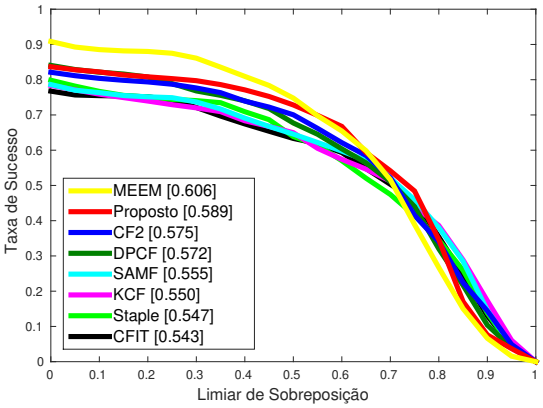
(a) Rotação no plano



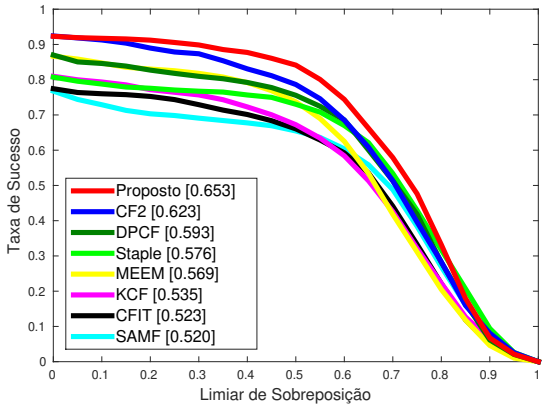
(b) Rotação fora do plano



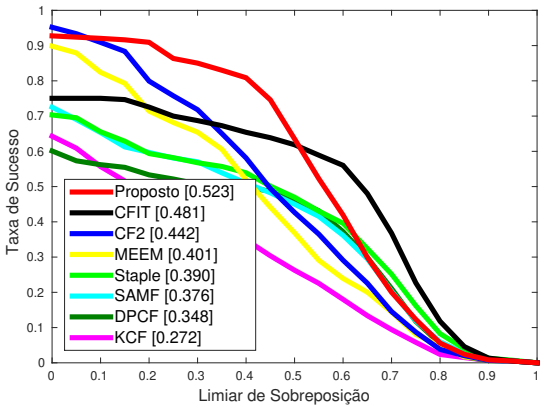
(c) Fora de visão



(d) Plano de fundo confuso



(e) Baixa resolução

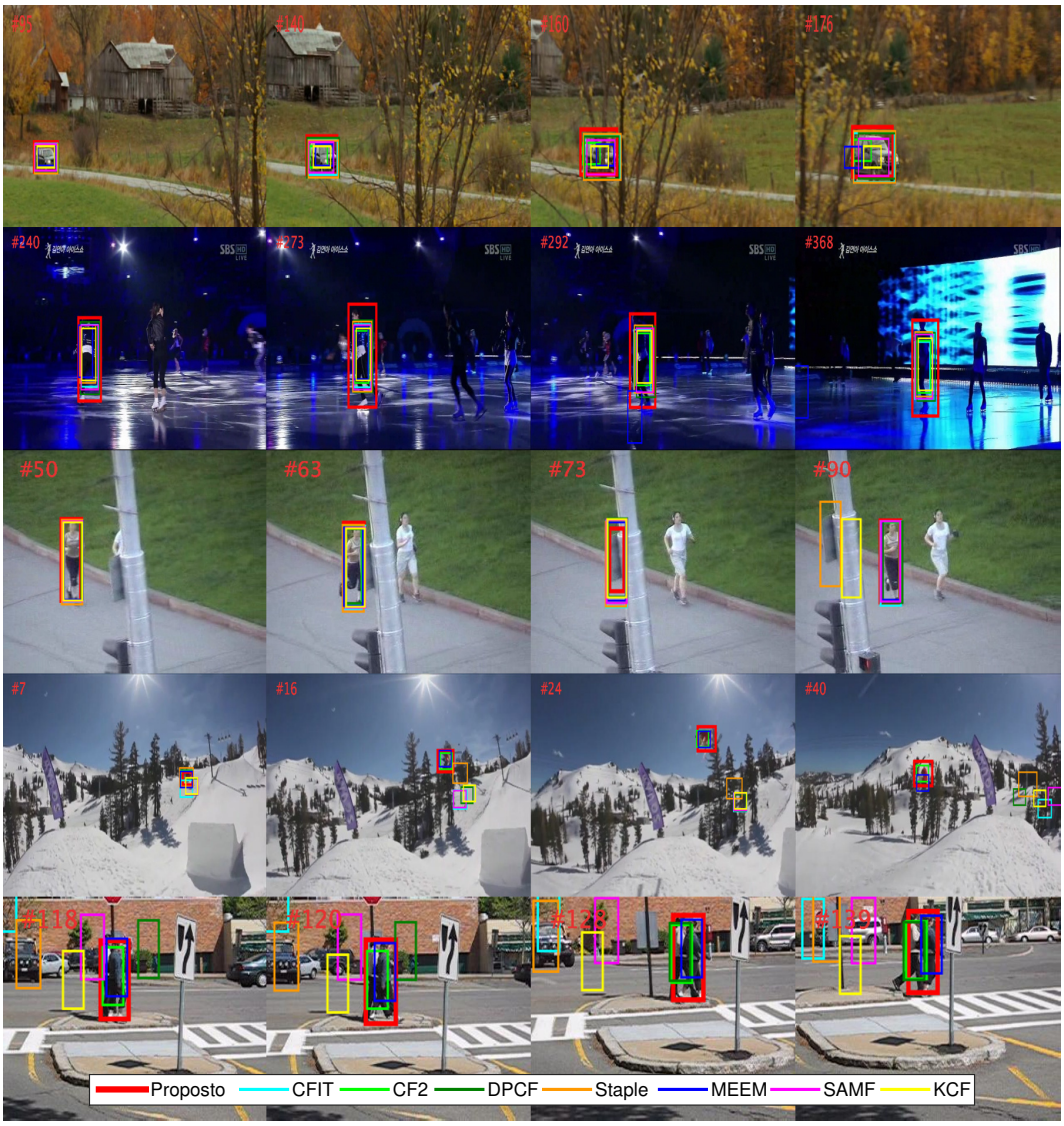


Gráficos com as taxas de sucesso para as sequências contendo os atributos rotação no plano, rotação fora do plano, fora de visão, plano de fundo confuso, e baixa resolução na base de dados OTB-2013.

Tabela 5.2: Taxa de sucesso para a base de dados OTB-2013.

Atributos	Proposto	CF2	CFIT	DPCF	Staple	SAMF	MEEM	KCF	SCM	TGPR	TLD	Struck	DLT
Geral	**0.653**	**0.605**	**0.614**	**0.605**	**0.600**	**0.579**	**0.566**	**0.513**	**0.499**	**0.503**	**0.437**	**0.474**	**0.415**
Variação de iluminação	**0.613**	0.560	0.565	0.584	0.568	0.513	0.533	0.493	0.447	0.476	0.379	0.431	0.380
Variação de escala	**0.622**	0.531	0.589	0.546	0.551	0.507	0.498	0.427	0.518	0.418	0.421	0.425	0.458
Oclusão	0.641	0.606	**0.642**	0.618	0.593	0.612	0.552	0.514	0.490	0.485	0.380	0.430	0.401
Deformação	**0.657**	0.626	0.591	0.634	0.618	0.625	0.560	0.534	0.448	0.515	0.340	0.418	0.350
Desfoque de movimento	**0.650**	0.616	0.511	0.532	0.541	0.461	0.541	0.497	0.298	0.434	0.404	0.433	0.329
Movimento rápido	**0.616**	0.578	0.497	0.527	0.508	0.483	0.553	0.459	0.296	0.396	0.417	0.462	0.353
Rotação no plano	**0.633**	0.582	0.584	0.545	0.580	0.525	0.535	0.497	0.458	0.479	0.416	0.444	0.383
Rotação fora do plano	**0.631**	0.587	0.598	0.586	0.575	0.559	0.558	0.495	0.471	0.486	0.405	0.446	0.406
Fora de visão	0.589	0.575	0.543	0.572	0.547	0.555	**0.606**	0.550	0.361	0.442	0.457	0.459	0.409
Plano de fundo cofuso	**0.653**	0.623	0.523	0.593	0.576	0.520	0.569	0.535	0.450	0.522	0.345	0.458	0.327
Baixa resolução	**0.523**	0.442	0.481	0.348	0.390	0.376	0.401	0.272	0.438	0.246	0.295	0.260	0.330
Média	**0.621**	**0.575**	**0.557**	**0.553**	**0.550**	**0.521**	**0.537**	**0.479**	**0.425**	**0.445**	**0.387**	**0.424**	**0.375**

Figura 5.5: Comparação visual dos métodos de rastreamento OTB-2013.



Avaliação qualitativa dos resultados obtidos pelos métodos de rastreamento nas sequências *CarScale*, *Skating1*, *Jogging-1*, *Skiing* e *Couple*.

5.3.2 Base de Dados OTB-2015

A Tabela 5.3 exibe as taxas de precisão para o método proposto e para os métodos comparativo para as 100 sequências de vídeos da base de testes OTB-2015. O melhor e o segundo melhor resultado estão destacados em verde e azul, respectivamente. Como observado na Tabela 5.3, o método proposto obteve o melhor resultado geral, 87.8%, seguido pelos métodos CLIP e CF2, com 84.5% e 83.7%, respectivamente. Considerando cada um dos atributos presentes nos vídeos, o método proposto obteve o melhor resultado em dez dos onze atributos, obtendo o segundo melhor resultado no atributo restante. O método proposto se destacou nas sequências que apresentam variação de iluminação, variação de escala, e principalmente em situações de baixa resolução, ou seja, situações em que o tamanho do alvo rastreado é pequeno, sendo 10% melhor que o melhor método comparativo nesse atributo. Na avaliação média dos atributos, o método proposto alcançou a melhor média, 84.5%, ficando o método CLIP com a segunda melhor, 80.5% e o método CF2 com a terceira melhor, 80.1%.

Tabela 5.3: Taxa de precisão para a base de dados OTB-2015.

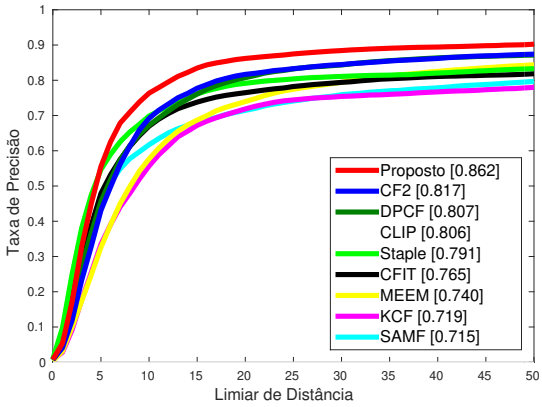
Atributos	Proposto	CLIP	CF2	CFIT	DPCF	Staple	SAMF	MEEM	KCF	SCM	TGPR	TLD	Struck
Geral	**0.878**	**0.845**	**0.837**	**0.802**	**0.775**	**0.783**	**0.751**	**0.781**	**0.695**	**0.572**	**0.643**	**0.592**	**0.635**
Variação de iluminação	**0.862**	0.806	0.817	0.765	0.807	0.791	0.715	0.740	0.719	0.587	0.617	0.547	0.554
Variação de escala	**0.848**	0.810	0.798	0.771	0.723	0.726	0.704	0.736	0.633	0.576	0.611	0.583	0.609
Oclusão	**0.835**	**0.835**	0.767	0.795	0.741	0.725	0.725	0.740	0.629	0.571	0.610	0.540	0.558
Deformação	**0.848**	0.821	0.791	0.712	0.727	0.748	0.686	0.754	0.617	0.570	0.653	0.495	0.564
Desfoque de movimento	**0.837**	0.762	0.802	0.731	0.751	0.706	0.653	0.729	0.599	0.268	0.529	0.527	0.575
Movimento rápido	**0.843**	0.778	0.814	0.711	0.706	0.695	0.653	0.751	0.620	0.330	0.548	0.580	0.631
Rotação no plano	**0.872**	0.789	0.854	0.800	0.737	0.770	0.721	0.794	0.701	0.543	0.659	0.604	0.626
Rotação fora do plano	**0.857**	0.845	0.806	0.792	0.754	0.737	0.738	0.794	0.676	0.581	0.655	0.577	0.616
Fora de visão	0.715	**0.760**	0.674	0.698	0.611	0.658	0.624	0.681	0.498	0.422	0.493	0.456	0.462
Plano de fundo cofuso	**0.845**	0.817	0.843	0.745	0.782	0.766	0.689	0.746	0.712	0.577	0.593	0.456	0.548
Baixa resolução	**0.930**	0.834	0.842	0.831	0.711	0.690	0.761	0.802	0.665	0.761	0.622	0.627	0.674
Média	**0.845**	**0.805**	**0.801**	**0.759**	**0.732**	**0.728**	**0.697**	**0.752**	**0.643**	**0.526**	**0.599**	**0.545**	**0.583**

A taxa de precisão para diferentes valores de limiares de distância podem ser observadas na Figura 5.6 e na Figura 5.7. Podemos observar que o método proposto se manteve consistente para os diferentes limiares de distância em cada atributo. Novamente atingindo os melhores resultados para a maioria dos atributos e também o melhor resultado no geral, com o método CF2 atingindo o segundo melhor resultado geral e o método CFIT o terceiro melhor resultado.

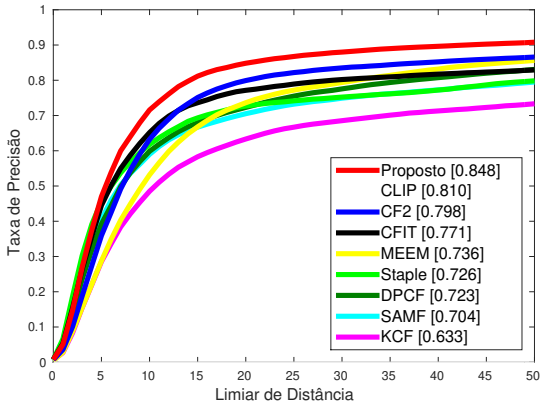
A Tabela 5.4 apresenta a comparação utilizando a taxa de sucesso, que permite ter fazer uma avaliação melhor do desempenho do método, com o melhor e o segundo melhor resultado destacados em verde e azul, respectivamente. O método proposto alcançou o melhor resulto geral, com 63.3%, ficando acima do método CLIP, que alcançou o

Figura 5.6: Gráficos com as taxas de precisão OTB-2015 - parte 1.

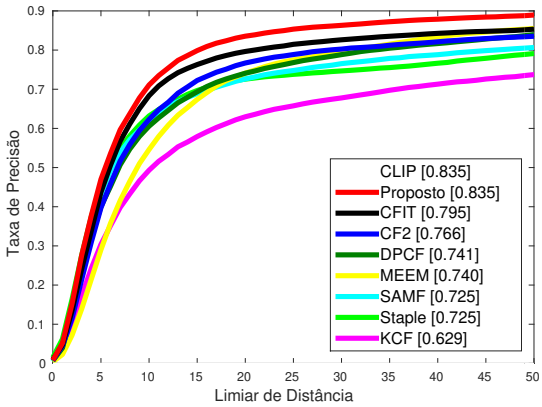
(a) Variação de iluminação



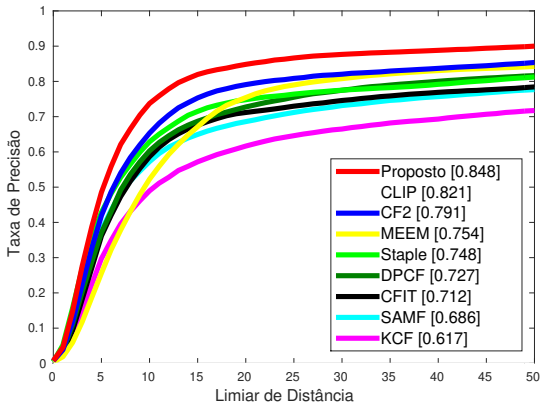
(b) Variação de escala



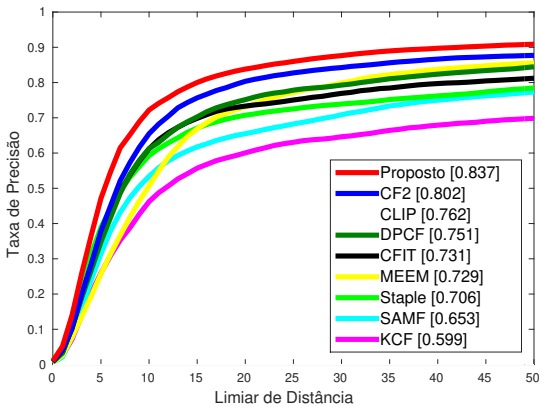
(c) Oclusão



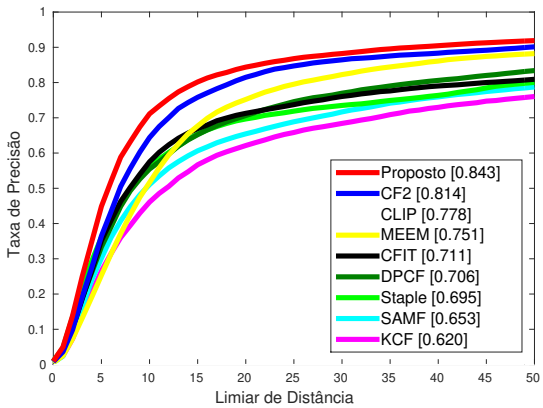
(d) Deformação



(e) Desfoque de movimento



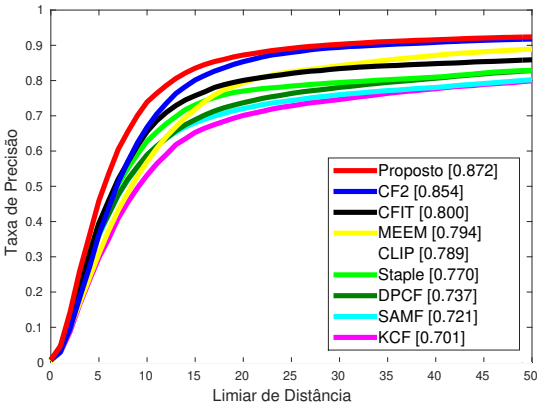
(f) Movimento rápido



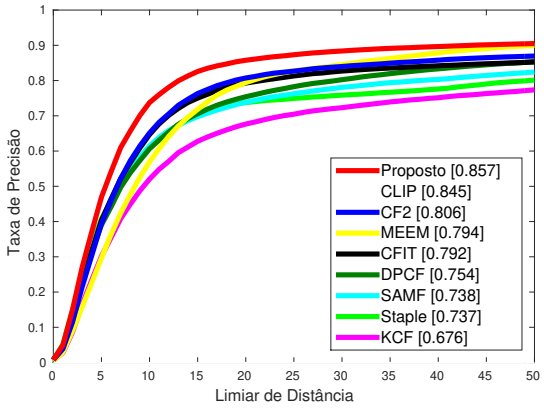
Gráficos com as taxas de precisão para as sequências contendo os atributos variação de iluminação, variação de escala, oclusão, deformação, desfoque de movimento, e movimento rápido para a base de dados OTB-2015.

Figura 5.7: Gráficos com as taxas de precisão OTB-2015 - parte 2.

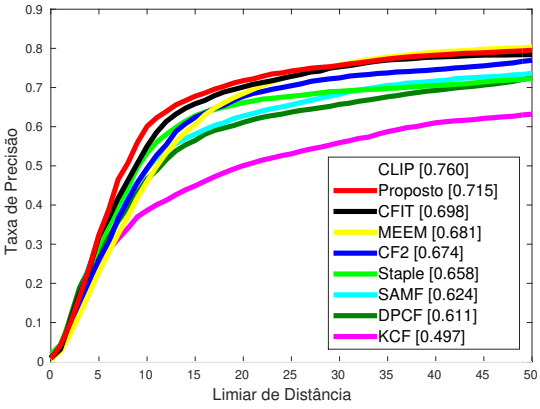
(a) Rotação no plano



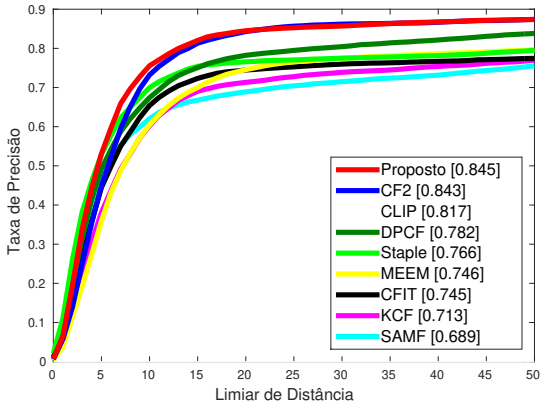
(b) Rotação fora do plano



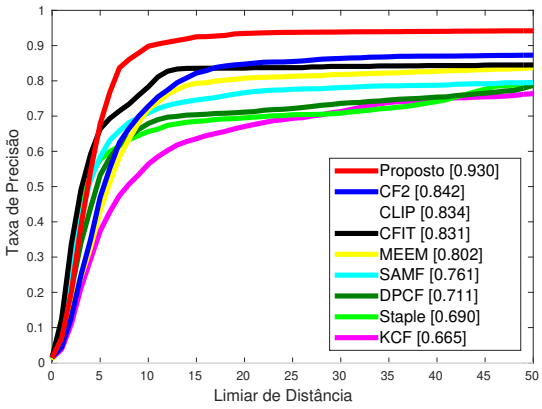
(c) Fora de visão



(d) Plano de fundo confuso



(e) Baixa resolução



Gráficos com as taxas de precisão para as sequências contendo os atributos rotação no plano, rotação fora do plano, fora de visão, plano de fundo confuso, e baixa resolução na base de dados OTB-2015.

segundo melhor resultado, com 62.8%, e do método CFIT, que obteve o terceiro melhor resultado, com 60.0%. Analisando individualmente cada atributo, observamos que o método proposto foi melhor em sete atributos, obtendo o segundo melhor resultado em cada um dos atributos restantes. O método CLIP obteve o melhor resultado em três atributos, e o método CFIT obteve o melhor resultado em um atributo. O método proposto se destacou em sequências que possuem desfoque de movimento e movimentos rápidos, e obteve também o melhor resultado na média dos atributos.

Tabela 5.4: Taxa de sucesso para a base de dados OTB-2015.

Atributos	Proposto	CLIP	CF2	CFIT	DPCF	Staple	SAMF	MEEM	KCF	SCM	TGPR	TLD	Struck
Geral	**0.633**	**0.628**	**0.562**	**0.600**	**0.565**	**0.581**	**0.553**	**0.530}**	**0.477**	**0.445**	**0.458**	**0.424**	**0.424**
Variação de iluminação	**0.625**	0.620	0.540	0.589	0.585	0.598	0.534	0.517	0.479	0.479	0.448	0.411	0.430
Variação de escala	**0.595**	0.589	0.485	0.572	0.518	0.525	0.495	0.470	0.394	0.439	0.405	0.399	0.406
Oclusão	0.606	**0.622**	0.525	0.593	0.546	0.548	0.540	0.504	0.443	0.435	0.430	0.370	0.405
Deformação	**0.595**	0.593	0.530	0.525	0.522	0.554	0.509	0.489	0.455	0.413	0.460	0.344	0.403
Desfoque de movimento	**0.648**	0.598	0.585	0.570	0.574	0.546	0.525	0.556	0.459	0.200	0.429	0.424	0.456
Movimento rápido	**0.636**	0.600	0.570	0.548	0.533	0.537	0.507	0.542	0.459	0.299	0.421	0.446	0.469
Rotação no plano	**0.614**	0.574	0.559	0.579	0.524	0.552	0.519	0.529	0.469	0.408	0.462	0.426	0.447
Rotação fora do plano	0.606	**0.616**	0.534	0.577	0.546	0.534	0.536	0.525	0.453	0.435	0.456	0.391	0.435
Fora de visão	0.540	**0.579**	0.474	0.531	0.482	0.481	0.480	0.488	0.393	0.333	0.373	0.338	0.359
Plano de fundo cofuso	**0.619**	0.610	0.585	0.547	0.575	0.574	0.525	0.519	0.498	0.462	0.428	0.352	0.427
Baixa resolução	0.533	0.518	0.388	**0.551**	0.410	0.396	0.425	0.382	0.290	0.478	0.344	0.346	0.313
Média	**0.602**	**0.593**	**0.525**	**0.562**	**0.529**	**0.531**	**0.509**	**0.502**	**0.436**	**0.398**	**0.423**	**0.386**	**0.414**

Os gráficos com as taxas de sobreposição para diferentes valores de limiares são exibidos na Figura 5.8 e na Figura 5.9. Como nos gráficos anteriores, são exibidos os resultados para os oito melhores métodos comparativos, e as legendas exibem os valores da área sob a curva dos gráficos.

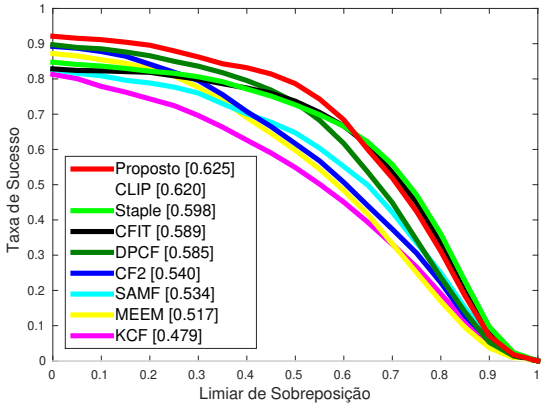
A Figura 5.10 faz uma uma comparação visual entre o método proposto e os métodos comparativos em em situações diversas. Na sequência *MotorRolling* temos a predominância de rotações , movimentos rápidos e variações de escalas, enquanto na sequência *Panda* apresenta baixa resolução, mudanças de escala, rotações, e deformações. A sequência *Biker* contém variações de escala, rotações, movimentos rápidos e desfoque de movimento, além de possuir baixa resolução. A sequência *Toy* apresenta rotações tanto no plano da imagem quanto fora do plano da imagem, além de apresentar e variações de escala e movimentos rápidos e plano de fundo confuso. Na sequência *KiteSurf* temos a presença de rotações no plano e fora do plano da imagem, mudanças de iluminação, e oclusões. Pelas ilustrações vemos que o método proposto conseguiu obter bons resultados, superando os métodos comparativos nos diversos cenários.

Analisando as duas métricas, mais uma vez percebemos que o método proposto obteve um melhor desempenho geral, com destaque para situações em que ocorrem mudanças de iluminação, desfoque de movimento e movimentos rápidos, beneficiados pelo tipo de feição utilizada no método, e outras situações em que ocorrem rotações do objeto,

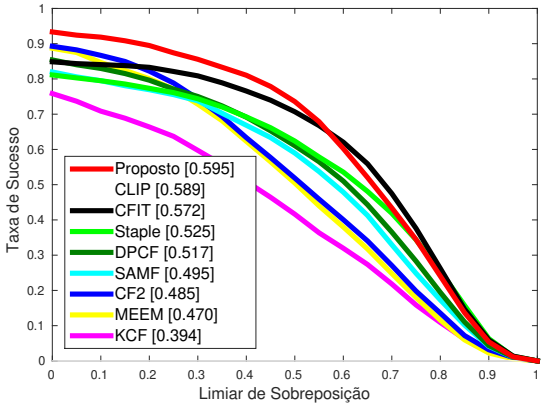
tanto no plano quanto fora do plano da imagem, e onde o objeto está fora de visão, onde o uso dos múltiplos filtros se mostrou vantajoso para melhorar o resultado do rastreamento.

Figura 5.8: Gráficos com as taxas de sucesso OTB-2015 - parte 1.

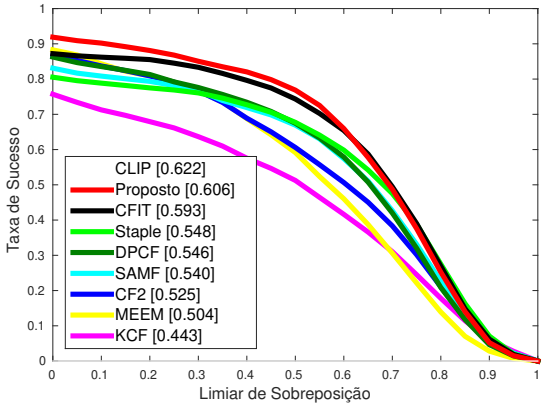
(a) Variação de iluminação



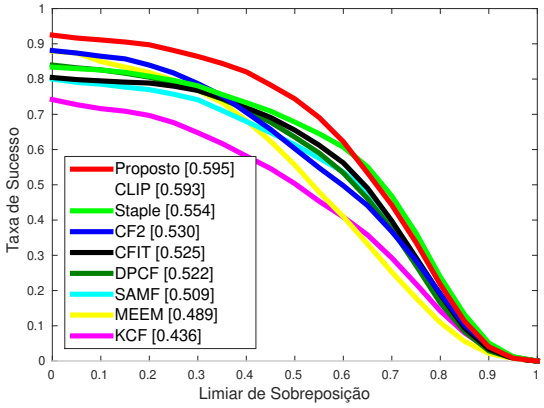
(b) Variação de escala



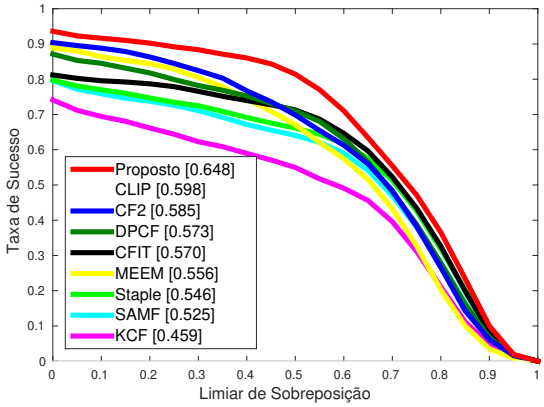
(c) Oclusão



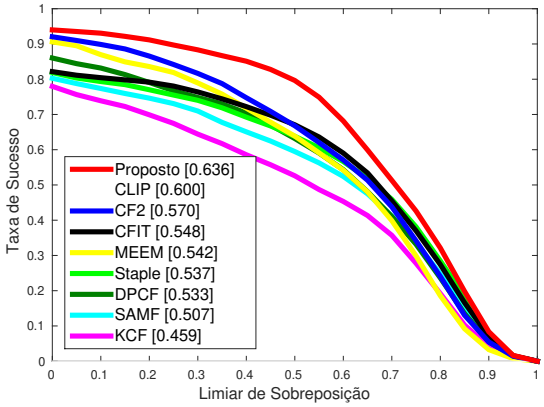
(d) Deformação



(e) Desfoque de movimento



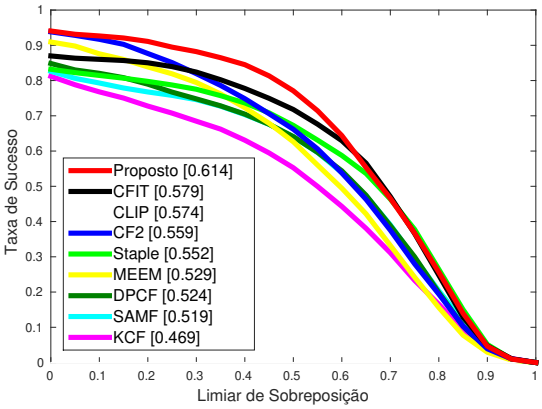
(f) Movimento rápido



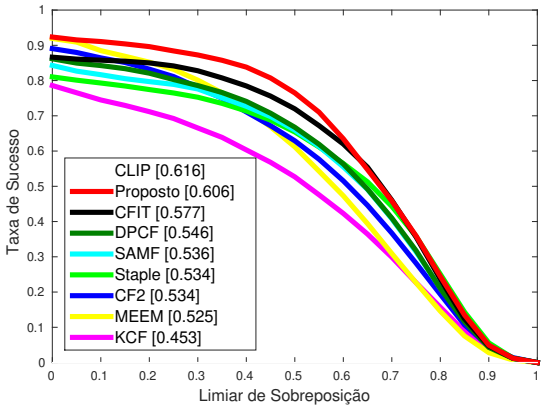
Gráficos com as taxas de sucesso para as sequências contendo os atributos variação de iluminação, variação de escala, oclusão, deformação, desfoque de movimento e movimento rápido na base de dados OTB-2015.

Figura 5.9: Gráficos com as taxas de sucesso OTB-2015 - parte 2.

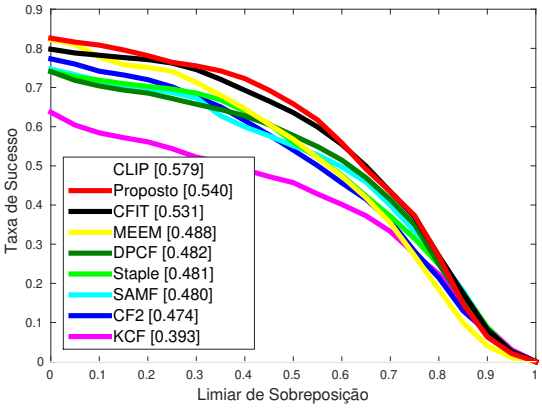
(a) Rotação no plano



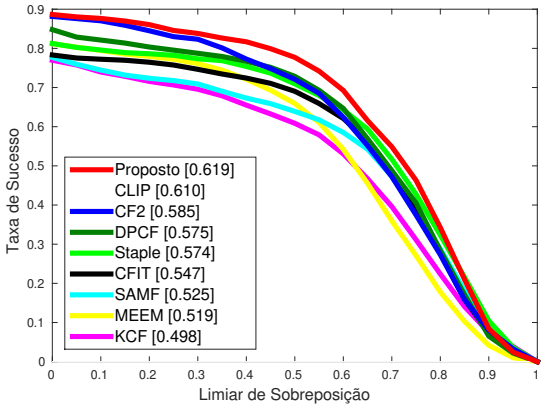
(b) Rotação fora do plano



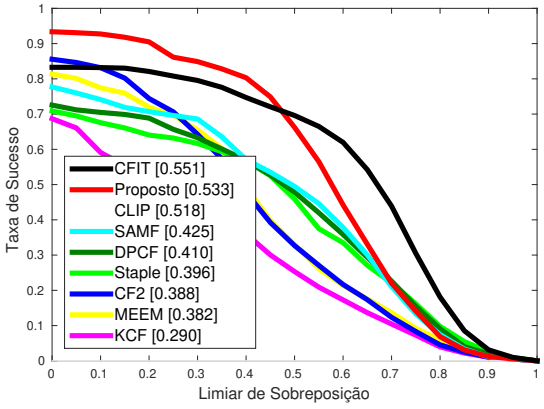
(c) Fora de visão



(d) Plano de fundo confuso

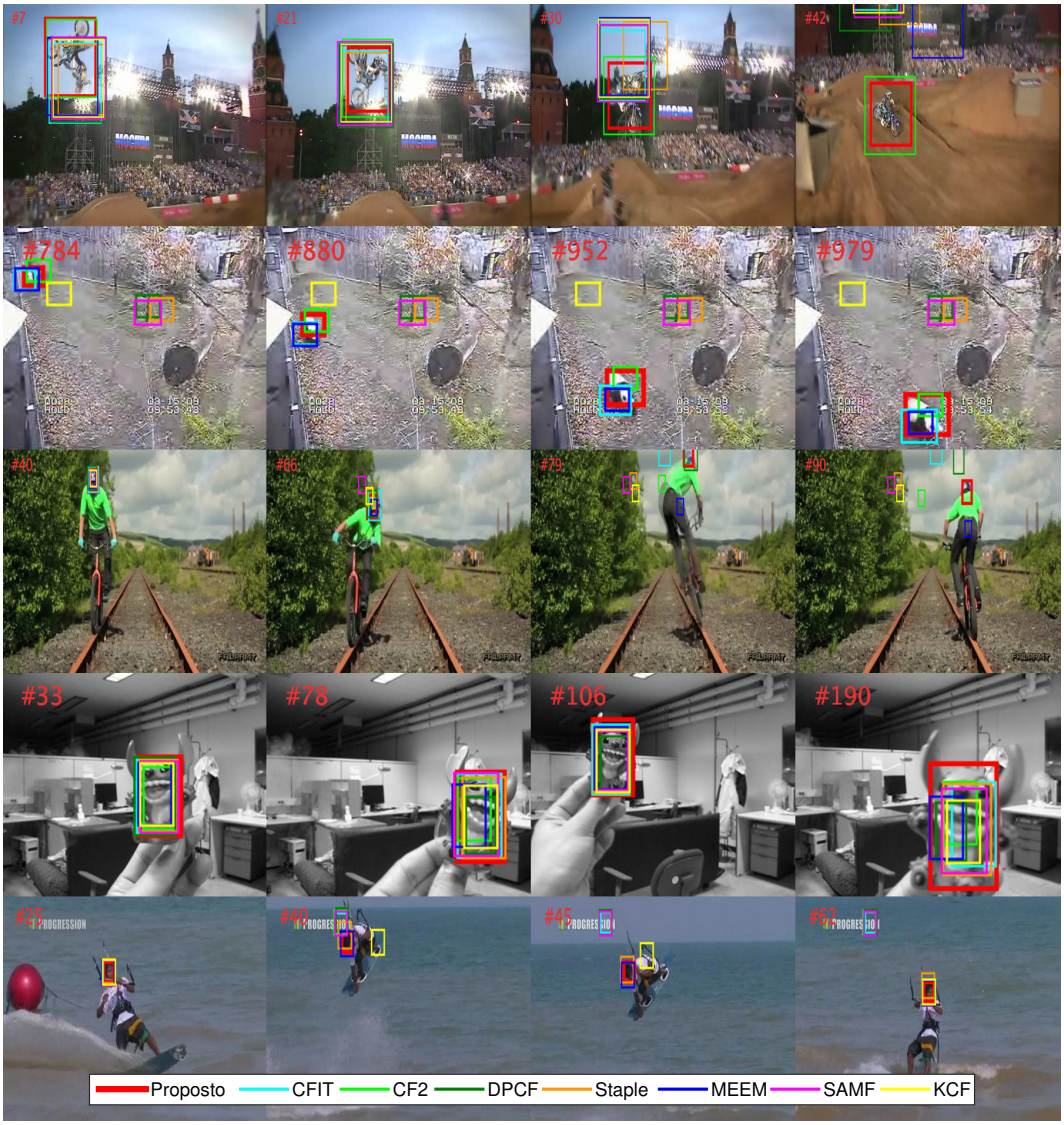


(e) Baixa resolução



Gráficos com as taxas de sucesso para as sequências contendo os atributos rotação no plano, rotação fora do plano, fora de visão, plano de fundo confuso, e baixa resolução na base de dados OTB-2015.

Figura 5.10: Comparação visual dos métodos de rastreamento OTB-2015.



Avaliação qualitativa dos resultados obtidos pelos métodos de rastreamento nas sequências *MotorRolling*, *Panda*, *Biker*, *Toy* e *KiteSurf*.

5.4 Estudo de Ablação

Neste trabalho foram empregados dois esquemas para melhorar a performance dos filtros de correlação, o esquema com múltiplos filtros colaborativos e o esquema de atualização dos modelos. Para avaliar o comportamento do método em diferentes configurações realizamos experimentos com uma versão do método proposto sem o uso do esquema de múltiplos filtros, utilizando apenas o esquema de atualização dos modelos e apenas um único filtro global para rastrear os objetos, e outra versão sem o esquema de atualização, com apenas o uso dos múltiplos filtros colaborativos, considerando os valores obtidos nos filtros de correlação como sempre confiáveis e atualizando os modelos em todos os quadros.

A Tabela 5.5 exibe a taxa de precisão para a base de dados OTB-2013 para a versão original do método e também para as outras duas configurações. A versão com apenas o esquema de múltiplos filtros obteve 82.3% no resultado geral enquanto a versão utilizando apenas o esquema de atualização dos filtros se saiu melhor, obtendo 84.2%. Comparando os atributos das duas configurações podemos observar que a versão com o esquema de múltiplos filtros foi melhor em situações com variação de iluminação, rotações no plano da imagem, baixa resolução e quando o objeto e fundo são semelhantes. Para os outros atributos, os esquema de atualização do filtros foi mais relevante para melhorar a performance e atingir resultados melhores.

Tabela 5.5: Taxa de precisão para a base de dados OTB-2013 comparando a versão original do método proposto, a versão com apenas o esquema de múltiplos filtros colaborativos e a versão com apenas o esquema de atualização de modelos.

Atributos	**Proposto**	**Múltiplos Filtros**	**Esquema de Atualização**
Geral	**0.907**	**0.823**	**0.842**
Variação de iluminação	0.869	0.791	0.745
Variação de escala	0.895	0.776	0.796
Oclusão	0.890	0.748	0.847
Deformação	0.892	0.826	0.831
Desfoque de movimento	0.873	0.776	0.803
Movimento rápido	0.820	0.720	0.743
Rotação no plano	0.886	0.835	0.792
Rotação fora do plano	0.887	0.797	0.807
Fora de visão	0.724	0.567	0.604
Plano de fundo cofuso	0.903	0.837	0.806
Baixa resolução	0.923	0.792	0.745

A Tabela 5.6 exibe os resultados para a base OTB-2013 em termos de taxa de sucesso. Aqui a versão com o esquema de múltiplos filtros obteve o melhor resultado no geral, 60.2%, e a versão com apenas o esquema de atualização obteve 56.7%. Considerando os atributos individuais, dessa vez a versão com o esquema de atualização se destacou apenas em situações de oclusão e objetos fora de visão, com a versão utilizando múltiplos filtros obtendo os melhores resultados na maioria dos atributos.

Tabela 5.6: Taxa de sucesso para a base de dados OTB-2013 comparando a versão original do método proposto, a versão com apenas o esquema de múltiplos filtros colaborativos e a versão com apenas o esquema de atualização de modelos.

Atributos	**Proposto**	**Múltiplos Filtros**	**Esquema de Atualização**
Geral	**0.653**	**0.602**	**0.567**
Variação de iluminação	0.615	0.567	0.502
Variação de escala	0.623	0.563	0.477
Oclusão	0.641	0.546	0.570
Deformação	0.657	0.594	0.595
Desfoque de movimento	0.650	0.583	0.580
Movimento rápido	0.618	0.556	0.542
Rotação no plano	0.633	0.611	0.526
Rotação fora do plano	0.632	0.577	0.541
Fora de visão	0.589	0.487	0.518
Plano de fundo cofuso	0.655	0.613	0.570
Baixa resolução	0.523	0.451	0.346

Como podemos observar nos resultados, a versão sem o esquema de atualização teve um desempenho pior quando levamos em conta a taxa de precisão, enquanto a versão sem sem os múltiplos filtros foi pior considerando a taxa de sucesso. Este resultado já era esperado, uma vez que a taxa de sucesso avalia a sobreposição entre a *bounding box* estimada e *bounding box* verdadeira do alvo, assim essa métrica seria mais afetada pela remoção do esquema de múltiplos filtros. Similarmente, a taxa de precisão não avalia sobreposição das detecções, apenas a localização do centro do objeto, assim a utilização de um único filtro para fazer o rastreamento não causa um impacto tão grande quanto a remoção do esquema de atualização, que impediria que o modelo aprendesse uma representação incorreta do objeto rastreado. Além disso, observamos que o uso de apenas um dos esquemas não é suficiente para fazer o método obter um resultado tão bom quanto o método combinado com os dois esquemas. Vale notar que o esquema de atualização também está presente no esquema de múltiplos filtros, assim a remoção do esquema de atualização também afeta o modelo de cada um dos filtros. Desta forma, fica demonstrado

mais uma vez que os dois esquemas são complementares, sendo esse uma das razões do método proposto conseguir obter bons resultados tanto na métrica que considera a localização central do objeto quanto na métrica que considera as deformações e variações de tamanho sofridas pelo alvo.

5.5 Avaliação do Esquema de Rastreamento Colaborativo

O método proposto combina vários filtros de correlação baseados em partes com um filtro de correlação global para melhorar o rastreamento de objetos. Os filtros baseados em partes rastreiam partes individuais do objeto, e suas posições são usadas para determinar o centro do objeto rastreado.

O uso de filtros baseados em peças introduz robustez e flexibilidade ao processo de rastreamento, especialmente em situações onde o objeto sofre variações de escala ou mudanças de aparência, uma vez que algumas peças permanecem visíveis e preservam a aparência original. O filtro global é aprendido usando toda a região do objeto, e é usado para definir a janela de pesquisa para cada um dos filtros de correlação baseados em partes na imagem, evitando que os filtros locais percam o rastreamento das partes locais devido a mudanças abruptas no aparências de objetos.

Para avaliar o potencial do esquema colaborativo proposto e demonstrar que ele pode ser integrado a outros métodos de rastreamento baseados em filtros de correlação, decidimos integrar nosso método colaborativo ao conhecido método de rastreamento *kernelized correlation filter* (KCF) (HENRIQUES et al., 2015), que utiliza filtros de correlação com feições HoG multicanais, usando nossa abordagem para melhorar seu desempenho.

Escolhemos o método KCF como *baseline* devido ao fato dele ser um método rápido, confiável e que ajudou a popularizar o uso de métodos baseados em filtros de correlação, servindo como base para diversos outros métodos de filtros de correlação que surgiram nos últimos anos. Através desse experimento podemos avaliar de forma isolada a eficácia do esquema colaborativo proposto, sem a influência do nosso filtro de correlação, e determinar se existe um ganho de desempenho obtido com a abordagem proposta e o quanto ela é capaz de melhorar os resultados de um método baseado em filtros de correlação.

Assim, embora a abordagem proposta de utilizar o esquema colaborativo seja utilizada em conjunto com o nosso método proposto de filtro de correlação, a ideia é de-

Tabela 5.7: Taxa de precisão para a base de dados OTB-2013 utilizando KCF com esquema colaborativo.

Atributos	KCF Mod	KCF	SCM	TGPR	TLD	Struck	DLT
Variação de iluminação	0.712	0.728	0.559	0.644	0.498	0.552	0.479
Variação de escala	0.731	0.679	0.672	0.620	0.606	0.639	0.606
Oclusão	0.762	0.749	0.642	0.680	0.537	0.588	0.517
Deformação	0.781	0.740	0.583	0.700	0.465	0.553	0.481
Desfoque de movimento	0.691	0.650	0.339	0.537	0.518	0.551	0.427
Movimento rápido	0.637	0.602	0.333	0.493	0.551	0.604	0.435
Rotação no plano	0.790	0.725	0.597	0.675	0.584	0.617	0.510
Rotação fora do plano	0.769	0.729	0.618	0.682	0.579	0.616	0.545
Fora de visão	0.687	0.650	0.429	0.505	0.576	0.539	0.505
Plano de fundo confuso	0.736	0.752	0.578	0.717	0.428	0.585	0.440
Baixa resolução	0.714	0.630	0.661	0.438	0.566	0.550	0.554
Geral	**0.794**	**0.741**	**0.649**	**0.705**	**0.608**	**0.656**	**0.548**

monstrar que outros métodos existentes também podem se beneficiar deste esquema, uma vez que o esquema proposto pode ser também incorporado a outros métodos baseados em filtros de correlação. Primeiro, comparamos o método KCF original com uma versão do KCF que modificamos para incluir o esquema colaborativo proposto. Em seguida, modificamos o tipo de kernel e as feições utilizadas em ambos os métodos para avaliar se há também uma diferença de desempenho entre nosso método KCF modificado e a implementação do KCF original.

As tabelas 5.7 e 5.8 mostram as taxas de precisão para os conjuntos de dados OTB-2013 e OTB-2015, com o primeiro e o segundo melhores valores destacados verde e azul, respectivamente. Para efeito de comparação, também são exibidos alguns dos métodos comparativos vistos anteriormente. Conforme observado, nossa abordagem proposta obteve os melhores resultados gerais em ambos os conjuntos de dados, atingindo 5.3% a mais no conjunto de dados OTB-2013 e 2.4% no conjunto de dados OTB-2015 em comparação com o *baseline* (KCF). Considerando os diferentes atributos, o método proposto obteve o melhor resultado na maioria dos casos em relação ao KCF original, principalmente em sequências contendo baixa resolução e rotação no conjunto de dados OTB-2013 e baixa resolução e desfoque de movimento no conjunto de dados OTB-2015.

As tabelas 5.9 e 5.10 mostram a taxa de sucesso para os conjuntos de dados OTB-2013 e OTB-2015. Se considerarmos as métricas de taxa de sucesso, podemos ver que a abordagem proposta também superou o KCF original, alcançando 6.8% a mais no conjunto de dados OTB-2013 e 5.7% no conjunto de dados OTB-2015.

Por meio dessa métrica, podemos reconhecer os benefícios de nossa abordagem,

Tabela 5.8: Taxa de precisão para a base de dados OTB-2015 utilizando KCF com esquema colaborativo.

Atributos	KCF Mod	KCF	SCM	TGPR	TLD	Struck	DLT
Variação de iluminação	0.710	0.719	0.587	0.617	0.547	0.554	0.524
Variação de escala	0.653	0.633	0.576	0.611	0.583	0.609	0.556
Oclusão	0.621	0.629	0.571	0.610	0.540	0.558	0.488
Deformação	0.627	0.617	0.570	0.653	0.495	0.564	0.490
Desfoque de movimento	0.689	0.599	0.268	0.529	0.527	0.575	0.387
Movimento rápido	0.663	0.620	0.330	0.548	0.580	0.631	0.408
Rotação no plano	0.740	0.701	0.543	0.659	0.604	0.626	0.471
Rotação fora do plano	0.682	0.676	0.581	0.655	0.577	0.616	0.534
Fora de visão	0.556	0.498	0.422	0.493	0.456	0.462	0.558
Plano de fundo confuso	0.705	0.712	0.577	0.593	0.456	0.548	0.515
Baixa resolução	0.727	0.665	0.761	0.622	0.627	0.674	0.751
Geral	**0.719**	**0.695**	**0.572**	**0.643**	**0.592**	**0.635**	**0.526**

Tabela 5.9: Taxa de sucesso para a base de dados OTB-2013 utilizando KCF com esquema colaborativo.

Atributos	KCF Mod	KCF	SCM	TGPR	TLD	Struck	DLT
Variação de iluminação	**0.533**	0.493	0.447	0.476	0.379	0.431	0.380
Variação de escala	**0.530**	0.427	0.518	0.418	0.421	0.425	0.458
Oclusão	**0.550**	0.514	0.49	0.485	0.38	0.43	0.401
Deformação	**0.574**	0.534	0.448	0.515	0.34	0.418	0.350
Desfoque de movimento	**0.540**	0.497	0.298	0.434	0.404	0.433	0.329
Movimento rápido	**0.500**	0.459	0.296	0.396	0.417	0.462	0.353
Rotação no plano	**0.579**	0.497	0.458	0.479	0.416	0.444	0.383
Rotação fora do plano	**0.558**	0.495	0.471	0.486	0.405	0.446	0.406
Fora de visão	**0.567**	0.550	0.361	0.442	0.457	0.459	0.409
Plano de fundo confuso	**0.560**	0.535	0.450	0.522	0.345	0.458	0.327
Baixa resolução	0.380	0.272	0.438	0.246	0.295	0.26	0.330
Geral	**0.581**	**0.513**	**0.499**	**0.503**	**0.437**	**0.474**	**0.415**

pois ela considera as variações do objeto, que é uma característica que nossa abordagem pode tratar bem. Analisando os atributos individualmente, nosso método proposto obteve o melhor resultado em quase todos os atributos, exceto para baixa resolução, onde ainda assim foi 10% superior em comparação com KCF original em ambos os conjuntos de dados. A abordagem proposta destaca-se, principalmente, em situações de rotações, desfoque de movimento e variações de escala, onde a utilização de filtros para rastrear partes individuais contribui para uma melhora significativa nos resultados nessas situações.

A tabela 5.11 mostra os resultados comparativos para o método KCF original e para nossa versão modificada do KCF utilizando um kernel linear em vez de um kernel Gaussiano, valores de intensidade dos pixels (gray) como feições em vez de HoG, e ker-

Tabela 5.10: Taxa de sucesso para a base de dados OTB-2015 utilizando KCF com esquema colaborativo.

Atributos	KCF Mod	KCF	SCM	TGPR	TLD	Struck	DLT
Variação de iluminação	0.531	0.479	0.479	0.448	0.411	0.430	0.412
Variação de escala	0.473	0.394	0.439	0.405	0.399	0.406	0.407
Oclusão	0.468	0.443	0.435	0.430	0.370	0.405	0.360
Deformação	0.461	0.455	0.413	0.460	0.344	0.403	0.319
Desfoque de movimento	0.541	0.459	0.200	0.429	0.424	0.456	0.320
Movimento rápido	0.526	0.459	0.299	0.421	0.446	0.469	0.330
Rotação no plano	0.537	0.469	0.408	0.462	0.426	0.447	0.348
Rotação fora do plano	0.497	0.453	0.435	0.456	0.391	0.435	0.387
Fora de visão	0.434	0.393	0.333	0.373	0.338	0.359	0.384
Plano de fundo confuso	0.538	0.498	0.462	0.428	0.352	0.427	0.372
Baixa resolução	0.391	0.290	0.478	0.344	0.346	0.313	0.465
Geral	**0.534**	**0.477**	**0.445**	**0.458**	**0.424**	**0.424**	**0.384**

Tabela 5.11: Taxas de sucesso e precisão para o método KCF original e para a versão modificada do KCF com nosso esquema colaborativo.

	KCF Mod	KCF	KCF Mod Linear	KCF Linear	KCF Mod Gray	KCF Gray	KCF Mod Linear+Gray	KCF Linear+Gray
Dataset	**Taxa de Precisão**							
OTB2013	0.794	0.741	0.778	0.713	0.612	0.559	0.517	0.453
OTB2015	0.719	0.695	0.705	0.682	0.600	0.542	0.490	0.415
Dataset	**Taxa de Sucesso**							
OTB2013	0.581	0.513	0.568	0.506	0.454	0.405	0.392	0.322
OTB2015	0.534	0.477	0.521	0.474	0.450	0.401	0.383	0.313

nel linear e intensidade dos pixels combinados (HENRIQUES et al., 2015), juntamente com a precisão geral e as taxas de sucesso. Como podemos observar, mesmo com o uso de diferentes kernels e feições a versão modificada com o esquema de múltiplos filtros colaborativos continuou obtendo os melhores resultados que a versão sem a modificação.

5.6 Discussão

Como observado nos resultados das análises quantitativa e qualitativa, o método proposto apresentou resultados consistentes em todos os testes. Considerando a taxa de precisão, nos dois conjuntos de dados o método proposto obteve o melhor resultado geral para as sequências de vídeos, assim como a maior média para os onze atributos. Na análise dos atributos individualmente, o método proposto também alcançou o melhor resultado na maioria dos atributos, e, nos casos onde não obteve o melhor resultado, obteve o segundo

melhor resultado. O uso de feições extraídas de Redes Neurais Convolucionais Profundas e o esquema de atualização seletivo dos filtros contribuiu para que o método se destacasse e obtivesse boas taxas de precisão em cenários com mudanças de iluminação, desfoque de movimento, e movimento rápido dos objetos. Além disso, o uso de múltiplos filtros locais contribuiu para que o método também se destacasse em cenários onde ocorrem variações de escala, o objeto está fora de visão, rotações no plano da imagem e rotações fora do plano da imagem.

Considerando a taxa de sucesso, o método proposto também obteve o melhor resultado geral nos dois conjuntos de dados, assim como a maior média considerando os onze atributos. Novamente, o método também obteve o melhor resultado na maioria dos atributos, obtendo o segundo melhor resultado nos atributos onde não foi o melhor. Avaliando as taxas de sucesso, os destaques em relação aos outros métodos foram em situações onde ocorrem desfoque de movimento, movimento rápido dos objetos, variações de escala, rotações no plano e fora do plano da imagem, e semelhança entre o objeto e o plano de fundo.

De maneira geral, o método foi consistente e obteve os melhores resultados em ambos os conjuntos de dados. Além disso, o método proposto se destaca por conseguir atingir melhores resultados tanto na taxa de precisão quanto na taxa de sucesso. Alguns métodos comparativos até se aproximam em alguma das métricas, mas não conseguem obter um bom resultado na outra métrica. O método CFIT, por exemplo, obtém o segundo melhor resultado geral em termos de taxa de sucesso no conjunto de testes OTB-2013 e o terceiro melhor no conjuntos de testes OTB-2015, mas em termos de taxa de precisão obtém apenas o quinto e quarto melhores resultados, respectivamente. Com o método CF2 também ocorre algo semelhante, pois ele consegue obter o segundo melhor resultado no OTB-2013 e o terceiro melhor resultado no OTB-2015 em termos de taxa de precisão, indicando que tem um bom desempenho para se manter no centro do objeto, mas em termos de taxa de sucesso obtém apenas o quarto e o sexto melhores resultados, indicando que não possui um bom desempenho para rastrear o objeto como um todo, levando em consideração as variações de forma e deformações sofridas pelos objetos. Assim, o método proposto consegue lidar melhor com variações de escala e deformações do objeto, se adaptando às mudanças de tamanho e posições do objeto, ao mesmo tempo que consegue se manter centrado no alvo rastreado, indicando com maior precisão a localização do objeto.

Na avaliação do esquema de rastreamento colaborativo, os experimentos mostram

que a abordagem proposta é consistente e alcança melhores resultados do que método KCF, utilizado como *baseline* nos conjuntos de dados e métricas usadas. Considerando a taxa de precisão, em ambos os conjuntos de dados, a abordagem proposta obteve o melhor resultado geral para as sequências de vídeo, bem como o melhor resultado na maioria dos atributos em relação ao KCF original, mostrando que a abordagem pode ser utilizada para melhorar a localização dos objetos. Considerando a taxa de sucesso, a abordagem proposta também obteve o melhor resultado geral e os melhores resultados na maioria dos atributos. A taxa de sucesso ajuda a mostrar mais claramente os pontos fortes da abordagem proposta, indicando que ela pode se adaptar às variações na forma e outras transformações sofridas pelo objeto rastreado.

Além disso, nos experimentos utilizando usando um kernel linear e valores de níveis de cinza como feições nossa modificação proposta também obteve um resultado melhor do que o KCF original. Os resultados mostram que a integração do esquema colaborativo proposto a um método baseado em filtro de correlação leva a uma melhora geral no desempenho, independentemente das feições e kernels utilizados.

Apesar de apresentar bons resultados, algumas melhorias podem ser feitas para aprimorar o método proposto. Como sugestão para futuras melhorias no método proposto podemos citar a utilização de uma rede neural diferente para a obtenção de feições. Diversas CNNs foram propostas nos últimos anos, e algumas delas podem conseguir extrair feições mais discriminativas, melhorando a performance do método. Aumentar a resolução de entrada da rede, seja usando a VGG-19 ou outra CNN, também pode contribuir para aprimorar os resultados do rastreamento, uma vez que a rede vai ser capaz de extrair mais detalhes presentes na cena. Outro aprimoramento que podemos sugerir é usar a informação de movimento/velocidade dos objetos entre cada quadro. Dessa forma, poderíamos tratar oclusões totais que ocorrem por longos períodos de tempo. Com base na informação de direção e velocidade do objeto na cena, mesmo que ele sofra oclusão durante muitos quadros, seria possível estimar com precisão a posição futura desse objeto e evitar que o rastreamento seja perdido, desde que o objeto mantenha uma velocidade e direção constante.

Um fator que também pode ser explorado para melhorar o desempenho do método proposto é o tamanho da região de busca usada para cada filtro. Assim, aumentar a região de busca pode ajudar a rastrear objetos que estejam se movendo de forma muito rápida pela cena. Outra alternativa que pode ser usada para aprimorar o método é usar mais filtros para cada parte do objeto. No método proposto o objeto é dividido em quatro partes, mas

dependendo da aplicação pode ser interessante dividir o objeto em um número maior de partes. Dessa forma, objetos não rígidos ou que sofram oclusões em diversas partes de simultaneamente podem se beneficiar de um número maior de partes sendo rastreadas.

Apesar do método proposto ter atingido uma boa performance, nossa implementação atual roda apenas a cerca de 1 *FPS* em CPU, utilizando o código em Matab não otimizado e *single thread*. Desta forma, o método apresenta limitações dependendo do dispositivo, não podendo ser usada para processamento em tempo real em dispositivos embarcados ou drones, por exemplo, que possuem poder de processamento limitados. Note que o *framework* proposto pode ser facilmente estendido para um implementação paralela e otimizada, e também utilizar processamento em GPU para aumentar o desempenho. A extração de feições é uma das etapas que mais consome processamento, e é necessário realizar a extração para cada uma das partes do objeto. Assim, o uso de uma GPU faz com que o método obtenha um melhor desempenho, uma vez que esse é um processo que pode tirar vantagem do grande poder de paralelização das GPUs.

6 CONCLUSÃO

Neste trabalho, apresentamos uma nova estratégia para rastreamento de objetos em sequências de vídeos. Propusemos uma abordagem baseada em filtros de correlação, onde o uso de um filtro global é combinado também com filtros locais, responsáveis por rastrear partes individuais dos objetos.

Os filtros de correlação empregam feições extraídas de múltiplas camadas de redes neurais convolucionais profundas, possibilitando que o método proposto consiga identificar tanto detalhes de mais alto nível e a semântica dos objetos quanto detalhes de níveis mais baixos que representam detalhes mais finos. Além disso, são propostos um esquema colaborativo, combinando múltiplos filtros globais e locais, e também uma estratégia para identificar se o resultado do rastreamento é confiável, ajudando a decidir quando os filtros devem ser atualizados.

Como observado nos resultados experimentais, o método proposto foi capaz de obter bons resultados nos testes e comparações realizadas. O método obteve o melhor desempenho no geral, e foi bem em ambas as métricas de rastreamento, indicando que ele funciona bem tanto para determinar com precisão o local do objeto quanto para detectar corretamente a área do objeto, sendo capaz de se adaptar às variações de tamanho e deformações sofridas pelo alvo rastreado ao longo do vídeo.

O esquema proposto também foi integrado ao método KCF tradicional, que foi usado como *baseline* para avaliar o esquema de rastreamento colaborativo proposto. Conforme observado, o KCF modificado com nossa abordagem colaborativa também alcançou a melhor precisão geral e taxas de sucesso para ambos os conjuntos de dados em comparação com a versão original do KCF. Desta forma, verificamos a robustez do esquema proposto e também que ele pode ser facilmente integrada a outros métodos baseados em filtro de correlação existentes, e potencialmente fazendo com que eles obtenham um desempenho ainda melhor.

6.1 Trabalhos Futuros

O método proposto demonstrou lidar bem com situações adversas, tendo assim grande potencial para aplicações envolvendo rastreamento nos mais diversos cenários, como vigilância por vídeo e controle de tráfego de veículos. Futuramente, planejamos utilizar a abordagem utilizada nesse trabalho para criar um método para rastreamento de

múltiplos objetos (MOT *multiple object tracking*. Como uma parte crítica do rastreamento envolve a correta inicialização da *bounding box* do objeto a ser rastreado, será necessário utilizar um método de detecção de objetos como etapa anterior ao rastreamento. Como trabalho futuro, pretendemos fazer uso do método de rastreamento proposto em uma aplicação específica. Para explorar o potencial do método, será realizado a aplicação do método proposto em um *framework* de detecção de veículos para tarefas de rastreamento e contagem de veículos em sequências de vídeo de tráfego. Será utilizado uma CNN para fazer a detecção dos veículos na cena e, em seguida, nosso método baseado em múltiplos filtros de correlação será utilizado para fazer o rastreamento dos veículos. Além disso, planejamos submeter um artigo descrevendo os resultados deste *framework* de detecção de veículos para o periódico IEEE Instrumentation & Measurement Magazine.

6.2 Publicações

Durante o desenvolvimento deste trabalho, foram elaborados artigos científicos para publicação em periódicos da área relatando os resultados do trabalho de pesquisa realizado durante o período do doutorado. A seguir, são listadas cada uma dessas publicações.

Artigos publicados:

- Pablo Barcellos, Jacob Scharcanski. "Object tracking scheme using part-based correlation filters". 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). p. 1–6, 2020.
- Pablo Barcellos, Vitor Gomes, Jacob Scharcanski. "Shadow detection in camera-based vehicle detection: survey and analysis". JOURNAL OF ELECTRONIC IMAGING. v.25, p.051205, 2016.
- Pablo Barcellos, Christiano Bouvié, Fabiano Lopes Escouto, Jacob Scharcanski. "A novel video based system for detecting and counting vehicles at user-defined virtual loops". EXPERT SYSTEMS WITH APPLICATIONS.v.42, p.1845 - 1856, 2015.
- Vitor Gomes, Pablo Barcellos, Jacob Scharcanski. "Stochastic shadow detection using a hypergraph partitioning approach". PATTERN RECOGNITION. v.63, p.30 - 44, 2017.

- Vitor Gomes, Pablo Barcellos, Jacob Scharcanski. "Image-based approach for detecting vehicles in user-defined virtual inductive loops". JOURNAL OF ELECTRONIC IMAGING. v.27, p.1, 2018.

Artigos submetidos:

- Pablo Barcellos, Jacob Scharcanski. "Part-based Object Tracking Using Multiple Adaptive Correlation Filters". Trabalho submetido ao periódico IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Artigos a serem submetidos:

- Pablo Barcellos, Jacob Scharcanski. "Vehicle Counting Based on Multiple Correlation Filters". Trabalho a ser submetido ao IEEE Instrumentation & Measurement Magazine.

REFERÊNCIAS

ADAM, A.; RIVLIN, E.; SHIMSHONI, I. Robust fragments-based tracking using the integral histogram. In: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2006. ISBN 0769525970. ISSN 10636919.

AKIN, O. et al. Deformable part-based tracking by coupled global and local correlation filters. **Journal of Visual Communication and Image Representation**, v. 38, p. 763–774, jul 2016. ISSN 10473203. Available from Internet: <http://dx.doi.org/10.1016/j.jvcir.2016.04.018https://linkinghub.elsevier.com/retrieve/pii/S1047320316300517>.

BABENKO, B.; Ming-Hsuan Yang; BELONGIE, S. Visual tracking with online Multiple Instance Learning. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2009. p. 983–990. ISBN 978-1-4244-3992-8. ISSN 1939-3539. Available from Internet: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206737>.

BARCELLOS, P. et al. A novel video based system for detecting and counting vehicles at user-defined virtual loops. **Expert Systems with Applications**, v. 42, n. 4, p. 1845–1856, 2015. ISSN 09574174.

BARCELLOS, P.; GOMES, V.; SCHARCANSKI, J. Shadow detection in camera-based vehicle detection: survey and analysis. **Journal of Electronic Imaging**, v. 25, n. 5, p. 051205, 2016. ISSN 1017-9909.

BARCELLOS, P.; SCHARCANSKI, J. Object tracking scheme using part-based correlation filters. In: **2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)**. IEEE, 2020. p. 1–6. ISBN 978-1-7281-4460-3. Available from Internet: <https://ieeexplore.ieee.org/document/9128923/>.

BERTINETTO, L. et al. Staple: Complementary Learners for Real-Time Tracking. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 1401–1409, 2016.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006, 2006. 738 p. ISBN 0387310738, 9780387310732.

BOLME, D. et al. Visual object tracking using adaptive correlation filters. In: **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. IEEE, 2010. p. 2544–2550. ISBN 978-1-4244-6984-0. ISSN 10636919. Available from Internet: <http://ieeexplore.ieee.org/document/5539960/>.

BOUVIE, C. et al. Tracking and counting vehicles in traffic video sequences using particle filtering. In: **Conference Record - IEEE Instrumentation and Measurement Technology Conference**. [S.l.: s.n.], 2013. p. 812–815. ISBN 9781467346221. ISSN 10915281.

CHEN, Z.; HONG, Z.; TAO, D. An Experimental Survey on Correlation Filter-based Tracking. 2015. Available from Internet: <https://arxiv.org/pdf/1509.05520.pdf>.

CHEN, Z.; HONG, Z.; TAO, D. **An Experimental Survey on Correlation Filter-based Tracking**. 2015. Available from Internet: <https://arxiv.org/abs/1509.05520>.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005**. [s.n.], 2005. v. 1, p. 886–893. Available from Internet: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1467360>.

DANELLJAN, M. et al. Convolutional Features for Correlation Filter Based Visual Tracking. In: **2015 IEEE International Conference on Computer Vision Workshop (ICCVW)**. IEEE, 2015. p. 621–629. ISBN 978-1-4673-9711-7. Available from Internet: <http://ieeexplore.ieee.org/document/7406433/>.

DANELLJAN, M. et al. Learning Spatially Regularized Correlation Filters for Visual Tracking. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. IEEE, 2015. p. 4310–4318. ISBN 978-1-4673-8391-2. Available from Internet: <http://ieeexplore.ieee.org/document/7410847/>.

DANELLJAN, M. et al. Accurate Scale Estimation for Robust Visual Tracking. In: **Proceedings of the British Machine Vision Conference 2014**. British Machine Vision Association, 2014. p. 65.1–65.11. ISBN 1-901725-52-9. Available from Internet: <http://www.bmva.org/bmvc/2014/files/paper038.pdfhttp://www.bmva.org/bmvc/2014/papers/paper038/index.html>.

DANELLJAN, M. et al. Adaptive Color Attributes for Real-Time Visual Tracking. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2014. p. 1090–1097. ISBN 978-1-4799-5118-5. ISSN 10636919. Available from Internet: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909539>.

FENG, W. et al. Dynamic Saliency-Aware Regularization for Correlation Filter-Based Object Tracking. **IEEE Transactions on Image Processing**, IEEE, v. 28, n. 7, p. 3232–3245, 2019. ISSN 19410042.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. [S.l.]: Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981.

FU, H. et al. Learning reliable-spatial and spatial-variation regularization correlation filters for visual tracking. **Image and Vision Computing**, v. 94, p. 103869, feb 2020. ISSN 02628856. Available from Internet: <https://linkinghub.elsevier.com/retrieve/pii/S0262885620300019>.

GAO, J. et al. Transfer Learning Based Visual Tracking with Gaussian Processes Regression. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [s.n.], 2014. p. 188–203. ISBN 9783319105772. Available from Internet: <http://link.springer.com/10.1007/978-3-319-10578-9_13>.

GOLCHIN, M. et al. Shadow Detection Using Color and Edge Information. **Journal of Computer Science**, v. 9, n. 11, p. 1575–1588, nov 2013. ISSN 1549-3636. Available from Internet: <http://thescipub.com/abstract/10.3844/jcssp.2013.1575.1588>.

GOMES, V.; BARCELLOS, P.; SCHARCANSKI, J. Stochastic shadow detection using a hypergraph partitioning approach. **Pattern Recognition**, v. 63, p. 30–44, 2017. ISSN 00313203.

GOMES, V.; BARCELLOS, P.; SCHARCANSKI, J. Image-based approach for detecting vehicles in user-defined virtual inductive loops. **Journal of Electronic Imaging**, v. 27, n. 03, p. 1, jun 2018. ISSN 1017-9909. Available from Internet: <https://www.spiedigitallibrary.org/journals/journal-of-electronic-imaging/volume-27/issue-03/033026/Image-based-approach-for-detecting-vehicles-in-user-defined-virtual/10.1117/1.JEI.27.3.033026.full>.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (2nd Edition)**. Prentice Hall, 2002. ISBN 0201180758. Available from Internet: <http://www.amazon.com/Digital-Image-Processing-2nd-Edition/dp/0201180758?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0201180758>.

HARE, S. et al. Struck: Structured Output Tracking with Kernels. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2016. ISSN 01628828.

HENRIQUES, J. F. et al. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In: FITZGIBBON, A. et al. (Ed.). **Computer Vision – ECCV 2012**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 702–715. ISBN 978-3-642-33765-9. Available from Internet: <http://link.springer.com/10.1007/978-3-642-33765-9_50>.

HENRIQUES, J. F. et al. High-Speed Tracking with Kernelized Correlation Filters. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 37, n. 3, p. 583–596, mar 2015. ISSN 0162-8828. Available from Internet: <http://ieeexplore.ieee.org/document/6870486/>.

HU, Q. et al. Object Tracking Using Multiple Features and Adaptive Model Updating. **IEEE Transactions on Instrumentation and Measurement**, v. 66, n. 11, p. 2882–2897, nov 2017. ISSN 0018-9456. Available from Internet: <http://ieeexplore.ieee.org/document/8012414/>.

HUANG, D. et al. Applying Detection Proposals to Visual Tracking for Scale and Aspect Ratio Adaptability. **International Journal of Computer Vision**, 2017. ISSN 15731405.

HUANG, J.-B.; CHEN, C.-S. Moving cast shadow detection using physics-based features. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2009. p. 2310–2317. ISBN 978-1-4244-3992-8. Available from Internet: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5206629>.

Jia Deng et al. ImageNet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2009. p. 248–255. ISBN 978-1-4244-3992-8. ISSN 1063-6919. Available from Internet: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848>.

KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-Learning-Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 7, p. 1409–1422, jul 2012. ISSN 0162-8828. Available from Internet: <http://ieeexplore.ieee.org/document/6104061/>.

KRIZHEVSKY, A.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Neural Information Processing Systems**, 2012. ISSN 10495258.

LI, Y.; ZHU, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In: **Computer Vision - ECCV 2014 Workshops**. [s.n.], 2014. p. 254–265. Available from Internet: <http://link.springer.com/10.1007/978-3-319-16181-5_18>.

LI, Y.; ZHU, J.; HOI, S. C. Reliable Patch Trackers: Robust visual tracking by exploiting reliable patches. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. IEEE, 2015. p. 353–361. ISBN 978-1-4673-6964-0. ISSN 10636919. Available from Internet: <http://ieeexplore.ieee.org/document/7298632/>.

LI, Y. et al. **Robust estimation of similarity transformation for visual object tracking**. 2017. Available from Internet: <https://arxiv.org/pdf/1712.05231.pdf>.

LIANG, H. G. et al. Improved target tracking algorithm based on kernelized correlation filter. **Journal of Electronic Imaging**, v. 28, n. 02, p. 1, 2019. ISSN 1560-229X.

LIU, H. et al. Robust Long-Term Tracking via Instance-Specific Proposals. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 69, n. 4, p. 950–962, apr 2020. ISSN 0018-9456. Available from Internet: <https://ieeexplore.ieee.org/document/8782450/https://ieeexplore.ieee.org/document/8708248/>.

LIU, H.; LI, S.; FANG, L. Robust Object Tracking Based on Principal Component Analysis and Local Sparse Representation. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 64, n. 11, p. 2863–2875, nov 2015. ISSN 0018-9456. Available from Internet: <http://ieeexplore.ieee.org/document/7120989/>.

LIU, T.; WANG, G.; YANG, Q. Real-time part-based visual tracking via adaptive correlation filters. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. IEEE, 2015. p. 4902–4912. ISBN 978-1-4673-6964-0. ISSN 10636919. Available from Internet: <http://ieeexplore.ieee.org/document/7299124/>.

MA, C. et al. Hierarchical Convolutional Features for Visual Tracking. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. IEEE, 2015. p. 3074–3082. ISBN 978-1-4673-8391-2. Available from Internet: <http://ieeexplore.ieee.org/document/7410709/>.

MA, J. et al. Robust Scale Adaptive Tracking by Combining Correlation Filters with Sequential Monte Carlo. **Sensors**, v. 17, n. 3, p. 512, mar 2017. ISSN 1424-8220. Available from Internet: <www.mdpi.com/journal/sensorshttp://www.mdpi.com/1424-8220/17/3/512>.

MAHALANOBIS, A.; Vijaya Kumar, B. V. K.; CASASENT, D. Minimum average correlation energy filters. **Applied Optics**, 1987. ISSN 0003-6935.

MAHALANOBIS, A. et al. Unconstrained correlation filters. **Applied Optics**, 1994. ISSN 0003-6935.

NG, A. Y.; JORDAN, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. **Proceedings of Advances in Neural Information Processing**, 2002. ISSN 13704621.

NIELSEN, M. A. **Neural Networks and Deep Learning**. [S.l.]: Determination Press, 2015.

PATEL, S.; PINGEL, J. **Introduction to Deep Learning: What Are Convolutional Neural Networks?** 2017. Available from Internet: <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>.

RIFKIN, R.; YEO, G.; POGGIO, T. Regularized Least-Squares Classification. **Nato Science Series Sub Series III Computer and Systems Sciences**, 2003. ISSN 1387-6694.

ROSS, D. A. et al. Incremental Learning for Robust Visual Tracking. **International Journal of Computer Vision**, v. 77, p. 125–141, 2007. ISSN 09205691. Available from Internet: <http://www.springerlink.com/index/10.1007/s11263-007-0075-7>.

SANIN, A.; SANDERSON, C.; LOVELL, B. Shadow detection: A survey and comparative evaluation of recent methods. **Pattern recognition**, 2012. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S0031320311004043>.

SCHARCANSKI, J. et al. A particle-filtering approach for vehicular tracking adaptive to occlusions. **Vehicular Technology, IEEE Transactions on**, IEEE, v. 60, n. 2, p. 381–389, 2011. Available from Internet: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5669358>.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. sep 2014. Available from Internet: <https://arxiv.org/abs/1409.1556http://arxiv.org/abs/1409.1556>.

SMEULDERS, A. W. et al. Visual Tracking: An Experimental Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 36, n. 7, p. 1442–1468, jul 2014. ISSN 0162-8828. Available from Internet: <http://ieeexplore.ieee.org/document/6671560/>.

SUI, Y.; WANG, G.; ZHANG, L. **Correlation Filter Learning Toward Peak Strength for Visual Tracking**. 2017. Available from Internet: <https://scholar.harvard.edu/files/suiyao/files/sui_pscf_tcyb2017.pdf>.

SUN, S. et al. Robust Visual Detection and Tracking Strategies for Autonomous Aerial Refueling of UAVs. **IEEE Transactions on Instrumentation and Measurement**, v. 68, n. 12, p. 4640–4652, dec 2019. ISSN 0018-9456. Available from Internet: <https://ieeexplore.ieee.org/document/8667320/>.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: . [S.l.: s.n.], 2001. p. 511–518.

WANG, N.; YEUNG, D.-Y. Learning a Deep Compact Image Representation for Visual Tracking. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems 26 (NIPS 2013)**. Curran Associates, Inc., 2013. p. 2544–2550. ISBN 978-1-4673-1228-8. Available from Internet: <http://papers.nips.cc/paper/5192-learning-a-deep-compact-image-representation-for-visual-tracking.pdfhttps://papers.nips.cc/paper/5192-learning-a-deep-compact-image-representation-for-visual-tracking>.

Wei Zhong; Huchuan Lu; Ming-Hsuan Yang. Robust Object Tracking via Sparse Collaborative Appearance Model. **IEEE Transactions on Image Processing**, v. 23, n. 5, p. 2356–2368, may 2014. ISSN 1057-7149. Available from Internet: <http://ieeexplore.ieee.org/document/6777566/>.

WEIJER, J. van de et al. Learning color names for real-world applications. **IEEE Transactions on Image Processing**, 2009. ISSN 10577149.

WU, Y.; LIM, J.; YANG, M.-H. Online Object Tracking: A Benchmark. In: **2013 IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2013. p. 2411–2418. ISBN 978-0-7695-4989-7. ISSN 10636919. Available from Internet: <http://ieeexplore.ieee.org/document/6619156/>.

WU, Y.; LIM, J.; YANG, M.-H. Object Tracking Benchmark. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 37, n. 9, p. 1834–1848, sep 2015. ISSN 0162-8828. Available from Internet: <http://ieeexplore.ieee.org/document/7001050/>.

YANG, Y. et al. Parallel Correlation Filters for Real-Time Visual Tracking. **Sensors**, v. 19, n. 10, p. 2362, may 2019. ISSN 1424-8220. Available from Internet: <https://www.mdpi.com/1424-8220/19/10/2362>.

YILMAZ, A.; JAVED, O.; SHAH, M. Object tracking: A survey. **ACM Computing Surveys**, v. 38, p. 13, 2006. ISSN 03600300. Available from Internet: <http://portal.acm.org/citation.cfm?doid=1177352.1177355>.

YUAN, D. et al. Visual object tracking with adaptive structural convolutional network. **Knowledge-Based Systems**, v. 194, p. 105554, apr 2020. ISSN 09507051. Available from Internet: <https://linkinghub.elsevier.com/retrieve/pii/S0950705120300472>.

ZHANG, J.; MA, S.; SCLAROFF, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [s.n.], 2014. p. 188–203. ISBN 9783319105987. Available from Internet: <http://link.springer.com/10.1007/978-3-319-10599-4_13>.

ZHANG, M. et al. Joint Scale-Spatial Correlation Tracking with Adaptive Rotation Estimation. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2016. ISBN 9781467383905. ISSN 15505499.

ZHANG, T.; BIBI, A.; GHANEM, B. In Defense of Sparse Tracking: Circulant Sparse Tracker. In: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2016. ISBN 9781467388504. ISSN 10636919.

ZHENG, Y.; YANG, C.; MERKULOV, A. Breast cancer screening using convolutional neural network and follow-up digital mammography. In: ASHOK, A. et al. (Ed.). **Computational Imaging III**. SPIE, 2018. p. 4. ISBN 9781510618497. Available from Internet: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10669/2304564/Breast-cancer-screening-using-convolutional-neural-network-and-follow-up/10.1117/12.2304564.full>.

ZITNICK, C. L.; DOLLÁR, P. Edge boxes: Locating object proposals from edges. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.: s.n.], 2014. ISSN 16113349.