

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Combinação de rastreadores utilizando filtro de Kalman

Pedro Senna de Campos

Itajubá, junho de 2017

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Pedro Senna de Campos

Combinação de rastreadores utilizando filtro de Kalman

Dissertação submetida ao Programa de Pós-Graduação em
Ciência e Tecnologia da Computação como parte dos requisitos
para obtenção do Título de Mestre em Ciência e Tecnologia
da Computação

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Guilherme Sousa Bastos

Coorientador: Prof. Dr. Isabela Neves Drummond

junho de 2017
Itajubá - MG

Agradecimentos

Agradeço primeiramente aos meus pais e familiares pelo apoio e confiança.

Aos Professores Guilherme Sousa Bastos e Isabela Neves Drummond pelas orientações e paciência durante o desenvolvimento deste trabalho.

À CAPES pelo bolsa de estudos que possibilitou a execução deste trabalho e pelo apoio do PROAP para participação no SIBGRAPI.

Aos colegas do LRO, pela companhia e apoio, em especial ao Lucas, Audeliano, Alyson e Adriano.

E a todos os pesquisadores e desenvolvedores que disponibilizaram seus códigos e colaboraram de forma importante para desenvolvimento deste trabalho.

*“ Nada como procurar quando se quer achar alguma coisa.
Quando se procura geralmente se encontra alguma coisa,
sem dúvida, mas nem sempre o que estávamos procurando.
(Thorin Escudo de Carvalho)*

Resumo

O rastreamento visual de objetos é parte vital em sistemas que realizam a percepção visual do ambiente, sendo fundamental para que as máquinas sejam capazes de ver, entender e reagir ao ambiente. O seu principal objetivo é determinar a localização de objetos de interesse em cada quadro de um vídeo de forma precisa e robusta, permitindo que aplicações de alto nível utilizem este conhecimento em suas operações. Neste trabalho é apresentado um método de rastreamento de objetos baseado em combinações de rastreadores chamado KFebT. Este método é capaz de fundir o resultado de diversos métodos de rastreamento ou métodos especialistas utilizando um filtro de Kalman. O método de fusão proposto é simples e não precisa de treinamento prévio, necessitando apenas do resultado dos rastreadores e de uma medida de confiança nesses resultados. Foram realizados testes na base de dados do *Visual Object Tracking Challenge* (VOT) 2015. O KFebT foi testado com combinações de dois ou três métodos de rastreamento e os resultados demonstram uma clara melhoria nos resultados comparando com o dos rastreadores utilizados em sua composição.

Palavras-chaves: Rastreamento visual de objetos; combinação de rastreadores; Filtro de Kalman.

Abstract

Visual object tracking is a vital part of systems that performs the visual perception of the environment and is critical for the machines to be able to see, understand and react to the environment. Its main objective is to find the location of objects of interest in each frame of a video with accuracy and robustness, allowing that high-level applications use this knowledge in their operations. In this work we present a object tracking method based on combinations of trackers called KFebT. This method is able to fuse the result of several tracking methods or expert methods using a Kalman filter. The proposed fusion method is simple and does not require previous training, requiring only the result of the trackers and a measure of confidence in these results. Tests were performed on the Visual Object Tracking Challenge (VOT) 2015 database. KFebT was tested with combinations of two or three tracking methods and the results demonstrate a clear improvement in the results when compared to the trackers used on its composition.

Key-words: visual object tracking; trackers ensemble; Kalman Filter.

Lista de ilustrações

Figura 1 – Exemplo de formas de representação de estado.	19
Figura 2 – Imprecisões causadas pela utilização de retângulos para representar o estado do objeto de interesse.	19
Figura 3 – Exemplo de semelhança com o plano de fundo. O objeto de interesse é a folha destacada pelo retângulo vermelho, sendo que existem diversas folhas semelhantes a ela na imagem. A imagem faz parte da base de dados VOT2015.	22
Figura 4 – Exemplo de mudança de iluminação causada pela mudança na direção da câmera em relação a uma fonte de luz. As imagens fazem parte da base de dados VOT2015.	22
Figura 5 – Exemplo mudança na aparência do objeto de interesse. As imagens fazem parte da base de dados VOT2015.	23
Figura 6 – Exemplo de oclusão. As imagens fazem parte da base de dados VOT2015.	23
Figura 7 – Exemplo de mudança de escala, o objeto de interesse é o pássaro destacado pelo retângulo vermelho. As imagens fazem parte da base de dados VOT2015.	23
Figura 8 – Exemplo de borrão devido a movimentação rápida da câmera. As imagens fazem parte da base de dados VOT2015.	24
Figura 9 – Fluxograma do KCF.	26
Figura 10 – Iniciação do CMT.	27
Figura 11 – Processamento de um novo quadro realizado pelo CMT para encontrar a localização do objeto de interesse.	29
Figura 12 – Ilustração do processo de fusão utilizando dois rastreadores, o rastreador 1 que estima a translação e a mudança de escala do objeto de interesse e o rastreador 2 que estima apenas a translação. No caso A, os rastreadores reportam apenas as variáveis que estimam, fazendo com que a estimativa ruim do rastreador 1 afete a escala do resultado final. No caso B, o rastreador 2 reporta a sua escala atual, evitando que o resultado ruim do rastreador 2 possua muita influência sobre o resultado final.	36
Figura 13 – Diagrama de classes do sistema implementado.	39
Figura 14 – Forma de anotações utilizadas no desafio VOT. No ano de 2013 foi utilizado um retângulo alinhado com os eixos da imagem. No ano de 2014 em diante foi utilizado um retângulo rotacionado como anotação.	48

Figura 15	– Exemplo de como é realizado o cálculo da sobreposição, onde o retângulo vermelho é a anotação do objeto de interesse e o retângulo verde é o resultado de um rastreador. A sobreposição é calculada dividindo a área verde pela soma das áreas vermelhas e verde.	49
Figura 16	– Exemplo do funcionamento do sistema de detecção de falhas do VOT. Toda vez que a sobreposição entre a região reportada pelo rastreador e a região anotada for igual ou menor que um certo limiar, o rastreador é reiniciado e uma falha é anotada.	50
Figura 17	– Obtenção de sequências virtuais utilizando os resultados dos rastreadores.	51
Figura 18	– A curva de sobreposição é obtida calculando a média da sobreposição das sequências virtuais. A média dos valores da curva de sobreposição entre os valores 108 e 371 é a sobreposição esperada.	51
Figura 19	– Exemplo de imagens contidas na base de dados VOT2015. Os retângulos cinza são as anotações dessas imagens. São apresentadas imagens das sequências <i>basketball</i> , <i>fernando</i> e <i>gymnastics2</i>	52
Figura 20	– Interações entre o <i>toolkit</i> do VOT e os rastreadores.	55
Figura 21	– Testes de parâmetros da combinação KFebT-AK. O primeiro gráfico mostra os resultados da variação do ajuste α do KCF. O segundo gráfico mostra os resultados da variação do ajuste α do ASMS e último gráfico mostra o resultado da variação da incerteza do processo do FK.	59
Figura 22	– Testes de parâmetros da combinação KFebT-AKV. O primeiro gráfico mostra os resultados da variação do ajuste α do VDP. O segundo gráfico mostra os resultados da variação do ajuste α do KCF. O terceiro gráfico mostra os resultados da variação do ajuste α do ASMS e o último gráfico mostra o resultado da variação da incerteza do processo do FK.	60
Figura 23	– Testes de parâmetros da combinação KFebT-ACK. O primeiro gráfico mostra os resultados da variação do ajuste α do CBT. O segundo gráfico mostra os resultados da variação do ajuste α do KCF. O terceiro gráfico mostra os resultados da variação do ajuste α do ASMS e o último gráfico mostra o resultado da variação da incerteza do processo do FK.	61
Figura 24	– Testes do ajuste da penalidade de movimento(β). Foi utilizado o mesmo ajuste de penalidade para todos os rastreadores utilizados nas combinações.	61
Figura 25	– Gráfico de sobreposição esperada dos rastreadores na base do VOT2015.	64
Figura 26	– Resultado dos rastreadores por atributo visual na base do VOT2015.	65

Figura 27 – Comparação dos resultado dos rastreadores na sequência <i>girl</i> . O retângulo preto representa a anotação do objeto. Na legenda é apresentada a sobreposição do rastreador no quadro atual, seguido da quantidade total de falhas e a sobreposição média até o quadro atual. Os vídeos comparativos dos resultados das 60 sequências do VOT2015 podem ser encontrados em www.youtube.com/playlist?list=PLuCzbviUtv_Gn54-U8H-JqDICLALm-IZ1	65
Figura 28 – Gráficos da sobreposição do resultado com a anotação dos rastreadores KFebT-AK, ASMS e KCF para cada quadro das sequências <i>basketball</i> (esquerda) e <i>road</i> (direita) no teste sem supervisão.	69
Figura 29 – Resultado qualitativo do teste sem supervisão. São apresentados os quadros 1, 100, 200 e 300 das sequências <i>basketball</i> , <i>girl</i> , <i>road</i> , <i>singer1</i> , <i>wiper</i> e <i>car1</i> . No quadro 1 é apresentada apenas a anotação do objeto de interesse. Nos outros quadros são apresentados os resultados do KFebT-AK (vermelho), do ASMS (amarelo), do KCF(verde) e a anotação do objeto de interesse (cinza).	70

Lista de tabelas

Tabela 1 – Comparação entre o resultado do CMT e do CBT.	45
Tabela 2 – Comparação entre os rastreadores utilizados	46
Tabela 3 – Teste das combinações de rastreadores, onde A representa o ASMS, C o CBT, K o KCF, e V o VDP.	57
Tabela 4 – Resultado dos rastreadores utilizados como base.	58
Tabela 5 – Melhores parâmetros encontrados para cada combinação de rastreadores.	62
Tabela 6 – Teste da utilização da retroalimentação nas combinações ACK e AK. .	62
Tabela 7 – Teste adaptação. São apresentados os resultados das combinações quando o KCF utiliza o próprio resultado ou quando utiliza o resultado da fusão para realizar a atualização do seu modelo.	63
Tabela 8 – Comparação do método proposto com os rastreadores utilizados como base.	64
Tabela 9 – Rastreadores submetidos no VOT2015 utilizados na comparação. . . .	66
Tabela 10 – Comparação com os rastreadores submetidos no VOT2015.	67
Tabela 11 – Resultados do teste com ruído.	68
Tabela 12 – Resultados do teste sem supervisão.	69

Lista de abreviaturas e siglas

ASEF	<i>Average of Synthetic Exact Filters</i>
ASMS	<i>Adaptive scale mean-shift</i>
CBT	<i>Clustering of Static-adaptive Correspondences for Deformable Object Tracking</i>
CMT	<i>Consensus-based Matching and Tracking</i>
CSK	<i>Structure of tracking-by-detection with Kernels</i>
EFO	<i>Equivalent Filter Operations</i>
HGO	Histograma de Gradientes Orientados
KCF	<i>Kernelized Correlation Filter</i>
KFebT	<i>Kalman Filter ensemble-based Tracker</i>
MOSSE	<i>Minimum Output Sum of Squared Error</i>
MS	<i>Mean-Shift</i>
QPS	Quadros Por Segundo
UMACE	<i>Unconstrained Minimum Average Correlation Energy</i>
VDP	Variação da Distância entre os Pares de pontos
VOT	<i>Visual Object Tracking</i>

Índice de algoritmos

1	<i>KFebT - Kalman filter ensemble-based tracker</i>	38
2	<i>VDP - Variação de distância entre pontos</i>	43
3	<i>CBT - Consensus based tracker</i>	45

Sumário

1	Introdução	15
1.1	Motivação	15
1.2	Objetivos	17
1.3	Contribuições	17
1.4	Organização	17
2	Rastreamento Visual de Objetos	18
2.1	Rastreamento de objetos sem modelo prévio	20
2.2	Desafios	21
2.3	Métodos de rastreamento	24
2.3.1	<i>Mean-shift</i>	24
2.3.2	Filtros de correlação	25
2.3.3	CMT	27
3	Combinação de Rastreadores	30
3.1	Trabalhos relacionados	30
3.2	Combinação de rastreadores utilizando o Filtro de Kalman	31
3.2.1	Filtro de Kalman	32
3.2.2	Modelagem do Filtro de Kalman	34
3.2.3	Método proposto	37
3.2.4	Implementação	38
3.3	Rastreadores utilizados	41
3.3.1	ASMS	41
3.3.2	KCF	42
3.3.3	VDP	43
3.3.4	CBT	44
3.3.5	Comparação entre os rastreadores	46
4	Desafio VOT	47
4.1	História	47
4.2	Métricas de avaliação de rastreadores	49
4.3	Bases de dados	52
4.4	Metodologia de teste	53
4.4.1	Teste padrão	53
4.4.2	Teste com ruído	53
4.4.3	Teste sem supervisão	54

4.5	VOT <i>toolkit</i>	54
5	Resultados Experimentais	56
5.1	Ambiente de teste	56
5.2	Testes no modelo	57
5.2.1	Combinação de rastreadores	57
5.2.2	Análise de parâmetros	58
5.2.3	Retroalimentação	62
5.2.4	Adaptação	63
5.2.5	Comparação com os rastreadores base	64
5.3	Comparação com o VOT2015	66
5.4	Testes com ruído	67
5.5	Testes sem supervisão	68
6	Conclusão e trabalhos futuros	71
6.1	Conclusão	71
6.2	Trabalhos futuros	72
	Referências	73

1 Introdução

1.1 Motivação

O rastreamento visual de objetos é uma área da visão computacional que vem recebendo muita atenção nos últimos 15 anos (KRISTAN et al., 2015). O rastreamento estuda métodos para estimar a localização de objetos de interesse em quadros consecutivos de um vídeo. Um dos atrativos do rastreamento de objetos é a diversidade de aplicações em que ele pode ser utilizado, como no controle de trânsito (HWANG et al., 2004), na robótica (PAPANIKOLOPOULOS; KHOSLA; KANADE, 1993), análises esportivas (YAN et al., 2006; YU et al., 2003; EKIN; TEKALP; MEHROTRA, 2003), e vigilância (HARITAOGLU; HARWOOD; DAVIS, 2000).

O rastreamento pode ser reduzido a um problema de detecção de objetos executado em cada um dos quadros. Como o rastreamento é um processo mais simples que a detecção (BOLME et al., 2010), algoritmos de rastreamento utilizam menos recursos computacionais do que a execução de um algoritmo de detecção em todos os quadros. Desta forma, quando um objeto de interesse é encontrado em um quadro do vídeo, é interessante rastrear-lo nos quadros seguintes.

Uma variação popular desse problema é o rastreamento livre de modelo prévio, onde não há conhecimento prévio sobre o objeto de interesse. Nesta variação, o método de rastreamento recebe apenas o primeiro quadro do vídeo e a região da imagem onde se encontra o objeto de interesse para ser iniciado. Diferentemente de métodos de rastreamento e uso específico, que são treinados previamente para rastrear classes específicas de objetos, todo o conhecimento utilizado pelos métodos livres de modelo prévio devem ser obtidos no decorrer do processo de rastreamento. Essa característica torna o rastreamento mais complexo, mas permite a aplicação do rastreador em diversos cenários sem a necessidade de treinamento ou modelagem prévia. Devido a essas características, surgiram diversos trabalhos na literatura focados nessa variação do problema (KALAL; MIKOLAJCZYK; MATAS, 2012; NEBEHAY; PFLUGFELDER, 2014; BOLME et al., 2010; ZHANG; ZHANG; YANG, 2012).

Apesar da grande quantidade de trabalhos relacionados com o rastreamento visual de objetos, este problema ainda apresenta desafios, tais como: oclusão, mudança de iluminação, semelhança do objeto com o plano de fundo, e mudança de aparência do objeto durante o rastreamento. Alguns trabalhos propõem rastreadores focados em resolver somente alguns desses problemas, como mudança de escala (VOJIR; NOSKOVA; MATAS, 2014; MONTERO; LANG; LAGANIERE, 2015), oclusão (GRABNER et al., 2010),

ou rastreamento de objetos não rígidos (KWON; LEE, 2009; COMANICIU; RAMESH; MEER, 2000). Entretanto, a maioria destes rastreadores não apresenta bons resultados quando utilizados em situações diferentes das quais foi projetada para funcionar (VOJIR; MATAS; NOSKOVA, 2015). Desta forma, algumas propostas tentam unir rastreadores distintos (VOJIR; MATAS; NOSKOVA, 2015; WANG; YEUNG, 2014; BAILER; PAGANI; STRICKER, 2014). A combinação é realizada de forma a aproveitar as vantagens de cada método para obter melhores resultados.

Atualmente, a maioria dos rastreadores com melhor resultado apresenta um alto custo computacional. Isto pode ser visto no desafio VOT2015 (KRISTAN et al., 2015), onde os quinze primeiros rastreadores na classificação não são executados em tempo real, inviabilizando sua utilização em aplicações de tempo real. Quando um rastreador é utilizado em uma aplicação de tempo real, ele pode ser parte de um sistema mais complexo, sendo interessante que o rastreador não utilize todos os recursos do sistema.

Neste contexto, este trabalho propõe um método de combinação de rastreadores utilizando Filtro de Kalman (KALMAN, 1960). A proposta envolve a combinação de técnicas rápidas, de forma a produzir um rastreador rápido que ao mesmo tempo obtenha resultados superiores. Para facilitar a integração de métodos de rastreamento, o método proposto deve utilizar a saída dos rastreadores para realizar a fusão.

Métodos de rastreamento que utilizam a combinação de resultado de outros rastreadores (VOJIR; MATAS; NOSKOVA, 2015; BAILER; PAGANI; STRICKER, 2014) normalmente exigem a utilização de rastreadores completos, ou seja, métodos que sejam capazes de rastrear o objeto de interesse individualmente. O método proposto neste trabalho, chamado *Kalman filter ensemble-based tracker* (KFebT), também é capaz de trabalhar com métodos que resolvam apenas parte do problema, e não sejam capazes de rastrear o objeto de interesse sozinhos, como o método de estimativa de variação de escala proposto em (KALAL; MIKOLAJCZYK; MATAS, 2012).

Para testar o KFebT foram utilizados três rastreadores, um filtro de correlação que utiliza histogramas de gradientes orientados (HENRIQUES et al., 2015), um rastreador que utiliza histogramas de cores (VOJIR; NOSKOVA; MATAS, 2014) e um rastreador que utiliza pontos de interesse e fluxo óptico baseado no trabalho de Nebhay e Pflugfelder (2014). Para testar a utilização de métodos especialistas em resolver parte do problema na combinação, foi utilizado o método de estimativa de mudança de escala utilizado em Nebhay e Pflugfelder (2014).

Os testes do KFebT foram realizados utilizando a metodologia do desafio *Visual object tracking* (VOT) e foi utilizada a base de dados e a metodologia do desafio VOT2015 (KRISTAN et al., 2015). Os resultados foram comparados com os de outros rastreadores submetidos no desafio.

1.2 Objetivos

Este trabalho tem como principal objetivo o desenvolvimento de um método de combinação de rastreadores utilizando o filtro de Kalman. O sistema foi desenvolvido buscando a obtenção das seguintes características:

- Ser capaz de combinar diversos rastreadores utilizando apenas seus resultados e uma métrica de suas confianças;
- Ser executado em tempo real em um computador portátil, possibilitando sua utilização em aplicações móveis;
- Possuir uma interface bem definida de comunicação entre o método de combinação e os métodos de rastreamento que serão utilizados para facilitar a integração de novos métodos de rastreamento.

1.3 Contribuições

A principal contribuição deste trabalho é o desenvolvimento do método de combinação de rastreadores e composição do rastreador *Kalman filter ensemble-based tracker*(KFebT). Para gerar uma trajetória contínua e suave, é aplicada uma penalidade para movimentos inesperados nos resultados dos rastreadores.

Outra contribuição é uma estratégia que permite a utilização de métodos especialistas no métodos de combinação de resultados de rastreadores proposto.

1.4 Organização

Este trabalho é dividido em seis capítulos. No Capítulo 2 é apresentado o problema de rastreamento visual de objetos, seus desafios e os métodos de rastreamento utilizados neste trabalho. No Capítulo 3 são apresentadas as técnicas, definidas na literatura, que utilizam a combinação de rastreadores, além de definir o novo método proposto neste trabalho. O Capítulo 4 descreve o desafio VOT (*Visual object tracking*) destacando sua base de dados, métricas de avaliação e metodologia de teste. Os resultados experimentais e comparações entre os resultados obtidos a partir do método proposto e outras técnicas submetidas ao desafio VOT são apresentados no Capítulo 5. No último capítulo são discutidas as conclusões e possibilidades de melhorias futuras no método proposto.

2 Rastreamento Visual de Objetos

O rastreamento visual de objetos tem como objetivo determinar a localização de um ou mais objetos em cada um dos quadros de um vídeo. O conhecimento da localização dos objetos de interesse, assim como de suas trajetórias, são fundamentais para que as máquinas sejam capazes de ver, entender e reagir ao ambiente ([MAGGIO; CAVALLARO, 2011](#)).

Os avanços recentes nas câmeras de vídeo, tanto na melhoria de qualidade da aquisição, quanto na redução de preços, e o poder computacional dos computadores atuais favoreceram a criação de novos algoritmos e aplicações que utilizam o rastreamento visual de objetos.

As câmeras de vídeo são capazes de capturar informações do ambiente na forma de conjunto de pontos. Os métodos de rastreamento tentam modelar a relação entre a aparência do objeto de interesse e os valores dos pontos da imagem onde ele está presente para estimar a localização do objeto em todos os quadros do vídeo.

De acordo com [Maggio e Cavallaro \(2011\)](#), o principal objetivo dos métodos de rastreamento é estimar o estado do objeto de interesse no decorrer da sequência. Existem diversas formas de representar os estados dos objetos de interesse, tais como um retângulo, uma elipse, um conjunto de pontos ou um retângulo rotacionado. Na [Figura 1](#) podem ser observados exemplos dessas representações.

A forma de representação de estado mais refinada é a utilização dos pontos da imagem onde os objetos de interesse se encontram, porém devido a complexidade de se trabalhar com esta representação, ela não é muito utilizada. Na prática, a maioria dos rastreadores apresentados na literatura utilizam retângulos para representar o estado do objeto pela sua simplicidade e facilidade de uso ([NEBEHAY, 2016](#)).

É importante salientar que a representação com retângulos pode levar à imprecisão na representação do objeto de interesse, considerando que boa parte dos objetos em aplicações reais não são retangulares, ou parte do objeto de interesse ficará fora do retângulo ou boa parte da área interna do retângulo não conterá o objeto de interesse ([Figura 2](#)).

Existem alguns trabalhos que visam classificar os tipos de rastreadores de acordo com suas características. [Lepetit e Fua \(2005\)](#) identificaram dois paradigmas para as técnicas de rastreamento de objetos: o rastreamento recursivo e o rastreamento por detecção. Técnicas que utilizam o rastreamento recursivo estimam a localização do objeto de interesse no quadro atual utilizando como referência as informações do objeto no quadro anterior, esses métodos são suscetíveis a acúmulo de erros. Os métodos pertencentes ao

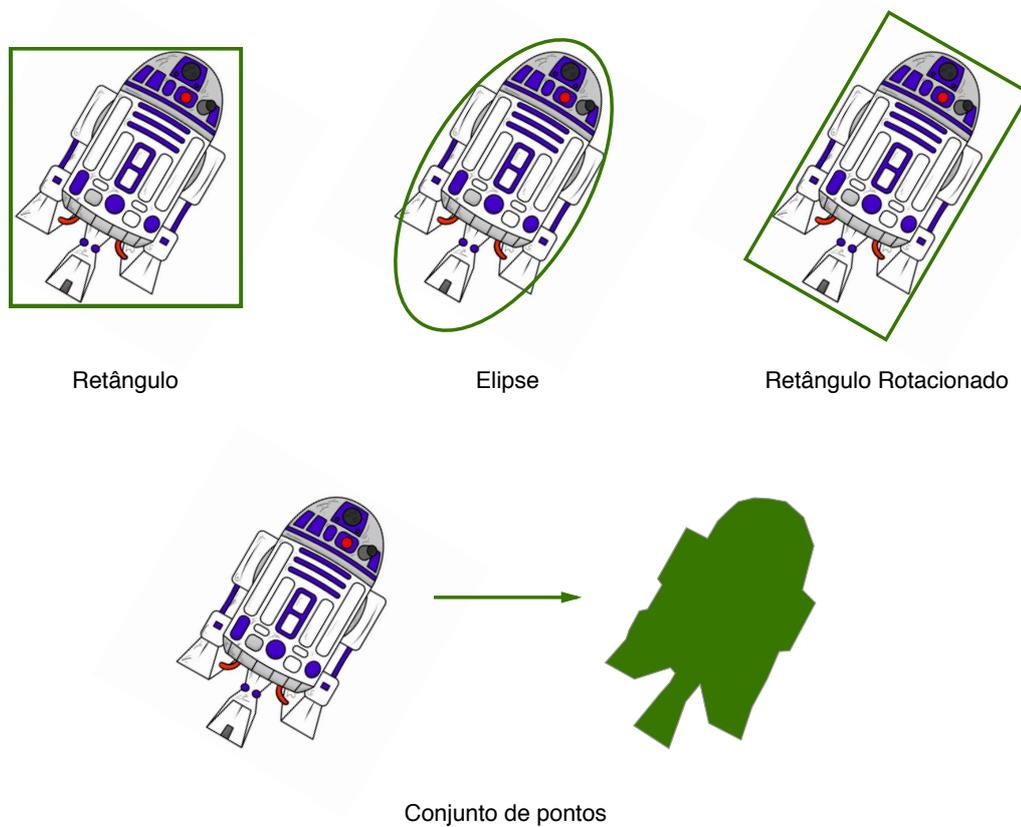


Figura 1: Exemplo de formas de representação de estado.

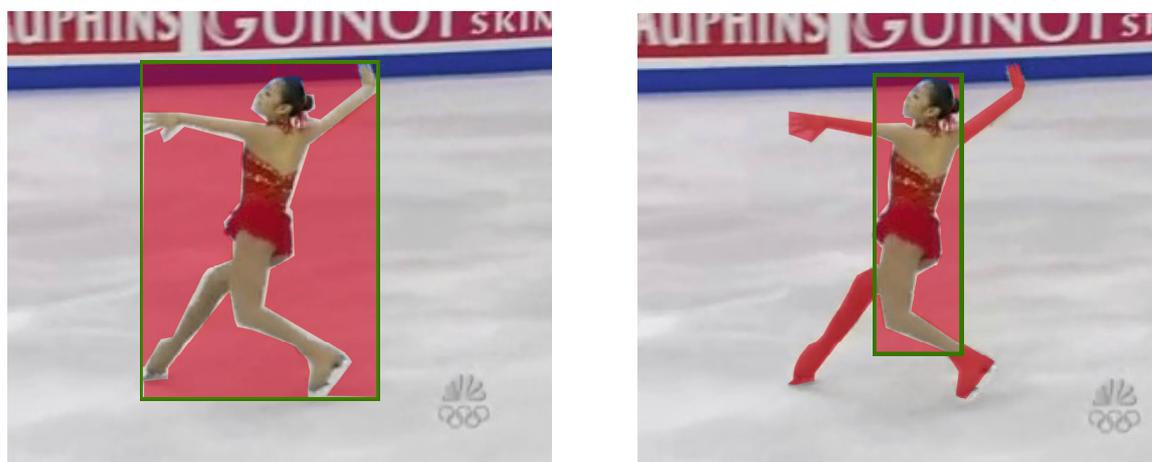


Figura 2: Imprecisões causadas pela utilização de retângulos para representar o estado do objeto de interesse.

paradigma de rastreamento por detecção utilizam somente informações pertencentes a imagem atual para encontrar o objeto de interesse, necessitando de um modelo prévio para funcionar. Entre esses paradigmas, está o paradigma de rastreamento por detecção adaptativo, onde o modelo utilizado para detectar o objeto de interesse é atualizado com as informações obtidas no decorrer do rastreamento.

Maggio e Cavallaro (2011) classificam os métodos de rastreamento de acordo com o nível de interação humana necessária para a localização do objeto de interesse. Métodos de rastreamento manuais necessitam de interação com um usuário em todos os quadros para localizar o objeto de interesse. Métodos de rastreamento automáticos não necessitam de intervenção humana, eles devem ser capazes de iniciar o rastreamento identificando o objeto de interesse com seus conhecimentos prévios e localizar o objeto nos quadros seguintes de forma autônoma. Já os métodos de rastreamento semi-automáticos precisam ser iniciados pelo usuário ou por outro sistema, indicando qual objeto deve ser rastreado. Depois da iniciação, eles devem ser capazes de realizar o rastreamento de forma autônoma. Neste trabalho são estudados métodos de rastreamento semi-automáticos.

2.1 Rastreamento de objetos sem modelo prévio

Alguns estudos visam rastrear uma classe específica de objetos, tais como carros (KOLLER; WEBER; MALIK, 1994) ou pessoas (GAVRILA, 1999; ANDRILUKA; ROTH; SCHIELE, 2008; SHU et al., 2012). Nessas situações, é possível realizar uma modelagem com conhecimento sobre o objeto de interesse e seu comportamento esperado para realizar o rastreamento, obtendo uma solução robusta para esses problemas. Esses rastreadores de uso específico possuem uma grande limitação, sendo necessário desenvolver ou treinar um rastreador para cada objeto ou classe de objetos que se deseje rastrear, dificultando sua aplicabilidade. A principal vantagem de métodos sem modelo prévio é sua aplicabilidade em diversos cenários, sem a necessidade de conhecimento prévio do objeto de interesse (NEBEHAY; PFLUGFELDER, 2014).

Neste trabalho é estudada uma variação do problema de rastreamento conhecida na literatura como *model-free object tracking* ou *one-shot object tracking*, referida aqui como rastreamento de objetos sem modelo prévio, onde o sistema de rastreamento não recebe nenhuma informação prévia sobre o objeto de interesse.

Na prática, a única informação que o rastreador recebe na sua iniciação é a primeira imagem da sequência junto com a localização do objeto de interesse presente. O sistema de rastreamento deve ser capaz de determinar a localização do objeto de interesse nas imagens seguintes utilizando apenas as informações obtidas nas imagens anteriores, sendo que a informação mais confiável sobre o objeto de interesse é a sua localização na primeira imagem, pois a localização dele nas imagens seguintes é determinada pelo próprio

rastreador.

Caso a técnica utilizada pelo rastreador precise criar um modelo para representar o objeto de interesse, ele deve ser criado em tempo de execução. O modelo criado pode ser atualizado no decorrer do processo de rastreamento, mas só podem ser utilizadas informações obtidas pelo próprio rastreador no processo de atualização.

Neste trabalho também foram consideradas algumas limitações, diminuindo assim o escopo do problema. O rastreador deve seguir um único alvo e as imagens são capturadas por uma única câmera. O rastreamento é de curto prazo, ou seja, o rastreador não precisa realizar a re-deteção do objeto de interesse caso o perca. Também foram colocadas algumas limitações nos rastreadores, eles não podem possuir memória, ou seja, não devem ser utilizadas informações anteriores à sua iniciação. Toda a informação utilizada pelo rastreador deve ser obtida no primeiro quadro e nos quadros que já foram processados. Os quadros seguintes ao que está sendo processado não podem ser utilizados.

2.2 Desafios

O rastreamento visual de objetos apresenta diversos desafios que podem alterar a forma como o objeto é capturado pela câmera ou fazer com que o rastreador confunda o objeto de interesse com outros objetos na imagem.

Um dos principais desafios para os algoritmos de rastreamento de objetos é a semelhança do objeto de interesse com outros objetos contidos na imagem (MAGGIO; CAVALLARO, 2011). Essa semelhança pode confundir o rastreador, fazendo com que ele não consiga diferenciar entre características extraídas do objeto de interesse e as extraídas de outros objetos. Um exemplo desse problema é apresentado na Figura 3, onde o objeto de interesse, a folha circulado pelo retângulo vermelho, é semelhante a diversas outras folhas presentes na imagem.

A mudança na iluminação é definida como a média das diferenças absolutas entre a intensidade do objeto no primeiro e no restante dos quadros. Ela pode ser alterada na sua direção, intensidade ou cor, mudando a forma como o objeto é capturado pela câmera. Por exemplo, essa alteração pode ocorrer em ambientes externos quando nuvens obstruem o sol. Na Figura 4 é apresentado um exemplo de mudança na iluminação causada pela mudança na direção da câmera em relação a uma fonte de luz.

A mudança na aparência atua de forma oposta à semelhança com outros objetos na imagem, fazendo com que as características extraídas do objeto de interesse não sejam classificadas como pertencentes a ele. Ela pode ocorrer por diversos motivos, como a mudança do ângulo de captura do objeto ou deformações no objeto de interesse. Na Figura 5 pode-se observar um exemplo de mudança na aparência, onde uma lanterna em



Figura 3: Exemplo de semelhança com o plano de fundo. O objeto de interesse é a folha destacada pelo retângulo vermelho, sendo que existem diversas folhas semelhantes a ela na imagem. A imagem faz parte da base de dados VOT2015.



Figura 4: Exemplo de mudança de iluminação causada pela mudança na direção da câmera em relação a uma fonte de luz. As imagens fazem parte da base de dados VOT2015.

formato de tigre é aberta, alterando a sua aparência.

Outro desafio é a oclusão, que ocorre quando a visibilidade do objeto de interesse é parcialmente ou totalmente obstruída por outro objeto. Oclusões ocorrem quando o objeto de interesse passa atrás de um objeto estático, como uma árvore ou poste, ou quando outro objeto fica entre ele e a câmera. A Figura 6 apresenta um exemplo de oclusão, onde o objeto de interesse, a moto branca, sofre oclusão parcial nas imagens 1, 2, 3 e 5 e oclusão total na imagem 4.

A mudança de escala é um problema que afeta diversos algoritmos de rastreamento, como o *Mean shift* (COMANICIU; RAMESH; MEER, 2000) e os filtros de correlação (BOLME et al., 2010), que trabalham com uma janela de busca de tamanho fixo. Existem diversos trabalhos com foco na resolução deste problema em rastreadores mais simples (VOJIR; NOSKOVA; MATAS, 2014; MONTERO; LANG; LAGANIERE, 2015; DANELLJAN et al., 2014). Um exemplo de mudança de escala no objeto de interesse pode ser visto na Figura 7.



Figura 5: Exemplo mudança na aparência do objeto de interesse. As imagens fazem parte da base de dados VOT2015.



Figura 6: Exemplo de oclusão. As imagens fazem parte da base de dados VOT2015.



Figura 7: Exemplo de mudança de escala, o objeto de interesse é o pássaro destacado pelo retângulo vermelho. As imagens fazem parte da base de dados VOT2015.

O borrão ocorre quando o objeto de interesse ou a câmera se movem muito rápido ou quando o foco da câmera não está ajustado para o objeto de interesse. Quando o objeto de interesse está borrado, suas bordas e seus contornos são atenuados, dificultando a operação de técnicas que utilizam bordas, pontos de interesse ou o formato do objeto de interesse. Na Figura 8 é apresentado um exemplo de borrão que ocorreu devido a movimentação rápida da câmera.

O método de representação do objeto de interesse deve ser robusto a esses desafios, sendo capaz de identificar o objeto de interesse em ocorrências moderadas desses problemas. Como o objeto de interesse é inicialmente desconhecido, é impossível utilizar técnicas de aprendizado de máquina *offline* para incorporar no modelo dos objetos as mudanças na sua aparência decorrentes destes desafios.



Figura 8: Exemplo de borrão devido a movimentação rápida da câmera. As imagens fazem parte da base de dados VOT2015.

2.3 Métodos de rastreamento

Nesta seção serão apresentados os métodos de rastreamento utilizados neste trabalho.

2.3.1 *Mean-shift*

O algoritmo *Mean-shift* (MS), proposto por [Fukunaga e Hostetler \(1975\)](#), é um método não paramétrico utilizado para encontrar os máximos de uma função de densidade de probabilidades.

Seu primeiro uso na visão computacional foi proposto por [Comaniciu, Ramesh e Meer \(2000\)](#), onde foi utilizado para realizar o rastreamento visual de objetos. O algoritmo MS realiza o rastreamento de forma iterativa pela minimização da distância entre duas funções de densidade de probabilidades, uma representando o histograma de cores do objeto de interesse e outro histograma da região candidata.

O algoritmo MS originalmente utiliza uma janela fixa, sendo incapaz de estimar a variação de escala do objeto alvo. Esta limitação inicialmente foi tratada por [Comaniciu, Ramesh e Meer \(2000\)](#) variando o tamanho da janela ($\pm 10\%$) durante as iterações do algoritmo, sendo escolhido o tamanho que maximizar a similaridade.

[Vojir, Noskova e Matas \(2014\)](#) propuseram melhorias no método definido por [Comaniciu, Ramesh e Meer \(2000\)](#). As principais foram uma estimativa de mudança de escala melhorada, uma checagem de consistência retroativa e a utilização da distância do ponto para o centro do objeto para ponderar o cálculo do histograma. Essa técnica é chamada de ASMS (*Adaptive Scale Mean-Shift*).

A estimativa de escala utiliza o termo de regularização $rs(h)$. Este termo é utilizado assumindo que a escala do alvo não vai mudar drasticamente entre dois quadros. Outra melhoria é a inclusão de uma checagem de consistência retroativo para se recuperar de erros grosseiros cometidos na estimativa de escala. Nela, o rastreamento é realizado do quadro atual, utilizando a estimativa que foi encontrada, para o quadro anterior. Caso a

diferença entre as escalas estimadas seja maior que um certo limiar, é utilizada a escala anterior.

2.3.2 Filtros de correlação

Os filtros de correlação são uma abordagem popular na área de rastreamento, sendo utilizados em 16 dos 62 rastreadores comparados no desafio VOT2015 (KRISTAN et al., 2015). Filtros de correlação mais antigos utilizam modelos simples e geralmente falham quando utilizados no rastreamento de objetos (BOLME et al., 2010). Abordagens mais modernas, como o *Average of Synthetic Exact Filters* (ASEF) (BOLME; DRAPER; BEVERIDGE, 2009) e o *Unconstrained Minimum Average Correlation Energy* (UMACE) (MAHALANOBIS et al., 1994), possuem melhores resultados no rastreamento, mas seus processos de treinamento necessitam de muitas amostras. Essa característica torna sua utilização inviável para o problema abordado neste trabalho, pois só é disponibilizada uma amostra do objeto de interesse para a realização do treinamento.

Os filtros de correlação começaram a chamar mais atenção na área de rastreamento visual de objetos quando passaram a utilizar o domínio de frequência, onde a operação de correlação é uma multiplicação ponto a ponto, tornando este processo muito mais rápido. Um exemplo de um desses filtros é o *Minimum Output Sum of Squared Error* (MOSSE) (BOLME et al., 2010), que obtém bons resultados enquanto roda a centenas de quadros por segundo.

O CSK (*Structure of tracking-by-detection with Kernels*) (HENRIQUES et al., 2012) aprimora o MOSSE utilizando uma matriz circular para aumentar a capacidade descritiva do modelo. A matriz circular consegue trabalhar com todos os deslocamentos cíclicos do objeto de interesse.

Inspirado no CSK, o *kernelized correlation filter* (KCF) (HENRIQUES et al., 2015) adiciona a capacidade de trabalhar com características multicanais. No KCF foi utilizado o Histograma de Gradientes Orientados (HGO) (DALAL; TRIGGS, 2005) como descritor de características.

Apesar das melhorias propostas no CSK e KCF, eles ainda sofrem da limitação de não estimarem a mudança de escala do objeto de interesse, utilizando uma janela de tamanho fixo durante todo o processo de rastreamento. Danelljan et al. (2014) propuseram um método de estimativa de escala utilizando um filtro de correlação discriminativo de uma dimensão. O método de estimativa de escala é aplicado após um filtro MOSSE (que utiliza características HOG) encontrar a translação do objeto. Montero, Lang e Laganier (2015) também propuseram a utilização de um método de estimativa de escala em conjunto com filtro de correlação. Ele utilizou um método baseado em pontos de interesse que calculava a variação da distância entre os pontos nos quadros de forma

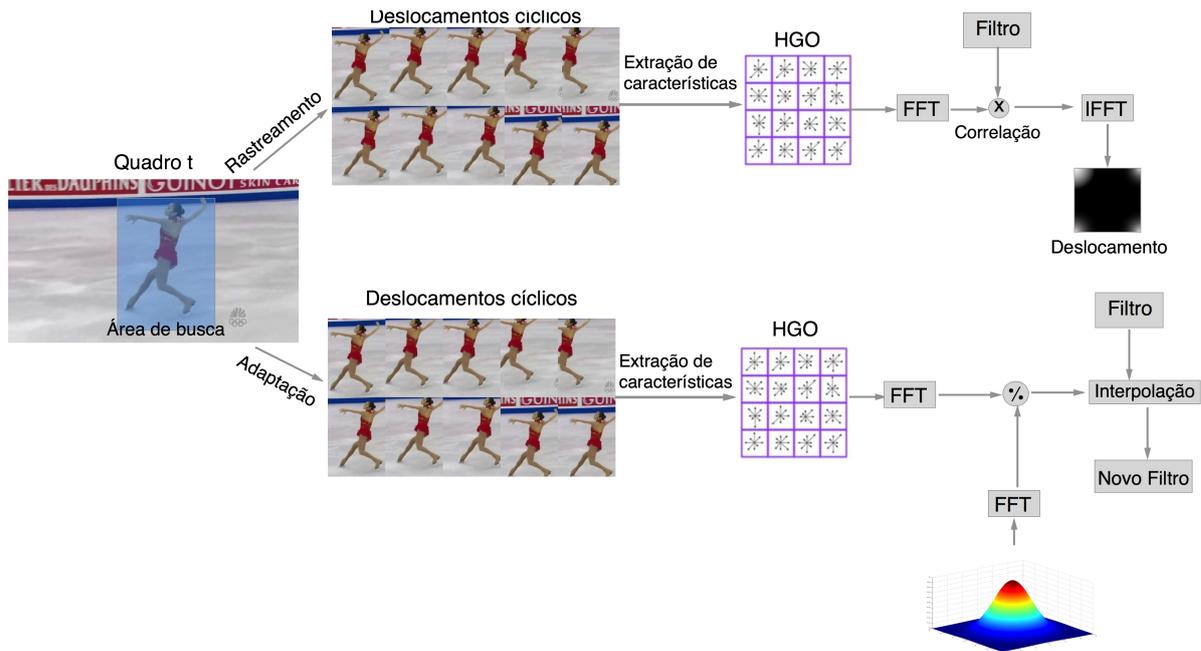


Figura 9: Fluxograma do KCF.

similar ao utilizado por [Kalal, Mikolajczyk e Matas \(2012\)](#) para estimar a variação de escala do objeto de interesse.

Neste trabalho foi utilizado o KCF. O fluxograma de funcionamento do KCF é apresentado na Figura 9. Para cada novo quadro, o processo do KCF é dividido em duas etapas, o rastreamento e a atualização do modelo. O processo de rastreamento é iniciado criando a matriz com os deslocamentos cíclicos da área de busca. Nas características HGO extraídas dessa matriz é aplicada a transformada rápida de Fourier (FFT) ([COCHRAN et al., 1967](#)). A operação de correlação ocorre no domínio de Fourier, onde ela é uma multiplicação ponto a ponto entre matrizes. A matriz das características HGO no domínio de Fourier é multiplicada pelo filtro que já está no domínio de Fourier. No resultado dessa multiplicação é aplicada a transformada inversa rápida de Fourier (IFFT), encontrando, então, o resultado da correlação. A região onde se encontra o pico do resultado da correlação indica o deslocamento do objeto de interesse, sendo que no caso do objeto estar na mesma posição, o pico se encontrará no canto superior esquerdo.

A operação de atualização é semelhante a de rastreamento. Ela começa aplicando o deslocamento encontrado no rastreamento na área de busca. Depois são encontradas os deslocamentos cíclicos e extraídas as características HGO. Nas características é aplicada a FFT. Para realizar o treinamento é utilizada uma função gaussiana de duas dimensões com pico no centro. Na função gaussiana é aplicada a FFT. O treinamento é realizada dividindo a função gaussiana no domínio de Fourier com as características HGO no domínio de Fourier. O resultado então é interpolado com o filtro existente, criando o novo filtro.

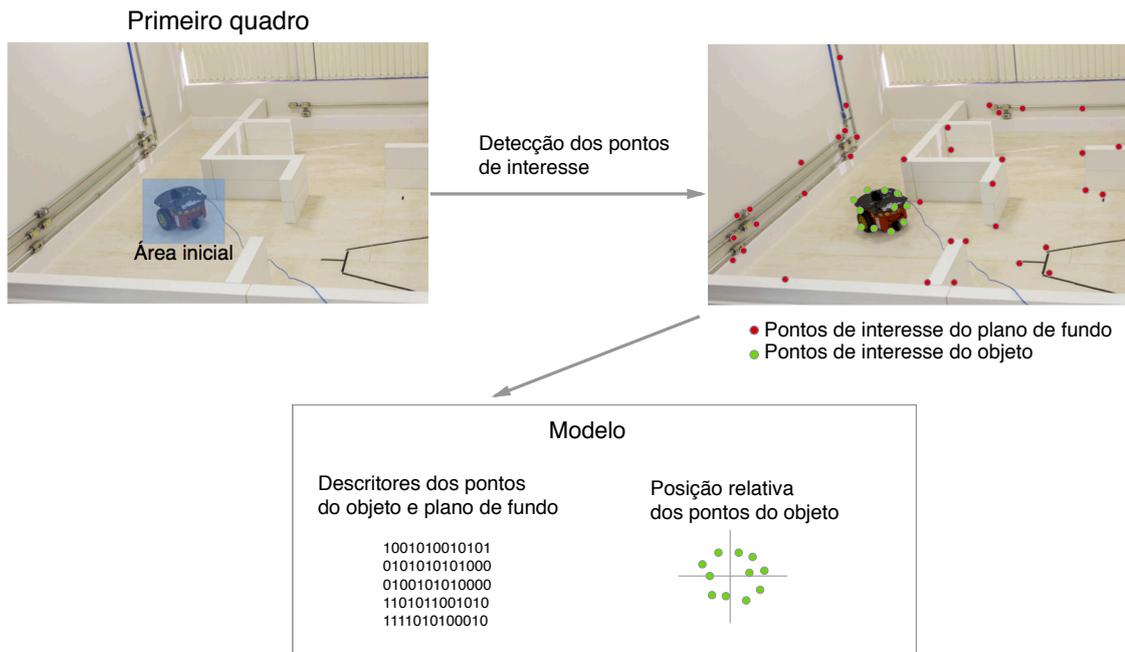


Figura 10: Iniciação do CMT.

2.3.3 CMT

O *Clustering of Static-adaptive Correspondences for Deformable Object Tracking* (CMT) (NEBEHAY; PFLUGFELDER, 2015) emprega uma representação baseada em pontos de interesse. Pontos de interesse correspondentes em cada quadro são obtidos pela combinação das abordagens de casamento de pontos e rastreamento utilizando o fluxo óptico.

A localização do objeto de interesse é estimada utilizando um mecanismo de votação que considera a geometria da constelação de pontos de interesse. A variação de escala é estimada utilizando a variação mediana de distância entre os pares de pontos de interesse do quadro atual e os pontos correspondentes na modelagem do objeto.

A iniciação do CMT é apresentada na Figura 10. O método começa recebendo o primeiro quadro e a área inicial onde se encontra o objeto de interesse. É utilizado um detector de pontos de interesses para encontrar os pontos de interesse do objeto e do plano de fundo. Os pontos são descritos e são armazenados os descritores dos pontos e a suas posições relativas como modelo. No trabalho original foi utilizado o *Features from Accelerated Segment Test* (FAST) como detector de características (ROSTEN; DRUMMOND, 2006) e como descritor de pontos de interesse foi utilizado o *Binary Robust Invariant Scalable Keypoints* (BRISK) (LEUTENEGGER; CHLI; SIEGWART, 2011).

Para cada novo quadro, é realizado o procedimento apresentado na Figura 11. Inicialmente são realizados dois procedimentos, a obtenção de pontos utilizando o fluxo óptico e do casamento de pontos de interesse. No fluxo óptico é utilizado o quadro ante-

rior (N-1) e seus pontos de interesse para encontra-los no quadro atual (N). É utilizado o Fluxo óptico de Lucas-Kanade (LUCAS; KANADE et al., 1981) e é realizada a validação retroativa (proposta por Kalal, Mikolajczyk e Matas (2010)) para eliminar inconsistências.

O casamento de pontos de interesse começa detectando e descrevendo os pontos de interesse do quadro N. Os descritores destes pontos são comparados com os descritores obtidos na iniciação para combiná-los com os pontos do objeto de interesse. Nesse processo é realizada uma validação com os descritores do plano de fundo obtidos na inicialização. Ao final, são encontrados os pontos mais similares aos do objeto de interesse.

Os pontos obtidos pelo fluxo óptico e pelo casamento de pontos são combinados. A combinação ocorre eliminando todos os pontos obtidos pelo fluxo óptico que correspondam a um ponto encontrado pelo casamento de pontos. Com os pontos restantes é realizada a estimativa da mudança de escala do objeto de interesse utilizando a mudança mediana de distância entre os pares de pontos. Por último, é realizada a estimativa do novo centro do objeto de interesse utilizando a estratégia de votação. Cada ponto de interesse vota em uma localização candidata a centro do objeto e os pontos são agrupados de acordo com a localização do centro estimado. O grupo que receber mais votos é utilizado para definir o centro do objeto de interesse e utilizados no fluxo óptico do próximo quadro, os demais grupos são descartados.

A utilização de detectores de pontos de interesse rápidos e de descritores binários permite que o CMT seja executado em tempo real (NEBEHAY; PFLUGFELDER, 2015).

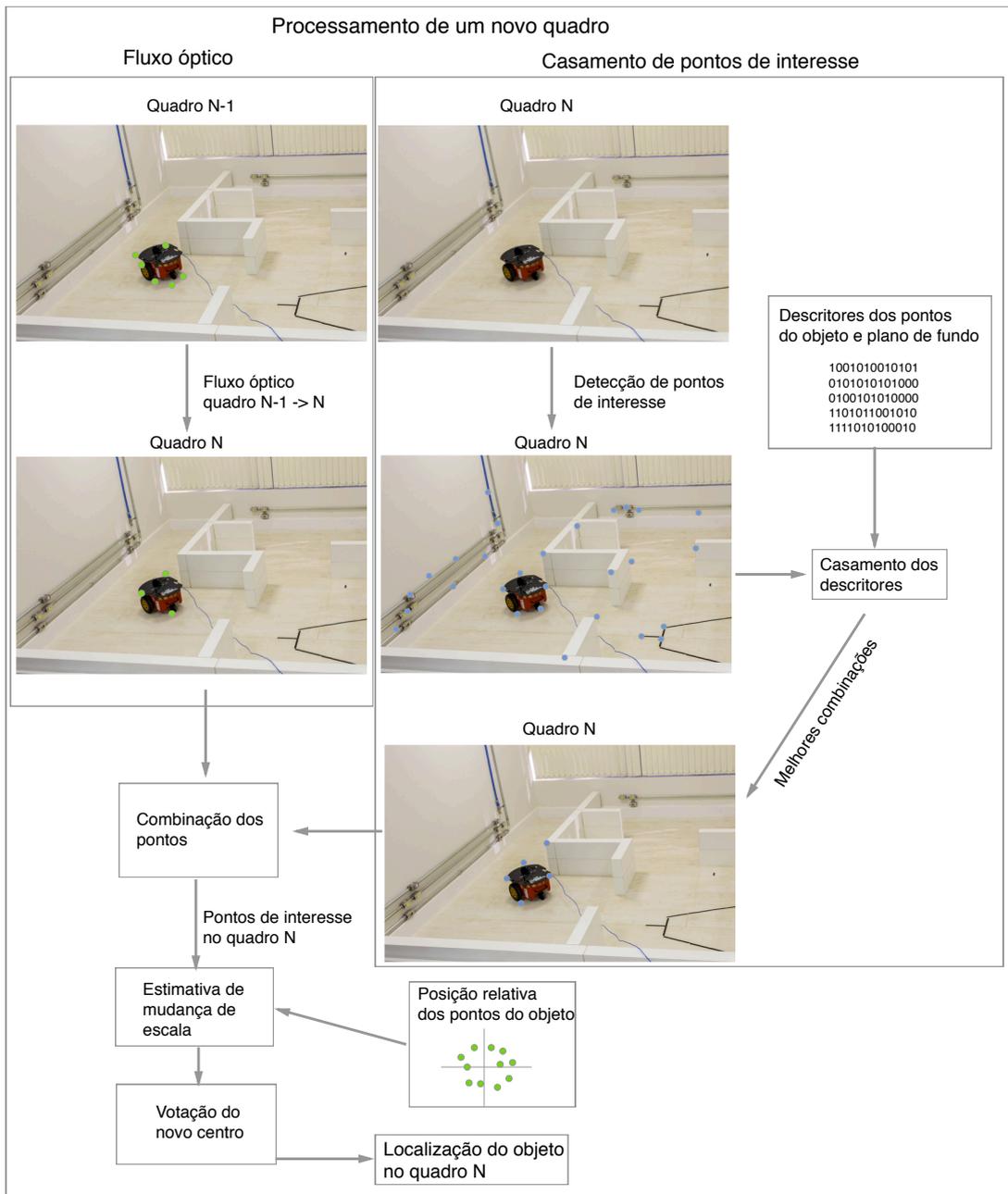


Figura 11: Processamento de um novo quadro realizado pelo CMT para encontrar a localização do objeto de interesse.

3 Combinação de Rastreadores

3.1 Trabalhos relacionados

Devido à popularidade da área de rastreamento visual de objetos, surgiram diversos algoritmos propostos nos últimos anos empregando as mais diversas estratégias, desde técnicas que utilizam o rastreamento de partes do objeto como o CMT (NEBEHAY; PFLUGFELDER, 2014) que utiliza fluxo óptico e combinação de características, até técnicas baseadas em redes neurais convolucionais (NAM; HAN, 2015). Cada uma destas abordagens possuem suas vantagens e desvantagens.

Com toda essa diversidade de abordagens, começaram a surgir rastreadores que utilizam combinações desses métodos com objetivo de aproveitar o melhor de cada técnica, superando assim limitações individuais. Segundo Bailer, Pagani e Stricker (2014), um rastreador com resultados gerais bons pode falhar em situações onde um rastreador com resultados gerais ruins não falham. Desta forma, mesmo um rastreador com resultados não tão bons pode contribuir para melhorar o resultado final.

Na literatura existem várias propostas que utilizam combinações de rastreadores de forma bem específica para superar limitações ou melhorar os resultados. Um exemplo são os filtros de correlação como o KCF, que utiliza um filtro de tamanho fixo, não sendo capaz de estimar a mudança de escala do objeto de interesse. Alguns trabalhos propõem a utilização de métodos que estimam a mudança de escala em conjunto com o KCF.

Ali et al. (2015) propuseram um método que utiliza um filtro de correlação em conjunto com um filtro de Kalman. Caso a diferença entre o resultado obtido pelo filtro de correlação e a previsão do filtro de Kalman seja maior que um determinado limite, é utilizado um rastreador que utiliza *mean shift* para reiniciar o filtro de correlação.

Em geral, essas combinações realizadas de forma muito específica limitam as técnicas que podem ser combinadas, pois uma técnica só pode ser substituída por outra técnica semelhante.

Com objetivo de flexibilizar a combinação de rastreadores, surgiram métodos que realizam a combinação dos resultados de rastreadores. Desta forma, é possível combinar qualquer rastreador que represente seu resultado de uma forma que possa ser utilizada pelo método de fusão.

A combinação de rastreadores pode ser utilizada para resolver um problema específico, como o rastreador proposto por Santner et al. (2010) que combina o resultado de três métodos de rastreamento com diferentes níveis de adaptação para reduzir o erro

causado por atualizações imprecisas em seus modelos. Os resultados dos rastreadores são combinados utilizando uma estratégia em cascata com regras fixas. Outra possibilidade é a fusão de um mesmo rastreador com diferentes parâmetros, como em (ZHANG; MA; SCLAROFF, 2014), onde é proposto um modelo de rastreamento que utiliza múltiplos especialistas. Na prática, os especialistas são instâncias de um mesmo rastreador criados em tempos diferentes. Essa abordagem tem como objetivo diminuir o erro causado por atualizações indesejadas no modelo do rastreador. Os melhores especialistas são escolhidos de forma a maximizar uma função de custo baseada em entropia.

Os métodos mais próximos ao proposto neste trabalho são apresentados por Bailer, Pagani e Stricker (2014) e Vojir, Matas e Noskova (2015), os quais propõem a fusão do resultado de múltiplos rastreadores independentes.

Vojir, Matas e Noskova (2015) apresentaram um método de treinamento para um Modelo Oculto de Markov (MOM), o qual é utilizado para combinar o resultado de três rastreadores e um detector. Os estados do MOM são representados por vetores binários, representando as falhas de cada rastreador. Um problema dessa modelagem é a adição de um novo rastreador que dobra a quantidade de estados no MOM.

Bailer, Pagani e Stricker (2014) combinam o resultado de diversos métodos de rastreamento. Os rastreadores são executados independentemente e seus resultados são combinados utilizando uma estratégia baseada em campos de atração. Neste trabalho também é aplicada uma otimização de trajetória para obter uma trajetória suave e contínua.

Segundo Bailer, Pagani e Stricker (2014), nem todos os rastreadores melhoram os resultados quando combinados. Caso sejam utilizados rastreadores semelhantes, que utilizam características similares e que falham nas mesmas situações, esses rastreadores podem não contribuir na melhoria do resultado final.

Essas técnicas baseadas na combinação de resultados de rastreadores atingem resultados melhores que os rastreadores de que são compostos.

3.2 Combinação de rastreadores utilizando o Filtro de Kalman

O método proposto neste trabalho combina o resultado de múltiplos rastreadores ou métodos especialistas de forma simples e rápida utilizando um Filtro de Kalman. Esta abordagem pode combinar diversos rastreadores, sendo necessário apenas uma medida que avalie o resultado de cada um deles.

Neste trabalho, são chamados de métodos especialistas os métodos que resolvem apenas parte do problema de rastreamento, não sendo capazes de rastrear o objeto de interesse sozinhos. Nessa categoria, são encontrados os métodos que estimam apenas a

mudança de escala do objeto de interesse, como o método do CMT para estimar a mudança de escala ou o filtro de correlação de uma dimensão, proposto por [Montero, Lang e Laganieri \(2015\)](#).

Para o filtro de Kalman, cada rastreador se comporta como um sensor que está medindo a localização do objeto de interesse. Os rastreadores devem entregar essas medições para o filtro de Kalman para que ele realize a fusão. Para simplificar a integração dos rastreadores com o método proposto, são utilizados os resultados dos rastreadores como medições. Utilizando esta estratégia, são realizadas poucas alterações nos rastreadores.

Segundo [Bailer, Pagani e Stricker \(2014\)](#), a fusão de rastreadores pode ser realizada com ou sem retroalimentação. Quando ele é realizada sem retroalimentação, a integração do rastreador é mais simples. Por outro lado, caso o rastreador perca o objeto de interesse e não receba retorno do método de fusão, ele pode continuar perdido no resto da sequência. Para evitar que algum rastreador fique perdido enquanto o resultado da combinação se mantém rastreando o objeto de interesse, o método proposto neste trabalho realiza a retroalimentação dos rastreadores, informando o resultado final da fusão.

Entre os métodos utilizados na combinação, podem existir métodos que realizem adaptações em seus modelos. Para facilitar a integração dos rastreadores, caso um método adapte seu modelo, ele deve utilizar seu próprio resultado ao invés do resultado final da fusão.

Nas próximas seções serão apresentados um resumo do Filtro de Kalman, a modelagem do filtro de Kalman utilizada neste trabalho, o método proposto e sua implementação.

3.2.1 Filtro de Kalman

O Filtro de Kalman (FK) ([KALMAN, 1960](#)) é um método matemático utilizado para filtrar e prever problemas lineares. Ele é composto por dois processos, a previsão e a correção. No processo de previsão, o estado do sistema no tempo t é estimado utilizando o modelo e o estado no tempo $t - 1$. O modelo é uma aproximação do comportamento do sistema que se está modelando (ex. um modelo físico). No processo de correção, os valores medidos são utilizados para corrigir o estado do sistema.

O estado do Filtro de Kalman é composto por duas variáveis, a matriz de valor do estado x e a matriz de covariância do estado P . O estado é modelado como uma distribuição normal(3.1), centrada no valor do estado e com a covariância do estado.

$$g(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2} \quad (3.1)$$

Onde μ é a média e σ^2 é a variância da distribuição.

O processo de previsão do Filtro de Kalman pode ser dividido em dois passos, estimativa dos valores do estado e estimativa da matriz de correlação. A estimativa dos valores do estado é definida na equação 3.2.

$$x_t^- = Ax_{t-1} + Bu_t \quad (3.2)$$

Onde A é a matriz de transição que modela o sistema que esta sendo observado, x_{t-1} é a matriz dos valores do estado no tempo $t - 1$ e o termo Bu_t é utilizado quando se possui controle sobre o fenômeno ao qual se está estimando. Por exemplo, caso se esteja estimando a distância entre um robô e uma parede, e o controle desse robô envia um comando para ele se aproximar uma distância d da parede, essa distância pode ser utilizada na variável u . B é o modelo que relaciona os valores de u com os valores de x .

O próximo passo é a estimativa da matriz de covariância (P), que é definido pela equação 3.3.

$$P_t^- = AP_{t-1}A^T + Q \quad (3.3)$$

Onde Q é a incerteza do processo de previsão.

O processo de correção acontece quando se possui uma nova medição. Ele é dividido em três passos: o cálculo do ganho de Kalman (K_t), a correção do estado, e a estimativa da matriz de correlação.

O ganho de Kalman define o quanto o estado deve ser alterado pelos valores obtidos pela medição. Ele é calculado de acordo com a equação 3.4.

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1} \quad (3.4)$$

Onde R é a matriz de incerteza das medidas e H é o modelo que relaciona as variáveis medidas com as variáveis do estado do Filtro de Kalman.

O novo estado é obtido utilizando o ganho de Kalman para definir o quanto da diferença entre os valores medidos z_t e os valores previstos x_t^- será adicionado ao novo estado x_t (equação 3.5).

$$x_t = x_t^- + K_t(z_t - Hx_t^-) \quad (3.5)$$

O último passo do processo de correção é a estimativa da matriz de covariância, definida na equação 3.6.

$$P_t = P_t^- - K_t * H * P_t^- \quad (3.6)$$

Toda vez que o processo de previsão é realizado, a matriz de covariância tem seu valor somado com a incerteza do processo (Eq. 3.3), fazendo o estado atual mais incerto. Já o processo de correção reduz a incerteza do estado.

3.2.2 Modelagem do Filtro de Kalman

Para criar a modelagem do filtro de Kalman, é necessário considerar as características e as limitações dos rastreadores utilizados. Neste trabalho foram utilizados rastreadores que modelam seus estados como retângulos com as laterais alinhadas com o eixo da imagem. Esse retângulos podem ser representados pela posição do seu centro $C(x, y)$ e pela sua escala (e).

O estado do filtro de Kalman no tempo t (matriz x_t na equação 3.7) é composto pelas três variáveis que descrevem o estado dos rastreadores (x , y , e e) e pela primeira e segunda derivada destas variáveis.

A matriz de transição A é definida na equação 3.7. Para simplificar a modelagem do sistema, é considerado que o intervalo de aquisição das imagens (dt) é constante, fazendo com que o problema possa ser modelado de forma linear. Em aplicações reais, pode existir uma pequena variação no intervalo entre a aquisição das imagens. O valor dt pode ser determinado pela capacidade de processamento do computador onde o rastreamento está sendo executado, ou pela velocidade de captura da câmera utilizada. Nos testes realizados neste trabalho, foram utilizadas sequências de vídeos previamente capturadas. Como na base de dados das sequências não existiam informações sobre a taxa de captura da câmera utilizada, e nem garantia que foi utilizada a mesma câmera, o valor de dt foi fixado em 50 milissegundos, que é o intervalo de aquisição de uma câmera que captura 20 quadros por segundo.

$$A = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_t = \begin{bmatrix} x \\ y \\ e \\ \dot{x} \\ \dot{y} \\ \dot{e} \\ \ddot{x} \\ \ddot{y} \\ \ddot{e} \end{bmatrix} \quad (3.7)$$

É importante considerar que o problema estudado lida com movimentações aleatórias, não existindo informação ou controle sobre a movimentação do objeto de interesse. Desta forma, o processo de previsão realizado pelo filtro de Kalman pode obter resultados

insatisfatórios caso o objeto de interesse subitamente altere o seu padrão de movimentação. Por esta razão, a incerteza do processo de previsão, Q na equação 3.3, é fixa e tende a ser alta se comparada com as incertezas obtidas pelos rastreadores.

Caso o rastreamento seja utilizado em aplicações onde a câmera esteja em movimento e exista conhecimento ou controle sobre seu movimento, essa informação pode ser utilizada para melhorar a previsão do filtro de Kalman na equação 3.2. Por exemplo, caso a câmera esteja em um robô e se tenha conhecimento sobre a sua movimentação, essa movimentação pode ser facilmente integrada no método proposto.

Considerando que a trajetória do objeto de interesse é contínua e que em curtos intervalos de tempo mantém um padrão de movimentação constante, foi utilizada uma penalidade (p) para produzir uma trajetória suave. A penalidade diminui a confiança dos rastreadores que estimarem a localização do objeto de interesse diferente da prevista pelo filtro de Kalman. A equação 3.8 apresenta o cálculo da penalidade.

$$p = ((bb_x - previsaoFK_x)/bb_{width})^2 + ((bb_y - previsaoFK_y)/bb_{height})^2 \quad (3.8)$$

onde bb é a localização estimada pelo rastreador, e $previsaoFK$ é a localização prevista pelo filtro de Kalman (x_t^- na equação 3.2). A penalidade é calculada separadamente para cada rastreador.

A incerteza dos rastreadores foi definida como uma função exponencial que depende da confiança do resultado do rastreador e da penalidade p (Eq. 3.9). A confiança dos rastreadores é uma medida que avalia a qualidade do resultado dos rastreadores. Na seção 3.3 são apresentados mais detalhes sobre o cálculo da confiança para cada rastreador utilizado neste trabalho.

$$u = e^{-(\alpha \text{ confiança} - \beta p)} \quad (3.9)$$

Considerando que a confiança dos rastreadores pode ter significados diferentes, são utilizados dois parâmetros, α e β , para ajustar a confiança e a penalidade para valores razoáveis na equação 3.9. O parâmetro α também pode ser utilizado para controlar a influência de cada rastreador na fusão. Caso um rastreador geralmente apresente resultados melhores que os outros, o seu α pode ser maior que a dos outros. O parâmetro β controla a quantidade de penalidade na incerteza aplicada no cálculo da incerteza dos rastreadores. O mesmo valor de β foi utilizado para todos os rastreadores para diminuir a quantidade de parâmetros a serem ajustados.

Para tornar o método mais flexível e adaptável, as variáveis (x,y,e) de cada um dos rastreadores podem ter valores de incerteza diferentes. O valor da incerteza de cada variável é igual a incerteza do rastreador multiplicada por uma constante. Esta modelagem

permite a fusão de rastreadores que não estimam todas as variáveis utilizadas no estado dos rastreadores. No caso de um rastreador não estimar uma das variáveis, esta variável deve ser reportada com o seu valor obtido na retroalimentação da iteração anterior. Na figura 12 são apresentados dois exemplos que ilustram essa situação.

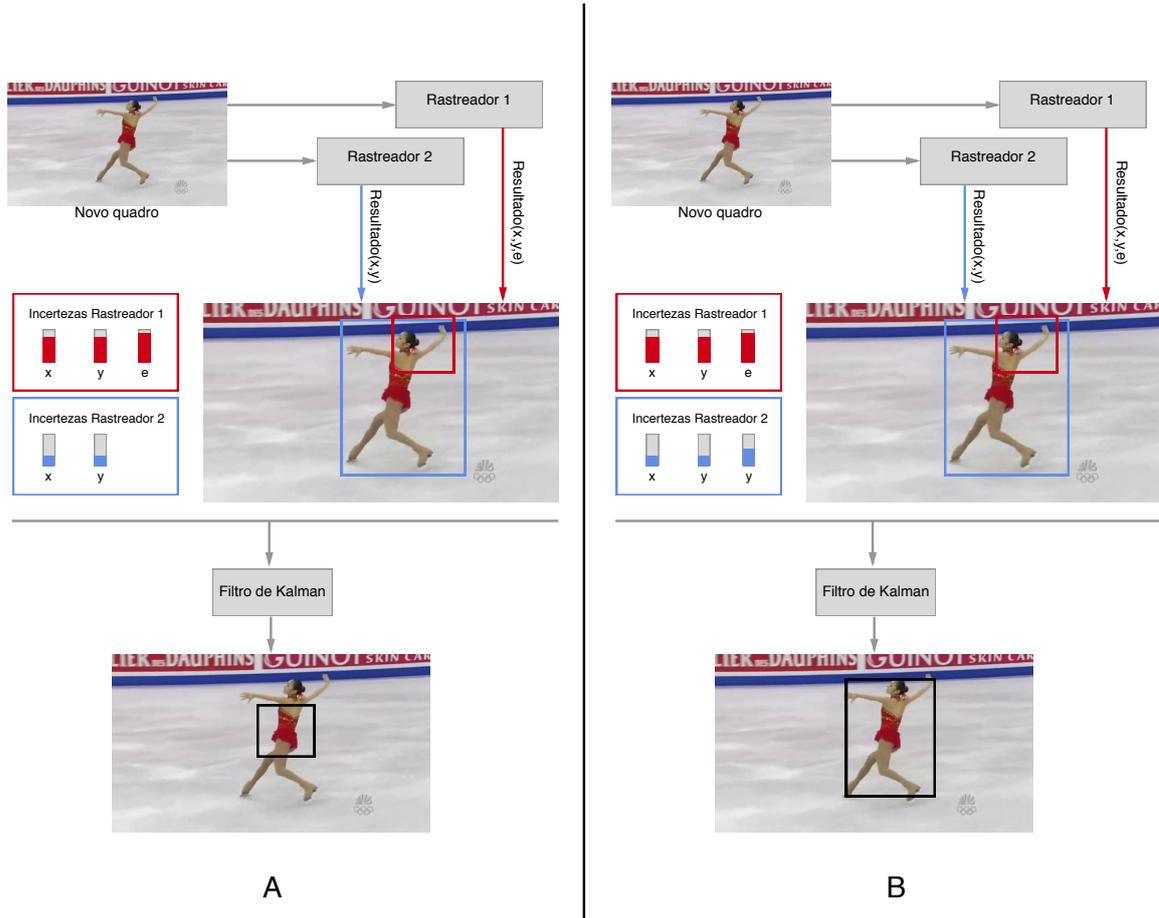


Figura 12: Ilustração do processo de fusão utilizando dois rastreadores, o rastreador 1 que estima a translação e a mudança de escala do objeto de interesse e o rastreador 2 que estima apenas a translação. No caso A, os rastreadores reportam apenas as variáveis que estimam, fazendo com que a estimativa ruim do rastreador 1 afete a escala do resultado final. No caso B, o rastreador 2 reporta a sua escala atual, evitando que o resultado ruim do rastreador 2 possua muita influência sobre o resultado final.

Essa estratégia pode ser utilizada para adicionar mais rastreadores no método. Podem ser incluídos rastreadores que resolvam outros problemas, como rotação do objeto de interesse, ou métodos especialistas que resolvam apenas parte do problema, como o método de estimativa de escala do CMT (NEBEHAY; PFLUGFELDER, 2015).

A incerteza é utilizada na matriz R da equação 3.4. Essa matriz é apresentada

abaixo, onde u_{tnx} é a incerteza do rastreador tn para a variável x .

$$R_t = \begin{bmatrix} u_{t1x} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & u_{t1y} & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & u_{t1e} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u_{tnx} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & u_{tny} & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & u_{tne} \end{bmatrix} \quad (3.10)$$

A localização estimada pelos rastreadores é utilizada na matriz de medição z_t (Eq. 3.5). A matriz de medição é estruturada como mostrado na equação 3.11, onde tn_x é a coordenada x do centro do objeto estimado pelo rastreador n . Para relacionar as variáveis da matriz de medição (z_t) com as do estado do filtro de Kalman (x_t) é utilizada a matriz H (Eq. 3.11).

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} z_t = \begin{bmatrix} t1_x \\ t1_y \\ t1_e \\ \dots \\ tn_x \\ tn_y \\ tn_e \end{bmatrix} \quad (3.11)$$

Os resultados reportados pelos rastreadores são combinados utilizando o processo de correção do filtro de Kalman. A influência de cada rastreador e do resultado previsto pelo filtro de Kalman é definida durante o cálculo do ganho de Kalman (Eq. 3.4), sendo proporcional a incerteza de cada um deles. O resultado final é calculado utilizando a equação 3.5, e a incerteza deste resultado é calculada na equação 3.6.

3.2.3 Método proposto

O método proposto, *Kalman filter ensemble-based tracker* (KFebT), é apresentado no algoritmo 1.

O método começa recebendo o primeiro quadro da sequência o retângulo com a localização do objeto de interesse (linha 2). Em seguida, os rastreadores e o filtro de Kalman são criados e iniciados (linhas 3 a 7).

Para cada quadro na sequência, o processo inicia com o filtro de Kalman realizando sua previsão. Depois, os rastreadores recebem o novo quadro e executam de forma independente seus processos de rastreamento, retornando seus resultados e a confiança

Algoritmo 1: KFebT - *Kalman filter ensemble-based tracker*

```

1 início
2   img, região ← obterPrimeiroQuadro()
3   vetorRastreadores ← criaRastreadores()
4   KF.cria();
5   para cada Rastreador no vetorRastreadores faça
6     | rastreador.inicia(img, região)
7   fim
8   para cada Quadro no sequência faça
9     | KF.previsão()
10    | img ← obterProximoQuadro()
11    | para cada Rastreador no vetorRastreadores faça
12      | rastreador.rastrea(img)
13      | vetorResultados ← rastreador.resultado()
14      | vetorConfiança ← rastreador.confiança()
15    | fim
16    | resultadoFusão ← KF.corrige(vetorResultados, vetorConfiança)
17    | para cada Rastreador no vetorRastreadores faça
18      | rastreador.retroAlimentação(resultadoFusão)
19    | fim
20  | fim
21 fim

```

deles (linhas 12 a 14). Durante a correção, o filtro de Kalman utiliza sua previsão para calcular a penalidade de cada rastreador.

A incerteza do resultado de cada rastreador é calculada utilizando sua confiança e a penalidade (Eq. 3.9), e as matrizes R e z_t são criadas utilizando os resultados dos rastreadores e suas incertezas. Essas matrizes serão utilizadas nas equações do processo de correção do filtro de Kalman para obter o resultado final do rastreamento para aquele quadro.

O último passo do algoritmo é a retroalimentação. Ela é realizada entregando aos rastreadores o resultado obtido após a correção do filtro de Kalman para todos os rastreadores (linhas 17 a 19). Os rastreadores utilizados neste trabalho utilizam esse resultado como área de busca inicial no próximo quadro.

3.2.4 Implementação

O método proposto neste trabalho foi implementado em C++ utilizando as bibliotecas OpenCV¹ e Qt².

A Figura 13 apresenta o diagrama de classes do sistema implementado. Para faci-

¹<http://opencv.org>

²<http://qt.io>

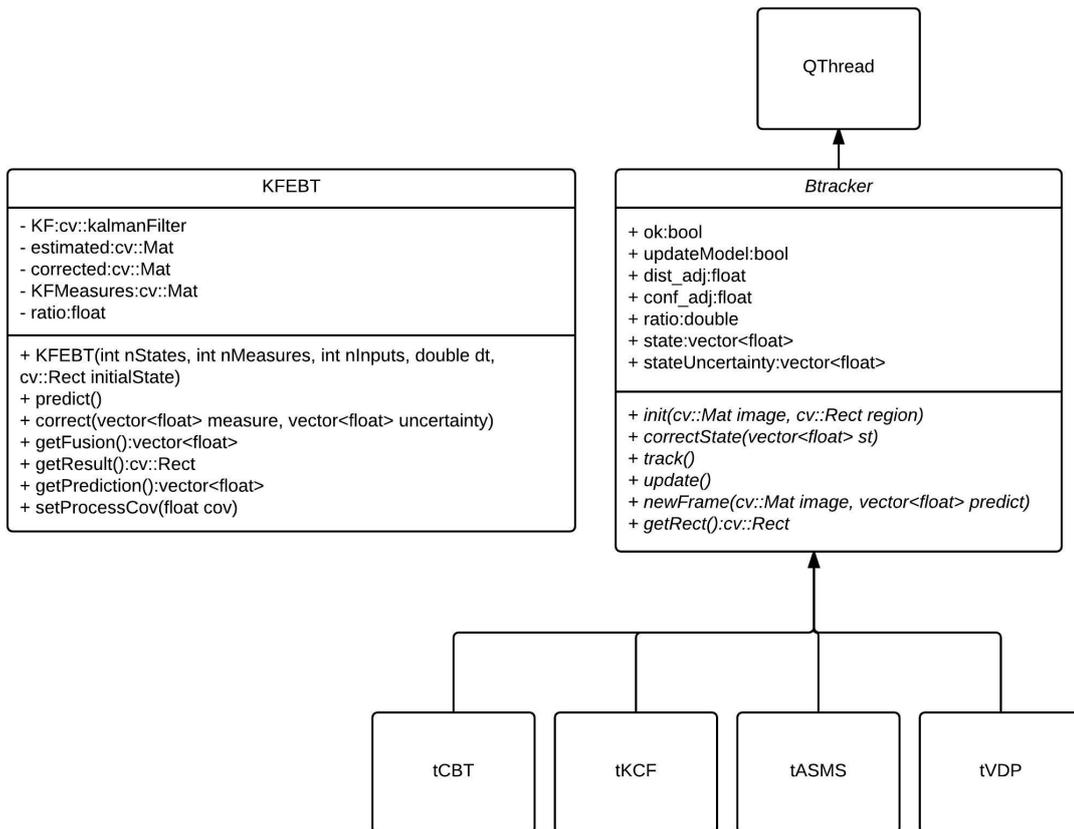


Figura 13: Diagrama de classes do sistema implementado.

litar a integração dos rastreadores com o método proposto, foi criada uma classe abstrata com as funções necessárias para realizar a interface entre o método de fusão e os rastreadores. Essa classe é chamada de *Base Tracker* (Btracker) e deve ser a classe pai das classes dos rastreadores que forem utilizados no sistema. Para ser filho da classe Btracker, as classes dos rastreadores deverão implementar os seguintes métodos:

init: Este método recebe uma imagem e a região onde o objeto de interesse se encontra. Deve ser usado para iniciar ou reiniciar o rastreador.

newFrame: Neste método o rastreador receberá um novo quadro da sequência e a localização do objeto de interesse prevista pelo filtro de Kalman. A previsão deve ser utilizada para calcular a incerteza do rastreador e o quadro deverá ser armazenado para ser utilizado posteriormente.

track: Este método deve realizar o rastreamento no último quadro recebido por *newFrame* e calcular a incerteza do rastreador. O resultado deve ser armazenado no vetor *state* e a incerteza deve ser armazenada no vetor *stateUncertainty*, de onde serão acessados pelo método de fusão. Para facilitar a integração, a adaptação do modelo pode ser realizada nesse método.

update: Caso o método de rastreamento atualize seu modelo em cada quadro, essa atualização deve ser implementada neste método.

getRect: Este método deve retornar a última localização do objeto de interesse estimada pelo rastreador. Esta função é utilizada para depuração e para quando se queira obter resultados visuais dos rastreadores.

correctState: Deve ser utilizado pelo rastreador para receber a retroalimentação do método de fusão e corrigir seu estado atual.

O método proposto é implementado na classe KFEBT do diagrama de classes. Ele implementa os seguintes métodos:

init: Este método recebe a quantidade de variáveis no estado do filtro de Kalman, a quantidade de medidas, a quantidade de variáveis de controle, o intervalo de medições dt e o estado inicial. Com essas informações são criados o modelo de medição H e a matriz de transição A .

predict: Este método realiza a previsão do filtro de Kalman utilizando o estado anterior e a matriz de transição (Eq. 3.2 e 3.3).

correct: Este método recebe os resultados dos rastreadores e suas incertezas e realiza o processo de correção do filtro de Kalman (Eq. 3.4, 3.5 e 3.6).

getFusion: Retorna o resultado da correção para ser utilizado na retroalimentação dos rastreadores.

getResult: Retorna o resultado da correção em formato próprio para ser utilizado no *toolkit* do VOT ou para gerar um resultado visual do rastreamento.

getPrediction: Retorna a previsão do filtro de Kalman.

setProcessCov: Ajusta a incerteza da previsão do filtro de Kalman.

Uma das vantagens de se utilizar rastreadores independentes é que eles podem ser executados de forma concorrente, tirando proveito dos processadores multinúcleos atuais. Para implementar esse paralelismo, foi utilizado a classe `QThread` da biblioteca Qt. A classe `QThread` implementa todo o código necessário para que seja executada em uma nova *thread* sempre que receber o sinal de início. Essa capacidade é herdada pelos filhos da classe `QThread` e é utilizada pelas classes filhas da `Btracker` para chamar as funções *track* e *update*, fazendo com que os rastreadores sejam executados em paralelo.

Para facilitar a implementação, ao invés de utilizar a escala do retângulo como variável de estado, foi utilizado o comprimento do retângulo e a razão entre o comprimento

e a altura é armazenada em todos os objetos dos rastreadores e no objeto KFEBT na variável *ratio*. Todos os métodos utilizados consideram que a razão entre o comprimento e a altura é mantida durante todo o processo de rastreamento.

Para realizar a comunicação entre o rastreador implementado e o *toolkit* do VOT é utilizado o protocolo TraX (ČEHOVIN, 2017). O código, a documentação e exemplos de uso deste protocolo podem ser encontrados no seu repositório oficial³.

A versão atualizada da implementação pode ser encontrada no repositório oficial do projeto⁴.

3.3 Rastreadores utilizados

Nesta seção serão apresentados os rastreadores utilizados para testar o método proposto, destacando as modificações necessárias para integrá-los ao método de fusão. No final é apresentada uma comparação entre esses rastreadores.

3.3.1 ASMS

O primeiro rastreador utilizado foi o ASMS (mais detalhes na Seção 2.3.1). Foi utilizado como base a implementação disponibilizada no repositório do autor do ASMS⁵.

Para integrá-lo ao sistema foram realizadas duas modificações, a primeira com objetivo de melhorar o tempo de execução do método, e a segunda para que o ASMS retorne sua medida de confiança após realizar o rastreamento.

Para melhorar o tempo de execução do código do ASMS obtido, foi utilizada uma *lookup table*, que é uma tabela que armazena valores que são recorrentemente utilizados no decorrer da execução. No método do ASMS, o cálculo do índice dos pontos da imagem no histograma é a operação mais realizada:

$$indice = \frac{vermelho * dim^2}{intervaloPorIndice} + \frac{verde * dim}{intervaloPorIndice} + \frac{azul}{intervaloPorIndice} \quad (3.12)$$

Onde *dim* é a dimensão do histograma para cada canal da imagem (o histograma total possui dimensão dim^3) e *intervaloPorIndice* é a quantidade de valores que são colocados em cada índice para cada canal. Para reduzir a quantidade de cálculos realizados,

³<https://github.com/votchallenge/trax>

⁴<https://github.com/psenna/KF-EBT>

⁵<https://github.com/vojirt/asms>

os valores para os valores das cores foram pré-computados e armazenados em um vetor. O índice passou a ser calculado conforme a equação 3.13.

$$indice = vetorVermelho[vermelho] + vetorVerde[verde] + vetorAzul[azul] \quad (3.13)$$

Reduzindo o cálculo realizado em quatro multiplicações e três divisões. Considerando que esse cálculo é realizado para cada ponto dentro da área de busca do algoritmo em cada uma das iterações, essa redução provoca uma diminuição significativa no tempo de execução do método.

Como medida de confiança é utilizado o coeficiente de Bhattacharyya calculado entre o histograma do objeto de interesse criado como modelo e o histograma da região estimada pelo rastreador no novo quadro. O coeficiente de Bhattacharyya é uma medida que avalia a sobreposição entre duas medidas estatísticas e é calculada da seguinte forma:

$$CB(p, q) = \sum_{i=1}^n \sqrt{p_i q_i} \quad (3.14)$$

onde p e q são duas funções de densidade de probabilidade.

Todos os parâmetros do ASMS foram mantidos os mesmos de (VOJIR; NOSKOVA; MATAS, 2014).

3.3.2 KCF

O segundo rastreador utilizado é o KCF (mais detalhes sobre o método são apresentados na seção 2.3.2). O código utilizado como base para a implementação é uma reimplementação em C++ (codificado originalmente em Matlab) que pode ser encontrado em seu repositório⁶.

Para integrar o método foram realizadas duas modificações, o ajuste da área de busca para o tamanho do filtro de correlação e o retorno da medida de confiança.

Como o KCF recebe retorno do método de fusão, e outros métodos que participam da fusão estimam mudança de escala, o tamanho do retângulo que representa o estado atual pode ser diferente do que foi utilizado para iniciar o filtro do KCF. O KCF utiliza uma janela de tamanho fixo em sua busca, sendo necessário ajustar o estado atual para o tamanho desta janela antes de realizar a operação de correlação. Para isso, foi utilizada uma interpolação linear.

Como medida de confiança foi utilizado o resultado da correlação. Como o KCF não estima a variação da escala do objeto de interesse, a incerteza da variável e do seu resultado é cinco vezes maior que a incerteza do rastreador.

⁶<https://github.com/psenna/kcf>

3.3.3 VDP

Para testar a inclusão de um método especialista, foi utilizado o método de estimativa de escala do CMT (NEBEHAY; PFLUGFELDER, 2015). Neste trabalho este método será chamado de variação de distâncias entre pontos (VDP). O VDP funciona estimando a variação mediana entre a distância de pares de pontos de interesse.

O algoritmo 2 apresenta o VDP. O VDP é iniciado encontrando pontos de interesse no quadro anterior utilizando a região recebida no *feedback* (linha 3). Foi utilizado o *Good Features to Track* (SHI; TOMASI, 1994) como detector de pontos de interesse.

Os pontos encontrados são utilizados no fluxo óptico para serem localizados na imagem atual (linha 4). É utilizado o método de Lucas–Kanade para estimar fluxo óptico (LUCAS; KANADE et al., 1981). Com esses pontos, é realizada a validação para trás, onde é aplicado o fluxo óptico nos pontos encontrados na imagem atual para a imagem anterior (linha 5). É realizada uma verificação da distância entre os pontos iniciais da imagem anterior e os pontos encontrados no fluxo óptico para trás, eliminando os pontos que estiverem a uma distância maior que certo limiar (linha 6).

Com os pares de pontos válidos é realizada a estimativa da mudança de escala. Para isso é encontrada a variação mediana de distância entre os pares de pontos (linha 7). Como o método não estima a translação do objeto de interesse, é utilizada a localização anterior. Após o cálculo da confiança (linha 9) o rastreador reporta seu resultado para ser utilizado na fusão (linha 10) e aguarda o método proposto retornar o resultado final (linha 11).

Algoritmo 2: VDP - Variação de distância entre pontos

Entrada: Posição inicial y_0 e tamanho inicial do objeto e_0

Saída: Escala (y_t, e_t) em cada um dos n quadro, onde $t \in \{1, \dots, n\}$

```

1 início
2   para cada Quadro  $t \in \{1, \dots, n\}$  faça
3      $pontosDeInteresse \leftarrow detectaPontos(imagem_{t-1}, y_{t-1}, e_{t-1})$ 
4      $novosPontos \leftarrow fluxoOptico(imagem_{t-1}, pontosDeInteresse, imagem_t)$ 
5      $pontosAnteriores \leftarrow fluxoOptico(imagem_t, novosPontos, imagem)$ 
6      $pontosValidos \leftarrow validacao(pontosDeInteresse, pontosAnteriores)$ 
7      $e_t \leftarrow estimativaEscala(pontosValidos)$ 
8      $y_t \leftarrow y_{t-1}$ 
9      $confianca \leftarrow pontosValidos/pontosDeInteresse$ 
10    reportaResultadoParaFusao()
11     $y_t, e_t \leftarrow feedback()$ 
12  fim
13 fim
```

Na implementação realizada, foram utilizadas algumas restrições no método. São utilizados no máximo 60 pontos de interesse encontrados pelo detector. Caso sejam encontrados mais de 60, os pontos com melhor pontuação de acordo com o detector serão mantidos. Na validação para trás foi utilizada a distância máxima de 3 pontos para validar os pares de pontos. Para garantir que o método tenha pontos suficientes para sua estimativa, é exigido que sejam encontrados no mínimo 20 pontos pelo detector e que no mínimo metade dos pontos encontrados pelo detector sejam válidos. Caso um desses requisitos não seja atingido, o resultado do método pode ser ruim e é utilizado um valor baixo na confiança do rastreador, de modo que ele tenha pouca influência no resultado final.

O VDP é utilizado para mostrar que o modelo proposto pode obter vantagens combinando o resultados de métodos especialistas em resolver apenas parte do problema.

A implementação foi realizada utilizando como base o código disponibilizado no repositório do autor do CMT⁷.

3.3.4 CBT

O último rastreador utilizado é uma simplificação do CMT (*clustering of static-adaptive correspondences for deformable object tracking*) (NEBEHAY; PFLUGFELDER, 2015), que foi nomeado *Consensus based tracker* (CBT).

Para criar o CBT, foi removida a parte de casamento de pontos de interesses do CMT e o armazenamento da posição relativa entre os pontos de interesse. Ele realiza o rastreamento utilizando apenas o fluxo óptico e o esquema de votação baseado em consenso utilizado no CMT. O algoritmo 3 apresenta o CBT.

O CBT utiliza o quadro anterior e o resultado final recebido na retroalimentação dele para encontrar pontos de interesse (linha 3). Os pontos de interesse são escolhidos utilizando o *Good Features to Track* (SHI; TOMASI, 1994). O fluxo óptico é aplicado sobre esses pontos para que sejam encontrados na imagem atual (linha 4).

Depois é realizada a validação para trás, da mesma forma descrita no VDP (linhas 5 e 6). Os pontos válidos são utilizados para estimar a variação de escala do objeto de interesse (linha 7).

Os pontos válidos são utilizados para estimar a mudança de escala do objeto de interesse (linha 7) e a translação utilizando a estratégia de votação baseada em consenso (linha 8).

Como o CBT não cria um modelo para realizar o rastreamento, é criado um histograma de cores do objeto de interesse para avaliar a confiança do seu resultado. O

⁷<https://github.com/gnebehay/CppMT>

Tabela 1: Comparação entre o resultado do CMT e do CBT.

Rastreador	Sobreposição esperada	Precisão	Falhas
CMT	0.12	0.40	4.09
CBT	0.055	0.40	6.72

histograma é criado quando o rastreador é iniciado. A confiança é calculada multiplicando o coeficiente de Bhattacharyya entre o histograma do objeto de interesse e da região estimada pelo rastreador e a proporção de pontos validados na verificação para trás (linha 9). O resultado do rastreador e sua confiança são entregues para o método de combinação (linha 10).

Algoritmo 3: *CBT - Consensus based tracker*

Entrada: Posição inicial y_0 e tamanho inicial do objeto e_0

Saída: Posição e escala (y_t, e_t) em cada um dos n quadro, onde $t \in \{1, \dots, n\}$

```

1 início
2   para cada Quadro  $t \in \{1, \dots, n\}$  faça
3     pontosDeInteresse  $\leftarrow$  detectaPontos(imagem $_{t-1}$ ,  $y_{t-1}$ ,  $e_{t-1}$ )
4     novosPontos  $\leftarrow$  fluxoOptico(imagem $_{t-1}$ , pontosDeInteresse, imagem $_t$ )
5     pontosAnteriores  $\leftarrow$  fluxoOptico(imagem $_t$ , novosPontos, imagem)
6     pontosValidos  $\leftarrow$  validacao(pontosDeInteresse, pontosAnteriores)
7      $e_t \leftarrow$  estimativaEscala(pontosValidos)
8      $y_t \leftarrow$  votacao(pontosValidos)
9     confianca  $\leftarrow$  histograma( $y_t$ ,  $e_t$ ) * pontosValidos/pontosDeInteresse
10    reportaResultadoParaFusao()
11     $y_t, e_t \leftarrow$  feedback()
12  fim
13 fim
```

As mesmas restrições quanto a quantidade mínima e máxima de pontos utilizada no VDP é utilizada no CBT.

Esse método é sensível a oclusão e pode utilizar pontos não pertencentes ao objeto de interesse caso a janela não esteja bem ajustada ao objeto de interesse no quadro $t - 1$.

O CBT foi utilizado para testar a utilização de um rastreador muito fraco no método proposto. Como pode ser visto na Tabela 1, os resultados do CBT são consideravelmente piores que os do CMT. O ideal é que o método de fusão seja capaz de obter resultados melhores quando foram adicionados mais rastreadores à fusão, mesmo quando adicionados rastreadores ruins.

A implementação foi realizada utilizando como base o código disponibilizado no repositório do autor do CMT⁸.

⁸<https://github.com/gnebehay/CppMT>

Tabela 2: Comparação entre os rastreadores utilizados

Rastreador	Estimativa de translação	Estimativa de mudança de escala	Medida de confiança
ASMS	Mean shift	Mean shift	Coefficiente de Bhattacharyya do histograma de cores
KCF	Correlação	Nenhum	Resultado da correlação
CBT	Consenso	Varição da distância entre os pares de pontos	Coefficiente de Bhattacharyya do histograma de cores e quantidade de pontos encontrados na imagem seguinte pelo fluxo óptico
VDP	Nenhum	Varição da distância entre os pares de pontos	Quantidade de pontos encontrados na imagem seguinte pelo fluxo óptico

3.3.5 Comparação entre os rastreadores

Neste trabalho foram utilizados três rastreadores e um método especialista. Nesta seção será apresentada uma breve comparação entre esses métodos.

A Tabela 2 apresenta a comparação entre os métodos de estimativa de translação, estimativa de mudança de escala e a medida de confiança de cada um dos rastreadores.

Quanto à estimativa de translação, o ASMS utiliza o algoritmo *mean shift*, o KCF utiliza um filtro de correlação e o CBT utiliza uma estratégia de votação baseada em consenso. O VDP não estima a translação do objeto de interesse.

Para estimar a mudança de escala do objeto de interesse, o ASMS utiliza o *mean shift* e uma penalidade para evitar que a escala seja alterada de forma exagerada. O VDP e o CBT estimam a mudança de escala encontrando a variação mediana da distância entre os pares de pontos de interesses. O KCF não estima a variação da mudança de escala.

Segundo [Vojir, Matas e Noskova \(2015\)](#), é interessante utilizar rastreadores que utilizem estratégias diferentes na combinação. Segundo ele, a utilização de rastreadores muito semelhantes pode fazer com que o resultado final da combinação possua as mesmas limitações que os rastreadores utilizados. Desta forma, os rastreadores selecionados para serem utilizados neste trabalho possuem estratégias diferentes para resolver o problema, sendo utilizados um filtro de correlação (KCF), uma técnica baseada em *mean shift* (ASMS) e um rastreador baseado no rastreamento de partes que utiliza pontos de interesse e fluxo óptico (CBT).

4 Desafio VOT

Neste capítulo será apresentado o desafio VOT (*Visual Object Tracking*), expondo um breve histórico, suas métricas de avaliação de rastreadores, metodologia de teste e conjunto de ferramentas (*toolkit*) utilizado para realizar a avaliação dos rastreadores.

4.1 História

O desafio VOT têm como objetivo estabelecer uma metodologia de testes e comparações entre métodos de rastreamento visual de objetos de curto prazo. Para isso, foram estabelecidas uma base de dados, medidas de avaliação de performance e uma ferramenta para a realização dos testes e comparação de resultados. Também foi criada uma plataforma para a discussões relacionadas a avaliação de rastreadores¹.

A cada ano é realizada uma edição do desafio VOT, sendo disponibilizada a base de dados, a ferramenta para a realização dos testes e as instruções para a integração de rastreadores à ferramenta. Qualquer pessoa que tenha desenvolvido um novo rastreador ou proposto melhorias em algum previamente existente pode participar do desafio.

A primeira edição do VOT foi realizada em 2013 (KRISTAN et al., 2013), junto com a ICCV 2013. Foram realizados três experimentos, no primeiro é executado o teste padrão. No segundo teste foi adicionado ruído durante a iniciação dos rastreadores. O terceiro teste foi realizado da mesma forma que o primeiro, mas as imagens foram convertidas de coloridas para escala de cinza. A base de dados utilizada nos testes continha dezesseis sequências anotadas utilizando retângulos, como mostrado na Figura 14. Foram enviados 19 rastreadores e o comitê do desafio realizou o teste em mais 8 rastreadores, tendo um total de 27 rastreadores testados. Os rastreadores foram avaliados utilizando sua sobreposição e a quantidade de falhas. Os rastreadores foram ranqueados utilizando a média do ranque dos rastreador nos três experimentos.

No ano de 2014 foi realizada uma nova edição do VOT (KRISTAN et al., 2014) em conjunto com o ECCV 2014. A diferença da edição de 2014 para a de 2013 foi a nova base de dados e a remoção do teste utilizando imagens em escala de cinza. A nova base de dados continha 25 sequências anotadas com retângulos rotacionados(Fig. 14). Foram submetidos 33 rastreadores para o VOT2014 e o comitê incluiu mais cinco rastreadores, tendo um total de 38 rastreadores comparados nessa edição.

Em 2015 foi realizada mais uma edição do VOT (KRISTAN et al., 2015) em conjunto com o ICCV 2015. Nessa edição foi realizado apenas um experimento, o experimento

¹<http://votchallenge.net>

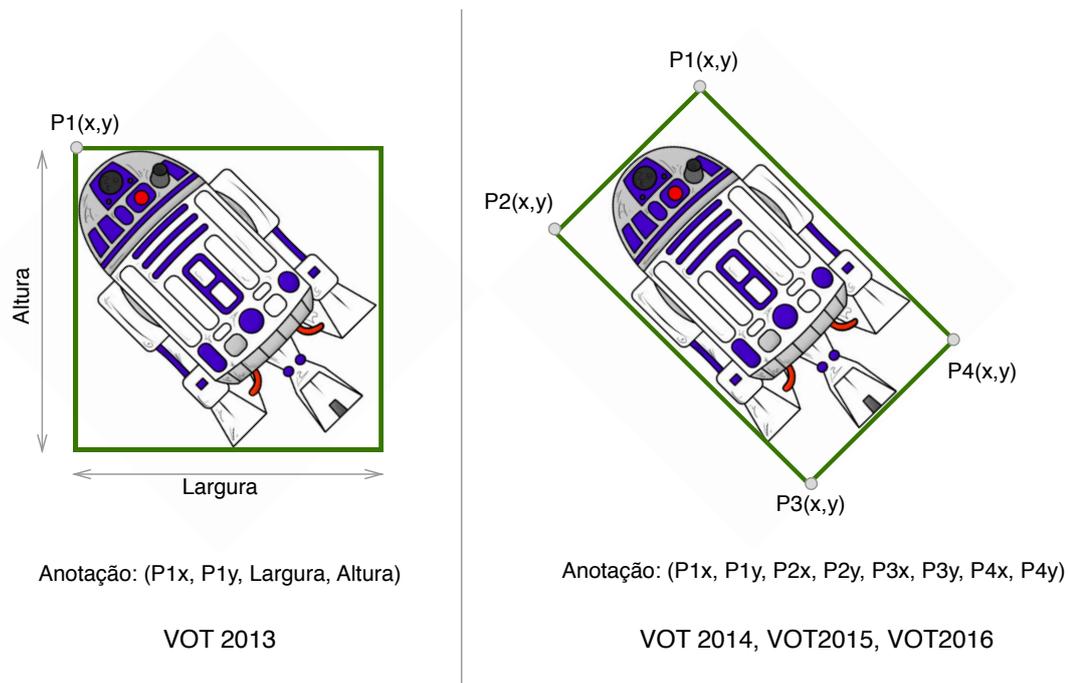


Figura 14: Forma de anotações utilizadas no desafio VOT. No ano de 2013 foi utilizado um retângulo alinhado com os eixos da imagem. No ano de 2014 em diante foi utilizado um retângulo rotacionado como anotação.

com adição de ruído na iniciação foi removido. Foi montada uma base de dados mais extensa para ser utilizada no VOT2015, contendo um total de 60 sequências anotadas com retângulos rotacionados. Nessa edição foram submetidos 41 rastreadores e foram adicionados mais 21 rastreadores pelo comitê do VOT, totalizando 62 rastreadores comparados. Os rastreadores passaram a ser ranqueados utilizando a expectativa de sobreposição.

A última edição do VOT, até a finalização deste trabalho, foi realizada em 2016 (KRISTAN et al., 2016), em conjunto com o ECCV 2016. Nela foi adicionado mais um experimento onde os rastreadores não eram reiniciados quando perdessem o objeto de interesse, chamado de teste sem supervisão. Foram utilizadas as mesmas sequências utilizadas no VOT2015, mas as anotações dessas sequências foram refeitas. Foram submetidos 48 rastreadores e mais 22 rastreadores foram adicionados pelo comitê do VOT, totalizando 70 rastreadores comparados.

Com o passar dos anos o VOT obteve um crescimento no número de rastreadores submetidos, de 19 em 2013 para 48 em 2016, na quantidade de sequências utilizadas, de 16 sequências em 2013 para 60 em 2016, e melhorias na sua metodologia de avaliação de rastreadores. Por esses motivos, os resultados obtidos nos desafios são capazes de apontar os rastreadores que estão entre o estado da arte. Nos relatórios do VOT2016 foi questionado que alguns trabalhos apresentados em grandes conferências (CVPR, ECCV, ICCV) nos anos de 2015 e 2016 apontavam técnicas como o MIL (BABENKO; YANG; BELONGIE, 2011) e o IVT (ROSS et al., 2008) como estado da arte na comparação de

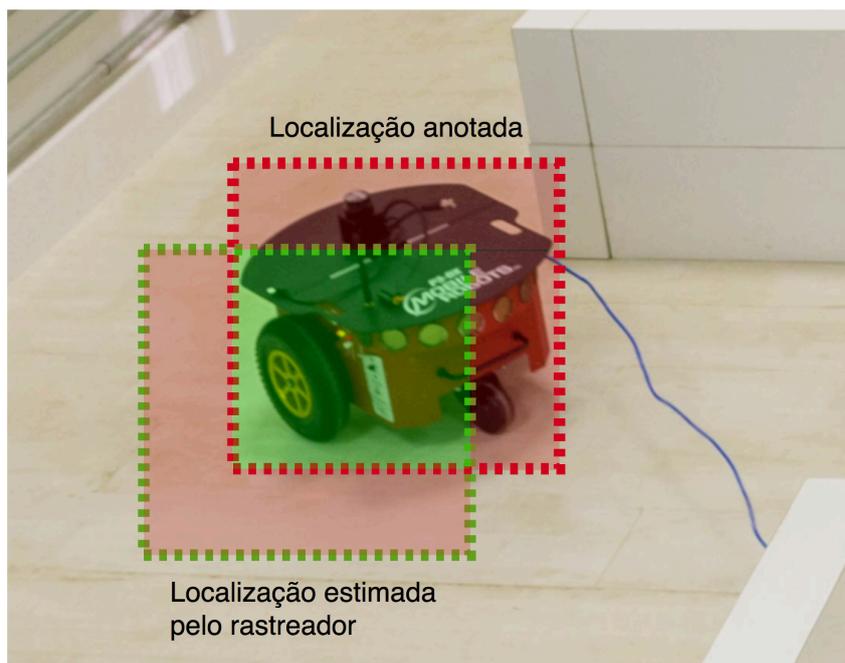


Figura 15: Exemplo de como é realizado o cálculo da sobreposição, onde o retângulo vermelho é a anotação do objeto de interesse e o retângulo verde é o resultado de um rastreador. A sobreposição é calculada dividindo a área verde pela soma das áreas vermelhas e verde.

seus resultados, mas esse rastreadores se encontram nas posições 49 e 63 respectivamente dentre os 70 rastreadores comparados no VOT2016, tendo resultados muito inferiores se comparado com os primeiros.

4.2 Métricas de avaliação de rastreadores

Existem diversas trabalhos na literatura que abordam a avaliação do desempenho de rastreadores (WU; LIM; YANG, 2013; SMEULDERS et al., 2014). Čehovin, Leonardis e Kristan (2015) avaliaram dezesseis medidas de desempenho diferentes, encontrando duas medidas que possuíam baixa correlação e complementares, a precisão e a robustez. A precisão mede a sobreposição entre a região reportada pelo rastreador e a região anotada com a localização do objeto. A robustez mede quantas vezes o rastreador perdeu o objeto de interesse durante o rastreamento.

A sobreposição é a intersecção da região reportada pelo rastreador com a região anotada dividida pela união dessas áreas. A Figura 15 apresenta uma ilustração de como a precisão é calculada. Devido ao processo de reinicialização, a precisão começa a ser calculada somente dez quadros após a inicialização do rastreador, reduzindo a vantagem no cálculo da precisão de rastreadores que possuem muitas falhas.

Já as falhas são calculadas contando a quantidade de vezes que o rastreador é reiniciado. O rastreador é reiniciado toda vez que a sobreposição no quadro for igual ou inferior a um certo limiar. A Figura 16 mostra um exemplo do processo de execução do

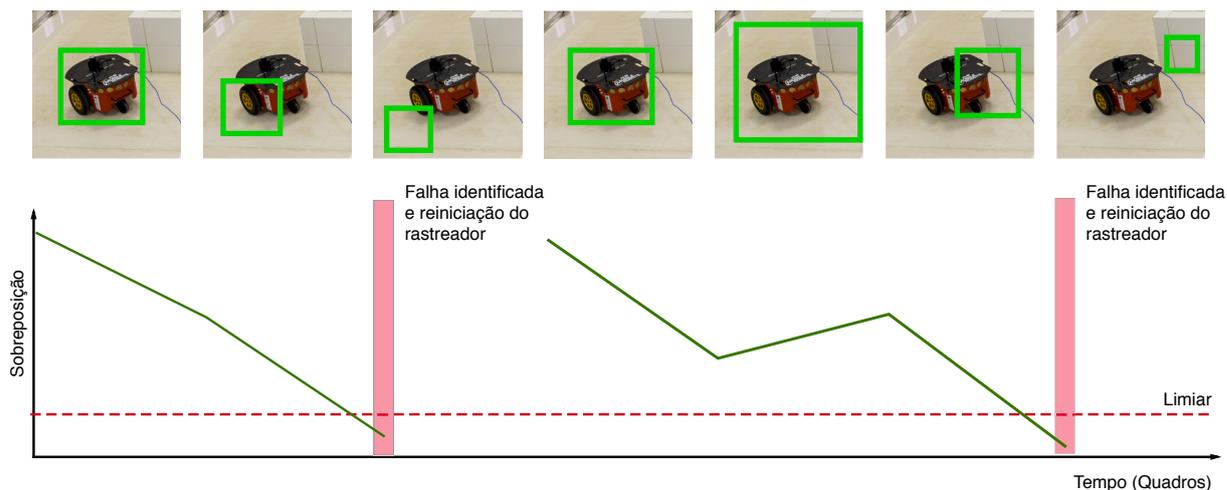


Figura 16: Exemplo do funcionamento do sistema de detecção de falhas do VOT. Toda vez que a sobreposição entre a região reportada pelo rastreador e a região anotada for igual ou menor que um certo limiar, o rastreador é reiniciado e uma falha é anotada.

rastreador, onde o rastreador é reiniciado toda vez que a sobreposição em determinado quadro é igual ou inferior a um certo limiar. No teste do VOT, esse limiar é zero, ou seja, o rastreador só é reiniciado quando não há sobreposição entre as duas regiões.

Essas duas medidas de desempenho, precisão e robustez, permitem uma boa avaliação do comportamento do rastreador, entretanto elas precisam ser analisadas em conjunto. Em casos extremos, um rastreador que possui um alto nível de falhas será constantemente reiniciado e pode obter uma precisão alta. De outra forma, um rastreador que reporta a imagem inteira nunca irá perder o objeto de interesse, mas irá obter uma sobreposição baixa. Nenhuma dessas situações é interessante, um bom rastreador precisa equilibrar essas duas características, falhar pouco ao mesmo tempo que mantém uma alta sobreposição dos seus resultados com a região onde o objeto de interesse está.

A utilização de duas medidas diferentes também dificulta a comparação direta do desempenho de dois rastreadores, principalmente quando cada um deles é melhor em uma delas. Desta forma, também é utilizada outra medida empregada no VOT2015 (KRISTAN et al., 2015) chamada sobreposição esperada. Ela representa a sobreposição média esperada em uma sequência com n quadros.

Para se calcular a sobreposição esperada são utilizadas as sequências virtuais, que consistem em partes de sequências que vão de uma iniciação ou reiniciação até uma falha ou o fim da sequência. Na Figura 17 é apresentado o processo de obtenção das sequências virtuais.

A curva de sobreposição de um rastreador representa a sobreposição esperada para o rastreador após ser executado em n quadros. Ela é obtida calculando a média das sequências virtuais. O valor de sobreposição esperado apresentado nos resultados é a média dos valores da curva de sobreposição entre os valores 108 e 371 (esse intervalo é o tamanho

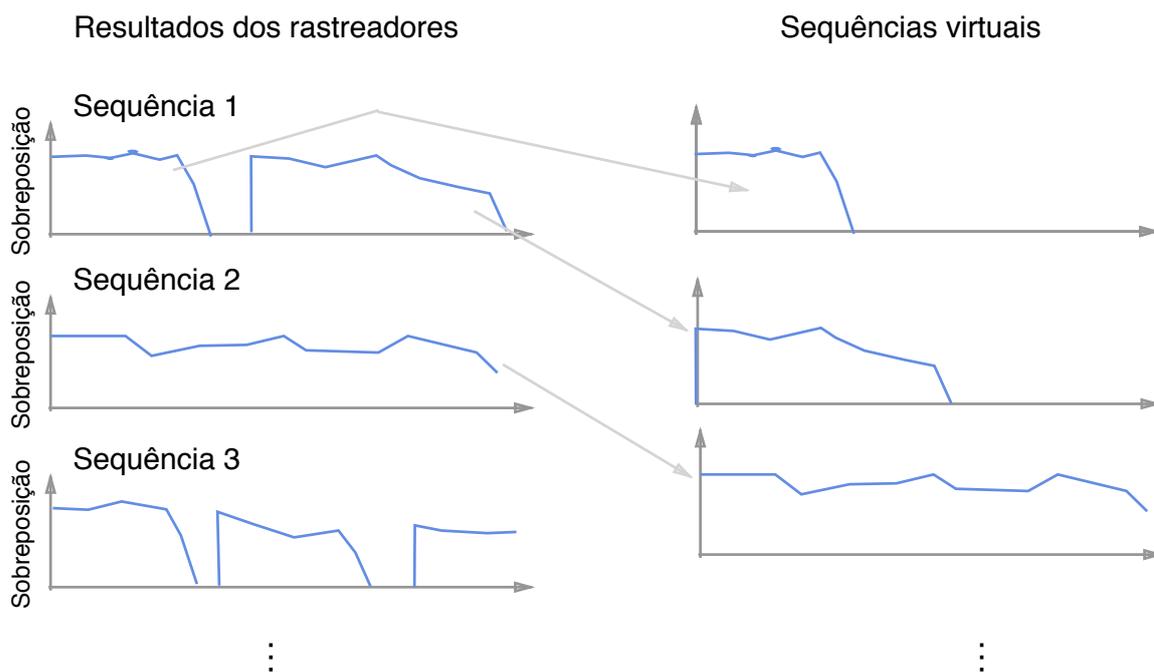


Figura 17: Obtenção de sequências virtuais utilizando os resultados dos rastreadores.

típico das sequência calculado estimado em (KRISTAN et al., 2015)). A Figura 18 ilustra o processo de cálculo da curva de sobreposição e da sobreposição esperada.

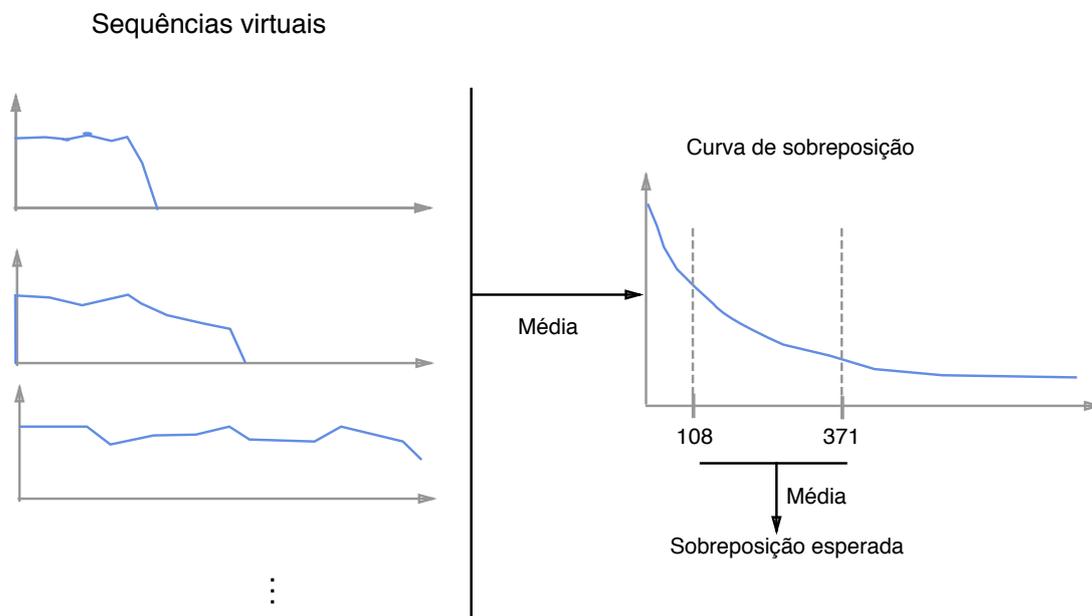


Figura 18: A curva de sobreposição é obtida calculando a média da sobreposição das sequências virtuais. A média dos valores da curva de sobreposição entre os valores 108 e 371 é a sobreposição esperada.

O desafio VOT possui sua própria unidade de medida para avaliar a velocidade do rastreador chamada *Equivalent Filter Operations* (EFO), a qual é computada de forma relativa com uma operação de filtro de máximo em uma imagem de 600x600 pontos utilizando uma janela de 30x30. O tempo utilizado pelo rastreador é dividido pelo tempo

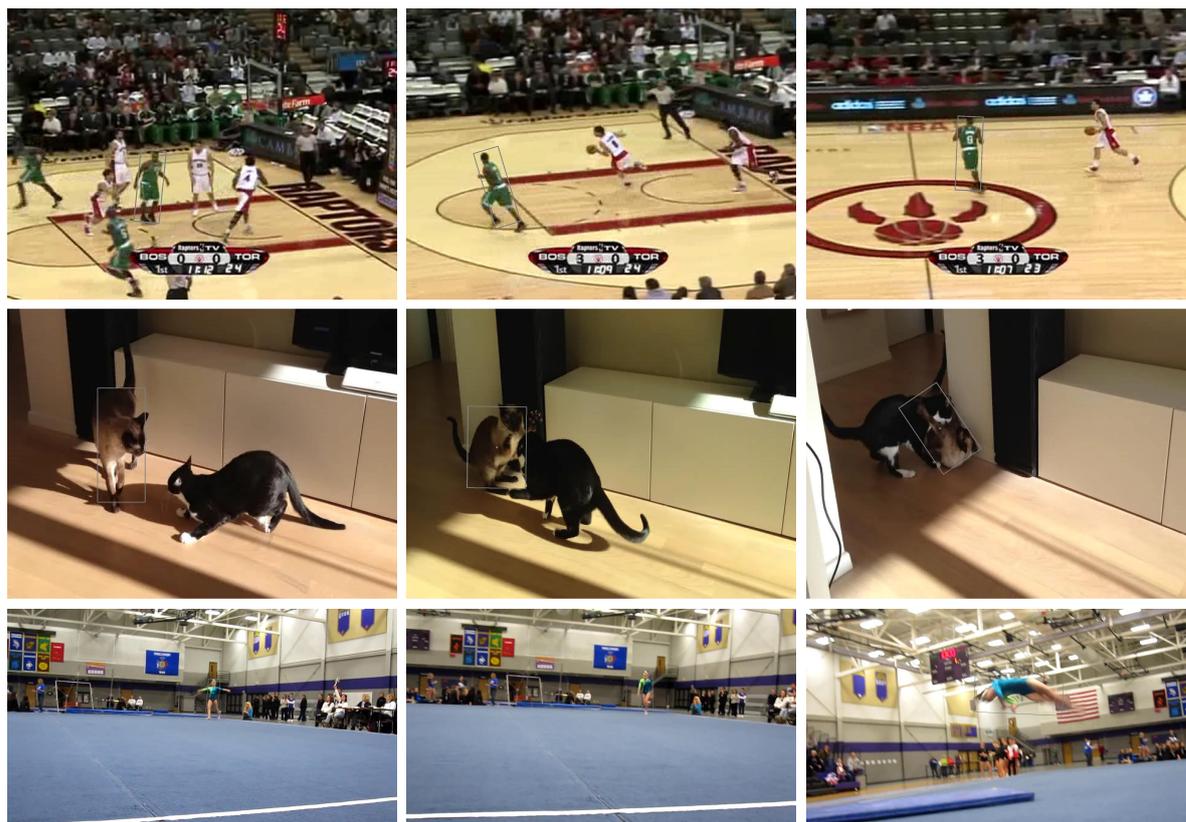


Figura 19: Exemplo de imagens contidas na base de dados VOT2015. Os retângulos cinza são as anotações dessas imagens. São apresentadas imagens das sequências *basketball*, *fernando* e *gymnastics2*.

de realizar essa operação de filtro no mesmo computador. Essa medida é menos sensível às características da máquina onde o rastreador foi executado do que a utilização da medida de quadros por segundo. No relatório do VOT2015 (KRISTAN et al., 2015) é estabelecido um limiar de 20 EFO para que o rastreador seja considerado de tempo real.

4.3 Bases de dados

A base de dados utilizada neste trabalho para realização dos testes foi a do VOT2015. Ela é composta por 60 sequências de vídeos selecionados dentre 443 obtidas das bases de dados OTB (WU; LIM; YANG, 2013), ALOV (SMEULDERS et al., 2014), PTR (VOJIR; NOSKOVA; MATAS, 2013), entre outras fontes. Depois da remoção das sequências duplicadas, em escala de cinza e contendo objetos com área menor que 400 pontos, restando 356 sequências.

Essas sequências foram selecionadas de forma automática para obter um conjunto rico em cinco atributos visuais (oclusão, mudança na iluminação, mudança de trajetória, variação de tamanho e movimentação da câmera), mas consideravelmente pequeno, de modo a manter o tempo de teste razoavelmente baixo. Na Figura 19 são apresentados alguns exemplos de sequências contidas na base de dados.

As sequências da base de dados do VOT2015 possuem de 41 a 1500 imagens com tamanho variando entre 320x180 e 1280x720 pontos. Devido à grande diferença no comprimento das sequências, os resultados gerais apresentados são a média ponderada dos resultados obtidos em cada sequência utilizando como peso o comprimento das sequências.

A base contém anotações na forma de retângulos rotacionados que são representados pela posição dos quatro cantos do retângulo, como mostrado na Figura 14. Cada quadro também contém anotações sobre a ocorrência de algum dos cinco atributos visuais, permitindo avaliar o comportamento dos rastreadores na presença deles.

4.4 Metodologia de teste

Nesta seção são apresentadas as três metodologias de teste do VOT utilizadas neste trabalho: teste padrão, teste com ruído, e teste sem supervisão.

4.4.1 Teste padrão

O teste padrão é uma metodologia padrão utilizada em todas as edições do VOT. Nela o rastreador é iniciado com a anotação do objeto de interesse no primeiro quadro da sequência. No segundo quadro em diante o rastreador deve reportar a localização do objeto de interesse estimada. O teste é executado até que o rastreador perca o objeto de interesse.

Quando o rastreador perde o objeto de interesse, caso a região reportada pelo rastreador não tenha sobreposição com a região anotada com a localização do objeto, o rastreador será reiniciado cinco quadros depois. Quando um rastreador é reiniciado, ele não pode armazenar nenhuma informação anterior a sua reiniciação. Para o rastreador, é como se o quadro em que foi reiniciado fosse o primeiro quadro de uma nova sequência.

Os testes são repetidos três vezes para cada sequência, caso o rastreador reporte resultados diferentes para alguma sequência, os testes dessa sequência são repetidos até se obterem quinze testes no total. Esse processo é realizado para economizar tempo, pois se o rastreador obteve o mesmo resultado em todos os quadros de uma sequência três vezes seguidas é provavelmente estocástico, não sendo necessário repetir o teste quinze vezes.

Os rastreadores são avaliados pela expectativa de sobreposição, sobreposição, robustez e velocidade. Os rastreadores são ranqueados pela sua expectativa de sobreposição.

4.4.2 Teste com ruído

A segunda metodologia utilizada é o teste com ruído. Ela foi utilizada originalmente no VOT2014 e não foi mais utilizada nas edições seguintes. Esse teste avalia o

comportamento do rastreador quando iniciado com ruído. Nesse teste, o ruído é adicionado na anotação.

O ruído é adicionado na escala, na posição e na rotação da anotação. Na escala e na posição são adicionados ruído de no máximo $\pm 10\%$ das dimensões da anotação. Na rotação são adicionados até ± 0.1 radianos de ruído. O valor do ruído é definido dentro dos intervalos citados por uma função aleatória uniformemente distribuída.

Devido ao método de iniciação estocástico, os testes com ruído são realizados 15 vezes para cada sequência. Desta forma são obtidos resultados estatisticamente mais confiáveis.

Este teste segue a mesma estratégia de reiniciação do teste padrão, mas a reiniciação também é realizada com uma anotação com ruído.

Neste teste são avaliados a expectativa de sobreposição, a sobreposição, a robustez e a velocidade dos rastreadores. Os rastreadores são ordenados pela expectativa de sobreposição.

4.4.3 Teste sem supervisão

No teste sem supervisão não é realizada a verificação do resultado do rastreador, caso o rastreador perca o objeto de interesse ele não é reiniciado. Esse teste foi proposto no VOT2016 e avalia o rastreador em uma situação onde ele não é supervisionado por outro sistema que indica se ele perdeu o objeto de interesse e o reinicia.

O teste é executado apenas uma vez para cada sequência. São avaliadas a expectativa de sobreposição, a sobreposição e a velocidade dos rastreadores.

4.5 VOT *toolkit*

O desafio VOT disponibiliza um conjunto de ferramentas (*toolkit*) de avaliação. Essas ferramentas são responsáveis por todo o processo de testes e avaliação dos rastreadores, desde o armazenamento das respostas dos rastreadores e reiniciar o rastreador até a geração dos relatórios com os resultados. O conjunto de ferramentas foi implementado na linguagem do Matlab e pode ser executado no próprio Matlab ou no programa GNU Octave.

O *toolkit* do VOT é multiplataforma e suporta rastreadores desenvolvidos em C/C++, Matlab ou Python. A comunicação entre o rastreador e o *toolkit* é realizada utilizando o protocolo TraX (ČEHOVIN, 2017).

Na Figura 20 são apresentadas as interações entre o *toolkit* do VOT e um rastreador. O rastreador não possui acesso direto à base de dados, o *toolkit* é responsável

por entregar ao rastreador as informações necessárias para sua execução. As principais mensagens enviadas para o rastreador são um comando de reiniciação, onde são enviados uma imagem e a localização do objeto de interesse, e uma mensagem contendo o próximo quadro da sequência. As únicas informações que os rastreadores enviam para o *toolkit* são seus resultados.

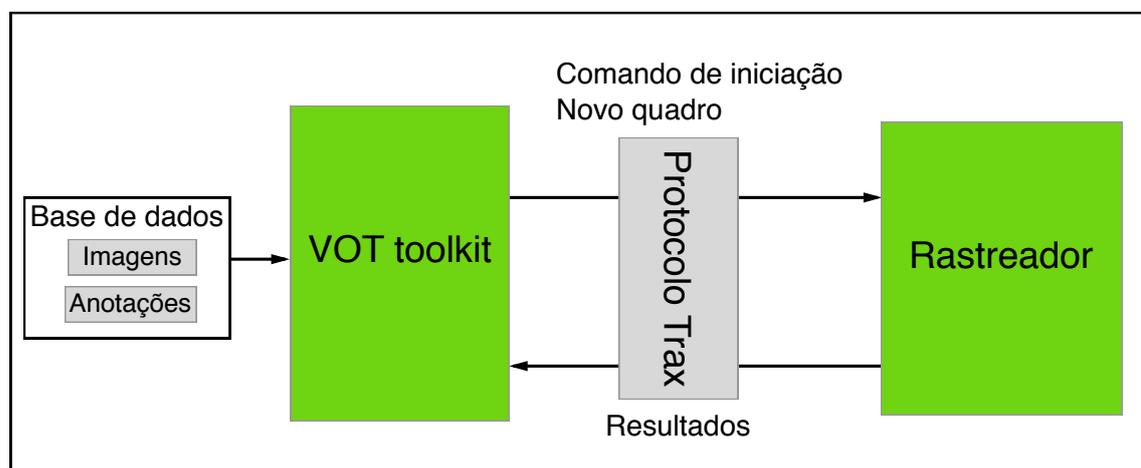


Figura 20: Interações entre o *toolkit* do VOT e os rastreadores.

Instruções de como integrar um rastreador no *toolkit* estão disponíveis no site do VOT².

²<http://www.votchallenge.net/howto/integration.html>

5 Resultados Experimentais

Neste capítulo são apresentados os resultados experimentais utilizados para validar o método proposto. Uma análise comparativa do método proposto com as técnicas utilizadas nas combinações e com as técnicas submetidas ao desafio VOT2015 também é realizada.

5.1 Ambiente de teste

Os testes foram realizados utilizando o kit de ferramentas disponibilizado pelo desafio VOT. Todos os testes foram executados em um computador móvel com as seguintes especificações:

- Intel Core i7 5500u 15W
- Nvidia Gforce 830m
- 8 GB de ram
- disco de estado sólido de 250 GB

No computador estavam instaladas as seguintes versões dos programas utilizados na implementação do método e na execução dos testes:

- Ubuntu Linux 16.04 kernel 4.4 amd64
- Qt 5.8
- gcc 4.8.5
- OpenCV 3.1.0
- Octave 3.8.2
- VOT *toolkit* versão 4.0

Todos os parâmetros dos rastreadores utilizados foram mantidos constantes durante todos os testes, somente foram testados os parâmetros diretamente ligados com o processo de fusão dos rastreadores.

Tabela 3: Teste das combinações de rastreadores, onde A representa o ASMS, C o CBT, K o KCF, e V o VDP.

Combinação	Sobreposição esperada	Precisão	Falhas	QPS
ACK	0,2612	0,5285	1,3643	53
AKV	0,2607	0,5241	1,3615	59
AK	0,2478	0,5236	1,3794	109
AC	0,2289	0,5128	1,7590	60
AV	0,2260	0,5134	1,7333	70
A	0,2147	0,5150	1,8973	188
K	0,1482	0,4333	3,1418	122
CK	0,1473	0,4282	3,1755	51
KV	0,1389	0,4297	3,3456	67
C	0,0508	0,3111	10,9783	89

5.2 Testes no modelo

Nesta seção serão apresentados os testes realizados para avaliar a melhor estrutura para o método proposto. Serão avaliados os impactos da combinação de rastreadores, da utilização da penalidade de movimento, da retroalimentação e da adaptação dos rastreadores.

5.2.1 Combinação de rastreadores

Estes testes visam encontrar as combinações de rastreadores mais interessantes para serem utilizadas nos próximos testes. A utilização de todas as combinações possíveis em todos os testes é proibitiva pelo tempo necessário para sua realização.

Neste teste foram avaliadas a sobreposição esperada do rastreador, sua precisão, sua velocidade em quadros por segundos (QPS) e a quantidade de falhas. Os resultados apresentados são as médias ponderadas dos resultados dos rastreadores para as sessenta sequências da base de dados do VOT2015.

Comparando os resultados das combinações dos rastreadores utilizando o KFebT (Tabela 3) com o dos rastreadores utilizados como base (Tabela 4), pode-se perceber que todas as combinações que utilizavam o ASMS obtiveram resultados melhores que os do ASMS.

Na maioria das vezes, os resultados obtidos com a inclusão foram melhores. Por exemplo, as combinações ACK e AKV obtiveram resultados melhores do que as combinações AK, AC e AV considerando sua sobreposição esperada. Para a realização desse teste, não foi realizado nenhum teste nos parâmetros utilizados na combinação, logo os resultados poderiam ser melhores se fosse realizada uma avaliação dos parâmetros.

A utilização de apenas um rastreador não é o objetivo deste trabalho, entretanto, nos testes realizados, o ASMS e o KCF obtiveram melhoras nas suas sobreposições esperadas de 1% e 8,7%, respectivamente, quando utilizados desacompanhados na fusão.

Tabela 4: Resultado dos rastreadores utilizados como base.

Rastreador	Sobreposição esperada	Precisão	Falhas	QPS
ASMS	0,2125	0,5112	1,9736	198
KCF	0,1364	0,4397	3,5252	130
CBT	0,0548	0,4000	6,72	96

A utilização do filtro de Kalman não diminuiu de forma significativa a velocidade dos rastreadores, diminuindo no máximo em 10 quadros por segundo suas velocidades, sendo uma opção para aumentar a sua robustez com baixo custo computacional.

A inclusão do CBT no KFebT-ACK contribuiu com uma melhora no resultado se comparado com a combinação KFebT-AK. Considerando que o CBT possui resultados consideravelmente inferiores ao ASMS e ao KCF (Tabela 4), o método proposto foi capaz de realizar a fusão do CBT sem piorar o resultado final.

A utilização do método VDP também ajudou a melhorar o resultado, comparando a combinação KFebT-AKV com a KFebT-AK. Esse resultado mostrou que o método proposto é capaz de combinar métodos especialistas. Essa é uma possibilidade interessante, pois é possível utilizar esses métodos na combinação sem a necessidade de grandes alterações nos códigos dos rastreadores ou arranjos específicos para integrá-los com os rastreadores.

Avaliando os resultados obtidos nesses testes, foram escolhidas as combinações ACK, AKV e AK para serem utilizadas nos demais testes.

5.2.2 Análise de parâmetros

Nesta seção são avaliados os parâmetros envolvidos na fusão dos rastreadores. Os parâmetros utilizados na fusão estão relacionados com a incerteza dos rastreadores ou do processo de previsão do filtro de Kalman.

Como os valores das confianças dos rastreadores podem possuir significados diferentes, é utilizado um fator para escalar os valores das confianças e para ponderar a influência de cada rastreador no resultado final, o termo α na Eq. 3.9. Rastreadores com melhores resultados globais podem exercer mais influência no resultado final para um mesmo valor de confiança utilizando um valor de α maior de que rastreadores com resultados piores.

Outro parâmetro relevante é o multiplicador da penalidade de movimento. Ele é um parâmetro utilizado para ponderar a quantidade de penalidade de movimento aplicada à incerteza dos rastreadores. Essa variável é o termo β na Eq. 3.9. Para todos os rastreadores foi utilizado o mesmo valor para este parâmetro.

Por último foi avaliada a influência da incerteza do processo de previsão do próximo

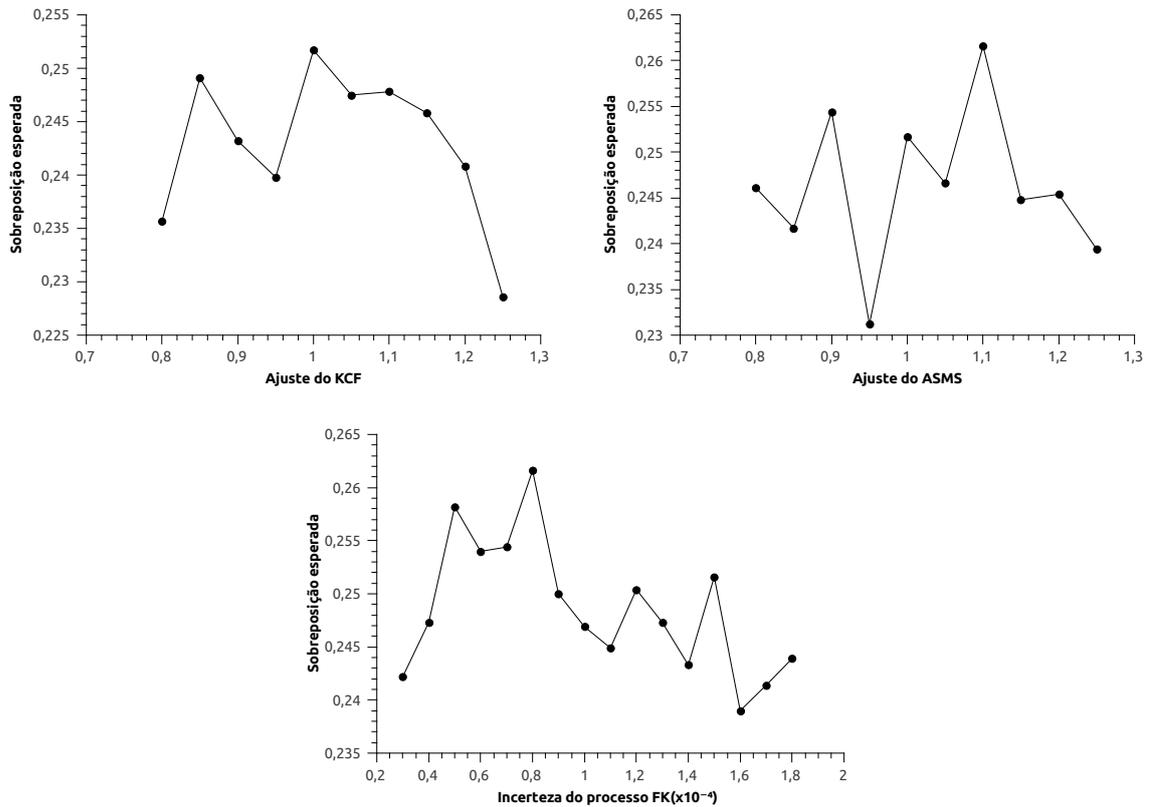


Figura 21: Testes de parâmetros da combinação KFEBT-AK. O primeiro gráfico mostra os resultados da variação do ajuste α do KCF. O segundo gráfico mostra os resultados da variação do ajuste α do ASMS e último gráfico mostra o resultado da variação da incerteza do processo do FK.

estado realizado pelo filtro de Kalman. Essa incerteza é representada pelo termo Q na Eq. 3.3 e é uma matriz diagonal onde todos os valores da diagonal são iguais. Quanto menor este valor, mais influência a previsão do filtro de Kalman terá no resultado final da fusão.

Esta avaliação de parâmetros foi realizada com as três combinações (AKV, ACK e AK) variando um dos parâmetros e fixando os demais. Foi utilizada apenas a sobreposição esperada para realizar a análise e foram escolhidos os parâmetros que a maximizassem, sem avaliar a quantidade de falhas e a sobreposição.

A primeira combinação testada foi a AK. O ajuste α do KCF foi variado de 0,8 a 1,25. O melhor resultado foi obtido com o valor de α igual a 1. Depois foi avaliado o ajuste α do rastreador ASMS. Ele também foi variado de 0,8 até 1,25. O melhor resultado foi obtido com o valor 1,1. Por último foi testado a incerteza do processo de previsão do FK. O melhor resultado foi obtido utilizando o valor $0,8 \times 10^{-4}$. Os gráficos com os valores obtidos nesses testes podem ser vistos na Figura 21.

O mesmo processo de teste de parâmetros foi realizado com a combinação AKV. Para essa combinação foram testados os parâmetros α do CBT, α do KCF, α do ASMS e a incerteza do processo do FK. Para esse parâmetros foram encontrados os valores 0,45,

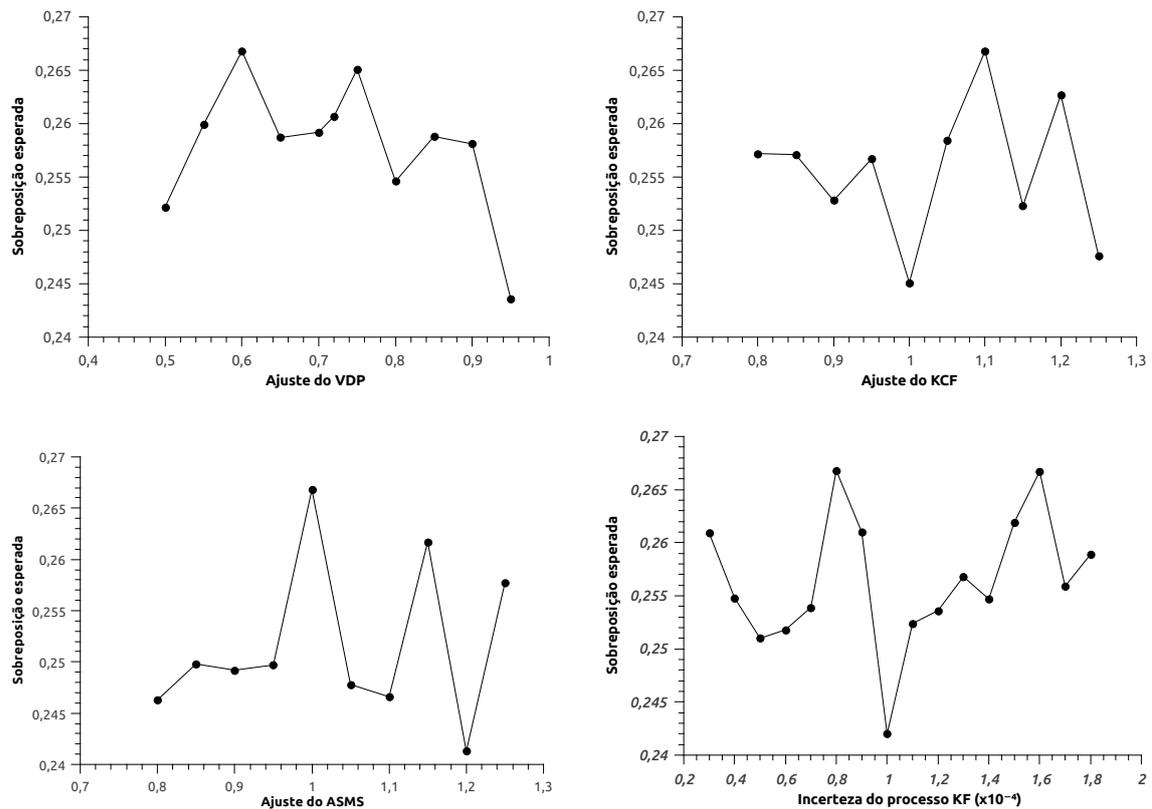


Figura 22: Testes de parâmetros da combinação KFebT-AKV. O primeiro gráfico mostra os resultados da variação do ajuste α do VDP. O segundo gráfico mostra os resultados da variação do ajuste α do KCF. O terceiro gráfico mostra os resultados da variação do ajuste α do ASMS e o último gráfico mostra o resultado da variação da incerteza do processo do FK.

1,1, 1 e $0,8 \times 10^{-4}$. Os gráficos apresentando os valores encontrados nos testes estão na Figura 22.

A seguir foram testados os parâmetros da combinação ACK. Para ela foram testados os parâmetros α do VDP, α do KCF, α do ASMS e a incerteza do processo do FK. Para esse parâmetros foram encontrados os valores 0,60, 1,1, 1 e $0,8 \times 10^{-4}$. Os gráficos apresentando os valores encontrados nos testes são apresentados na Figura 23.

Por último foi testada a penalidade de movimento β . Os testes partiram com o valor de β igual a 0, o que representa uma situação onde a penalidade não é utilizada. Os valores foram aumentados de 0,05 em 0,05 até chegar em 0,5. No caso do ACK, no valor de 5,5 o resultado encontrado era próximo ao melhor resultado encontrado, então o teste continuou até que fosse encontrado um resultado pior. O gráfico mostrando os resultados encontrados é apresentado na Figura 24. Como pode ser visto no gráfico, a utilização da penalidade contribui com a melhora do resultado quando aplicada de forma moderada. Entretanto, um valor de β muito alto faz com que os rastreadores com resultado muito diferente da previsão do filtro de Kalman tenham uma incerteza alta.

A Tabela 5 apresenta os parâmetros para todas as combinações, com os melhores

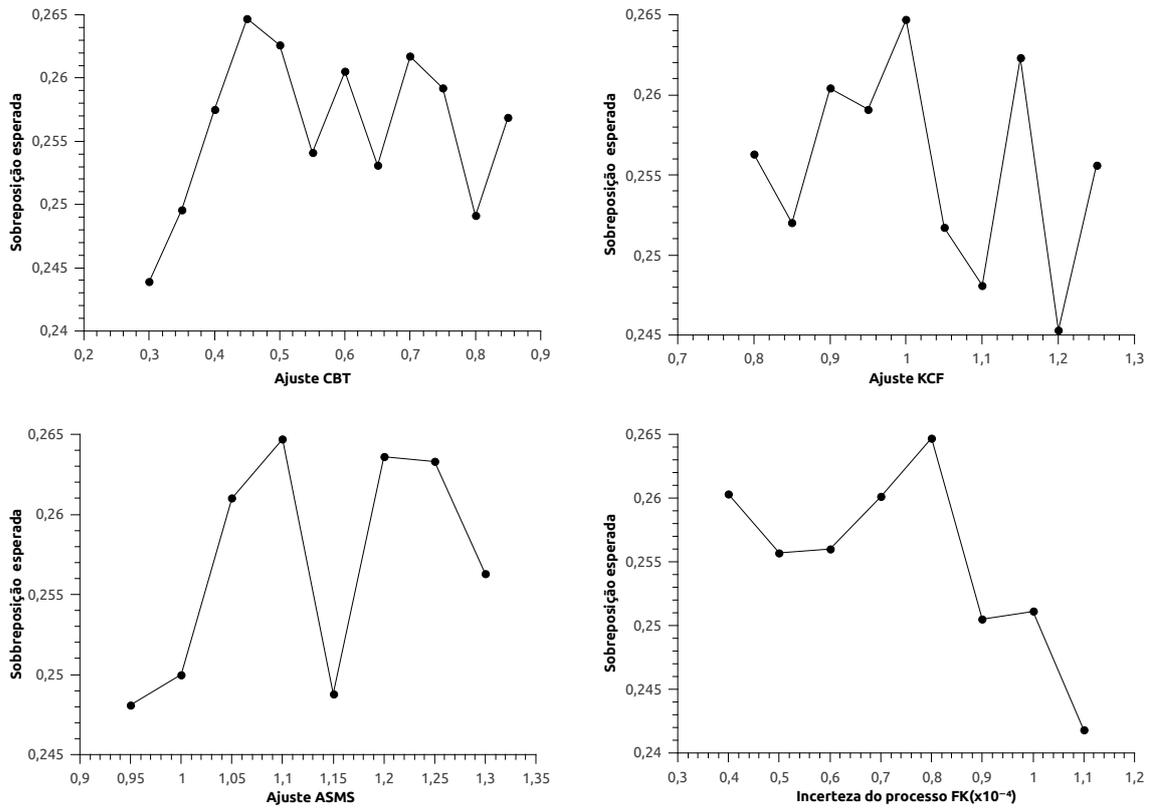


Figura 23: Testes de parâmetros da combinação KFebT-ACK. O primeiro gráfico mostra os resultados da variação do ajuste α do CBT. O segundo gráfico mostra os resultados da variação do ajuste α do KCF. O terceiro gráfico mostra os resultados da variação do ajuste α do ASMS e o último gráfico mostra o resultado da variação da incerteza do processo do FK.

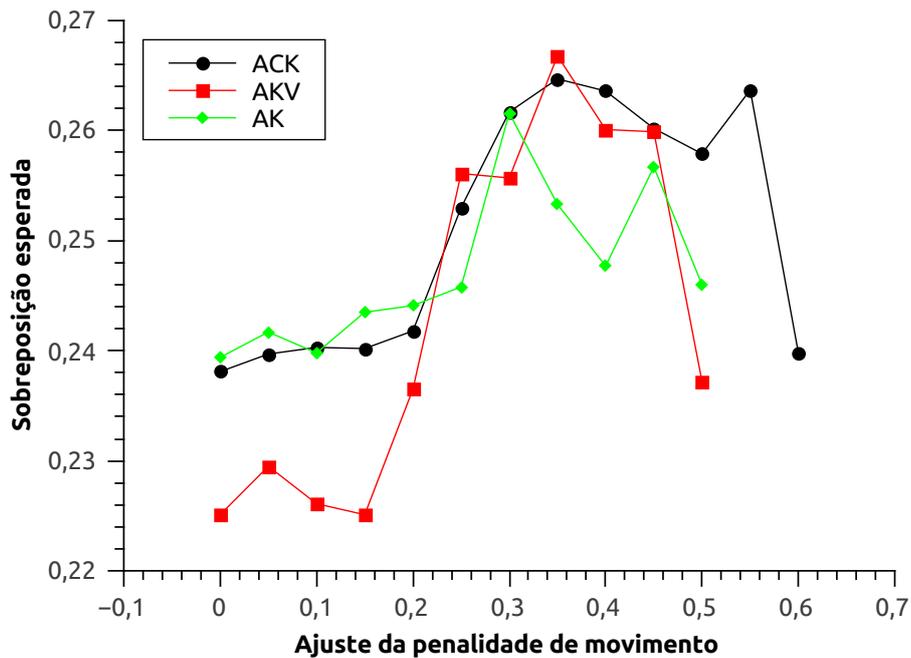


Figura 24: Testes do ajuste da penalidade de movimento (β). Foi utilizado o mesmo ajuste de penalidade para todos os rastreadores utilizados nas combinações.

Tabela 5: Melhores parâmetros encontrados para cada combinação de rastreadores.

Combinação	Ajuste					Incerteza do processo FK
	α ASMS	α CBT	α KCF	α VDP	β	
AK	1,1	-	1	-	0,3	$0,8 \times 10^{-4}$
ACK	1	0,45	1,1	-	0,35	$0,8 \times 10^{-4}$
AKV	1	-	1,1	0,6	0,35	$0,8 \times 10^{-4}$

Tabela 6: Teste da utilização da retroalimentação nas combinações ACK e AK.

Combinação	Sobreposição esperada	Precisão	Falhas
ACK com retroalimentação	0,2647	0,5229	1,3592
ACK sem retroalimentação	0,2239	0,4843	2,0063
AK com retroalimentação	0,2616	0,5239	1,2583
AK sem retroalimentação	0,2207	0,4938	1,9035

resultados obtidos. Esses parâmetros foram utilizados nos demais testes.

5.2.3 Retroalimentação

Os testes apresentados nesta seção avaliam o impacto da retroalimentação nos resultados da fusão. Segundo [Bailer, Pagani e Stricker \(2014\)](#), a utilização de retroalimentação na forma do último resultado obtido condiciona os rastreadores a seguirem uma única verdade, caso o resultado obtido esteja errado, todos os rastreadores passarão a errar. Por outro lado, caso um rastreador se perca e não receba nenhum retorno, ele poderá permanecer perdido pelo resto da sequência, influenciando negativamente o resultado final.

Os testes foram realizados com as combinações ACK e AK, a AKV foi removida pois o VDP não é capaz de estimar translação, precisando de um retorno para não perder o objeto de interesse. Na Tabela 6 estão os resultados do teste de retroalimentação, todos os parâmetros dos métodos foram mantidos os mesmos, sendo que a única diferença foi a utilização da retroalimentação. Comparando o AK, quando utilizada a retroalimentação o rastreador obteve uma sobreposição esperada e uma sobreposição 18% e 6% respectivamente maior, enquanto falhou 34% menos. Algo semelhante ocorreu com a combinação ACK que também obteve resultados melhores quando utilizada a retroalimentação.

Com base nos resultados obtidos, constatou-se que a utilização da retroalimentação colabora com a melhoria dos resultados no método proposto. A retroalimentação também possibilita que o método proposto utilize métodos especialistas, pois sem ela estes métodos são incapazes de seguir o objeto de interesse.

Tabela 7: Teste adaptação. São apresentados os resultados das combinações quando o KCF utiliza o próprio resultado ou quando utiliza o resultado da fusão para realizar a atualização do seu modelo.

Combinação	Sobreposição esperada	Precisão	Falhas
ACK rastreador	0,2647	0,5229	1,35
ACK fusão	0,2362	0,5129	1,4473
AKV rastreador	0,2668	0,5197	1,219
AKV fusão	0,2433	0,5196	1,4119
AK rastreador	0,2616	0,5239	1,2583
AK fusão	0,2415	0,5181	1,501

5.2.4 Adaptação

Nestes testes foi avaliado se deve ser utilizado o resultado do rastreador ou o resultado recebido pela retroalimentação durante a adaptação dos modelos dos rastreadores. Dos rastreadores utilizados, apenas o KCF realiza a adaptação do modelo. De forma geral, é mais simples para a integração dos rastreadores que a adaptação ocorra com o resultado do próprio rastreador. Nos códigos de outros autores que foram estudados no decorrer deste trabalho, os rastreadores costumam ser implementados como objetos. Nesses objetos costuma existir um método que recebe o novo quadro e realiza o rastreamento. Quando o rastreador realiza atualizações no seu modelo, ele costuma ser realizado nesse mesmo método.

Como os processos de rastreamento e adaptação do KCF já tinham sido separados para realizar testes anteriores, foi possível testar o resultado do KFebT quando o KCF utiliza o resultado da fusão para realizar a adaptação do seu modelo.

Na Tabela 7 estão os resultados dos testes da adaptação. Todos os demais parâmetros do método foram mantidos, a única diferença é quanto a qual resultado foi utilizado para realizar a adaptação do modelo do KCF. Em todas as combinações testadas, a utilização do resultado do próprio rastreador conduziu a resultados melhores do que a utilização do resultado final da fusão. Nas combinações ACK, AKV e AK a sobreposição esperada foi 12%, 9% e 8% maior, respectivamente, quando utilizado o resultado do próprio rastreador na adaptação.

Com os resultados obtidos é possível constatar que a utilização do resultado do próprio rastreador nas combinações testadas levou a resultados melhores. Só o KCF realiza a atualização, para reforçar os resultados obtidos seria interessante a utilização de mais rastreadores que também realizam atualização na seu modelo do objeto de interesse. De qualquer forma, não modificar a função de rastreamento dos rastreadores durante a integração reduz consideravelmente o esforço realizado e a probabilidade de serem adicionados erros no código.

Tabela 8: Comparação do método proposto com os rastreadores utilizados como base.

Rastreador	Sobreposição esperada	Precisão	Falhas	QPS
KFebT-AKV	0,2668	0,5197	1,219	59
KFebT-ACV	0,2647	0,5229	1,35	53
KFebt-AK	0,2668	0,5239	1,2583	109
ASMS	0,2125	0,5112	1,9736	198
KCF	0,1364	0,4397	3,5252	130
CBT	0,0548	0,4000	6,72	96

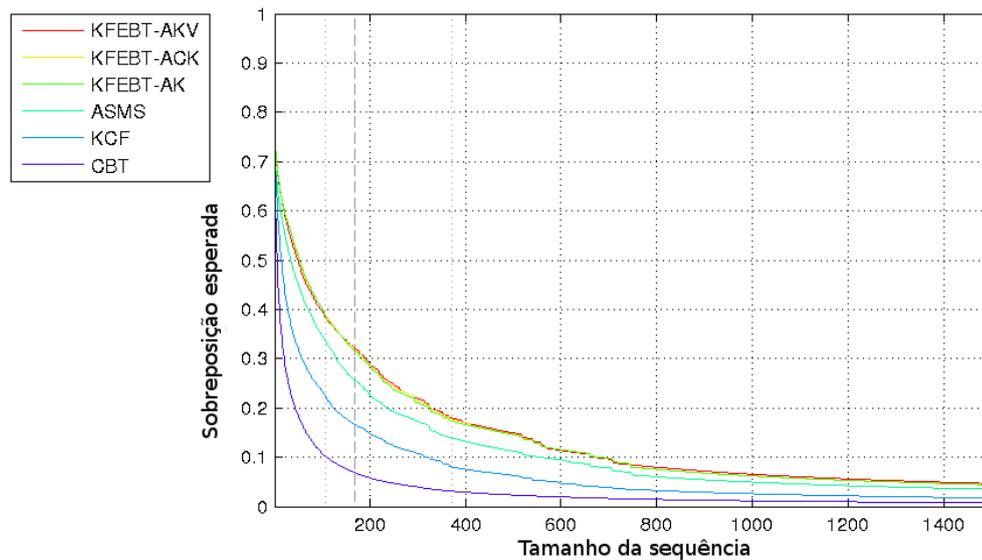


Figura 25: Gráfico de sobreposição esperada dos rastreadores na base do VOT2015.

5.2.5 Comparação com os rastreadores base

Nessa seção são comparados os resultados do método proposto com os rastreadores utilizados na combinação. Na Tabela 8 são apresentados os resultados do KFebT e dos rastreadores base. Todas as combinações do KFebT obtiveram resultados melhores que os rastreadores base em sobreposição esperada, precisão e falhas. Como esperado, o KFebT é mais lento que os rastreadores base, mas a combinação AK ainda é capaz de rodar a mais de 100 quadros por segundo e as combinações AKV e ACK a mais de 50 quadros por segundo em um computador móvel.

Na Figura 25 são apresentadas as curvas de sobreposição esperadas para o KFebT-AKV, ACK e AK e para os rastreadores utilizados como base ASMS, KCF e CBT. As curvas de sobreposição esperadas do KFebT são visualmente muito superiores às dos outros rastreadores. As três curvas do KFebT estão muito próximas e visivelmente acima do ASMS para todos os tamanhos de sequências.

O *toolkit* do VOT também é capaz de avaliar o comportamento dos rastreadores na ocorrência de cinco atributos visuais (movimentação da câmera, mudança na iluminação,

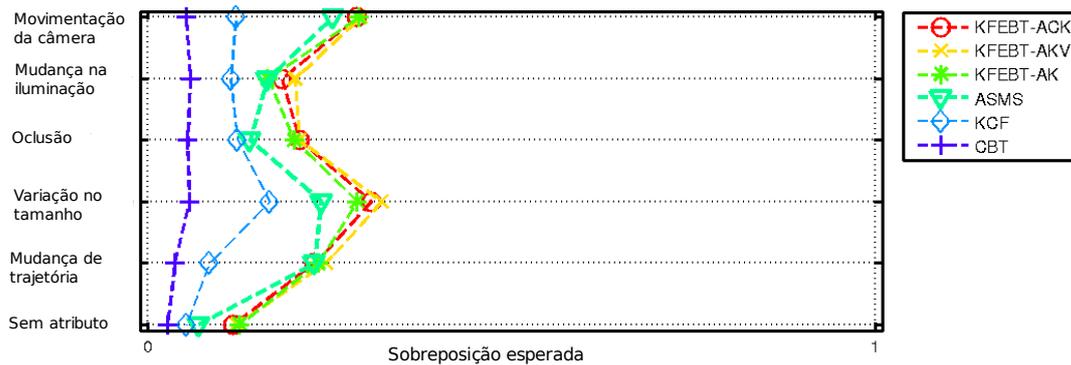


Figura 26: Resultado dos rastreadores por atributo visual na base do VOT2015.



Figura 27: Comparação dos resultado dos rastreadores na sequência *girl*. O retângulo preto representa a anotação do objeto. Na legenda é apresentada a sobreposição do rastreador no quadro atual, seguido da quantidade total de falhas e a sobreposição média até o quadro atual. Os vídeos comparativos dos resultados das 60 sequências do VOT2015 podem ser encontrados em www.youtube.com/playlist?list=PLuCzbviUtv_Gn54-U8H-JqDICLALm-IZ1

occlusão, variação no tamanho e mudança de trajetória) e quando nenhum desses atributos ocorre. Na Figura 26 tem-se a sobreposição esperada na ocorrência de cada atributo visual. Comparando com o ASMS, o melhor dentre os rastreadores utilizados nas combinações, o KFebT obteve resultados melhores ou iguais em todos os atributos. A diferença é mais visível nos atributos oclusão e variação de tamanho.

Na Figura 27 é apresentado um vídeo comparando visualmente os resultados do KFebT e dos rastreadores base.

Tabela 9: Rastreadores submetidos no VOT2015 utilizados na comparação.

Rastreador	Referência
ASMS	(VOJIR; NOSKOVA; MATAS, 2014)
bdf	(MARESCA; PETROSINO, 2014)
DAT	(POSSEGGER; MAUTHNER; BISCHOF, 2015)
DeepSRDCF	(DANELLJAN et al., 2015)
EBT	(ZHU; PORIKLI; LI, 2015)
FCT	(VARFOLOMIEIEV; LYSENKO, 2016)
FoT	(VOJIR; MATAS, 2014)
LDP	(LUKEŽIČ; ČEHOVIN; KRISTAN, 2016)
matflow	(MARESCA; PETROSINO, 2013)
MDNet	(PANG; LING, 2013)
MEEM	(ZHANG; MA; SCLAROFF, 2014)
NCC	(BRIECHLE; HANEBECK, 2001)
NSAMF	(LI; ZHU, 2014)
PKLTF	(GONZÁLEZ et al., 2014)
RAJSSC	(ZHANG et al., 2015)
S3Tracker	(LEE; YU, 2011)
SC-EBT	(WANG; YEUNG, 2014)
sKCF	(MONTERO; LANG; LAGANIERE, 2015)
SO-DLT	(WANG et al., 2015)
sPST	(HUA; ALAHARI; SCHMID, 2015)
SRDCF	(DANELLJAN et al., 2015)
Struck	(HARE; SAFFARI; TORR, 2011)
SumShift	(LEE; YU, 2011)

5.3 Comparação com o VOT2015

Nesta seção são comparados os resultados do método proposto com resultados submetidos para o VOT2015. Uma das vantagens da realização de desafios como o VOT é a padronização dos testes e a base de resultados que pode ser facilmente utilizada para comparar os resultados obtidos. O resultado de todos os rastreadores submetidos no VOT2015 estão disponíveis no site do VOT¹ e podem ser utilizados na comparação dos resultados sem a necessidade de serem executados novamente.

Foram utilizados na comparação os quinze rastreadores melhor classificados segundo a classificação do VOT2015, e os rastreadores capazes de executar em tempo real segundo os parâmetros definidos pelo VOT2015, ou seja, os rastreadores que atingem no mínimo 20 EFO. No total foram comparados 23 rastreadores e 3 combinações do KFebT. Os rastreadores utilizados e suas referências são apresentados na Tabela 9.

Os resultados deste teste são apresentados na Tabela 10. Avaliando os rastreadores propostos neste trabalho, a melhor combinação foi o KFebT-AKV, que atingiu a sétima

¹<http://www.votchallenge.net/vot2015/results.html>

Tabela 10: Comparação com os rastreadores submetidos no VOT2015.

Rastreador	Sobreposição esperada	Precisão	Falhas	EFO
MDNet	0,3783	0,6033	0,6936	0,87
DeepSRDCF	0,3181	0,5637	1,0457	0,38
EBT	0,3130	0,4732	1,0213	1,76
srdcf	0,2877	0,5592	1,2417	1,99
LDP	0,2785	0,4890	1,3332	5,19
sPST	0,2767	0,5473	1,4796	1,03
KFebT-AKV	0,2663	0,5185	1,2190	55,45
KFebT-ACK	0,2642	0,5224	1,3592	49,25
KFebT-AK	0,2610	0,5224	1,2583	102,65
SC-EBT	0,2548	0,5529	1,8587	0,79
NSAMF	0,2536	0,5305	1,2921	5,58
STRUCK	0,2458	0,4712	1,6097	2,44
RAJSSC	0,2420	0,5659	1,6296	2,12
S3Tracker	0,2403	0,5153	1,7680	14,70
SumShift	0,2341	0,5169	1,6815	17,27
SODLT	0,2329	0,5607	1,7769	0,83
DAT	0,2229	0,4856	2,2583	9,82
MEEM	0,2212	0,4993	1,8535	2,70
ASMS	0,2117	0,5066	1,8464	115,09
sKCF	0,1620	0,4847	2,6811	66,22
bdf	0,1526	0,4014	3,1058	200,24
PKLTF	0,1524	0,4527	2,7210	29,93
FCT	0,1512	0,4314	3,3383	83,37
matflow	0,1500	0,4199	3,1213	81,34
FoT	0,1385	0,4316	4,3605	143,62
NCC	0,0795	0,5000	11,3448	172,85

posição na comparação, sendo o quarto rastreador com menos falhas. É importante salientar que nenhum dos rastreadores melhor posicionados chegam perto de serem executados em tempo real, enquanto as três combinações do KFebT testadas atingem o tempo real com certo conforto.

Comparando com os rastreadores de tempo real, todas as combinações testadas do método proposto obtiveram melhor sobreposição esperada e precisão, enquanto falharam menos.

5.4 Testes com ruído

Na vida real, é improvável que o rastreador seja iniciado com a região da localização do objeto de interesse tão refinada quanto as anotações da base do VOT. Sendo assim, este teste visa avaliar o comportamento dos rastreadores quando sua iniciação é realizada com uma região com ruído.

Tabela 11: Resultados do teste com ruído.

Rastreador	Sobreposição esperada	Precisão	Falhas	QPS
KFebT-AKV	0,2443	0,4988	1,3699	63
KFebT-ACK	0,2392	0,4950	1,4705	60
KFebT-AK	0,2359	0,4959	1,5292	115
ASMS	0,2083	0,4899	1,9500	195
KCF	0,1309	0,4125	3,3278	140
CBT	0,0543	0,2833	9,3629	93

A metodologia de teste utilizada foi proposta no VOT 2014 (KRISTAN *et al.*, 2014), onde o algoritmo de rastreamento é iniciado com um ruído de no máximo 10% na região do objeto de interesse. Devido a essa iniciação estocástica, os testes são repetidos 15 vezes, mesmo quando o rastreador utilizado é determinístico. O teste foi realizado utilizando a base do VOT 2015.

Os resultados do teste com ruído são apresentados na Tabela 11. O rastreador KFebT-AKV obteve uma sobreposição esperada 17% maior que o ASMS e 86% maior que o KCF. As outras combinações do KFebT também obtiveram resultados melhores que os rastreadores comparados.

A maior vantagem dos rastreadores baseados no método proposto é a sua robustez. Comparado com o ASMS, o rastreador mais robusto utilizado na combinação, o KFebT-AKV, ACK e AK falharam 30%, 25% e 22% menos, respectivamente.

5.5 Testes sem supervisão

Assim como no teste com ruído, o teste sem supervisão tem como objetivo avaliar o comportamento dos rastreadores em um cenário mais próximo ao encontrado em uma aplicação real do que o teste padrão realizado no VOT.

Diferentemente do teste padrão do VOT, no teste sem supervisão o rastreador não é reiniciado. Ele deve ser capaz de reencontrar o objeto de interesse ou irá continuar falhando até o fim da sequência. Esta metodologia se aproxima mais de uma aplicação real, pois é improvável que o rastreador seja executado com outro sistema que monitore seus resultados quadro a quadro e o reinicie ou informe a localização correta do objeto caso ele falhe.

Este teste foi proposto no VOT 2016, mas os testes realizados neste trabalho utilizam a base de dados do VOT 2015. Para cada sequência, o teste foi realizado apenas uma vez.

A Tabela 12 apresenta os resultados do KFebT e dos rastreadores que foram utilizados nas combinações no teste sem supervisão. Nesse teste as falhas não são computadas e os rastreadores são avaliados com base nas suas sobreposições esperadas e precisão.

Tabela 12: Resultados do teste sem supervisão.

Rastreador	Sobreposição esperada	Precisão	QPS
KFebT-ACK	0,4527	0,3683	60
KFebT-AKV	0,4507	0,3682	61
KFebT-AK	0,4446	0,3599	114
ASMS	0,4004	0,3283	185
KCF	0,3922	0,2537	143
CBT	0,1687	0,0897	88

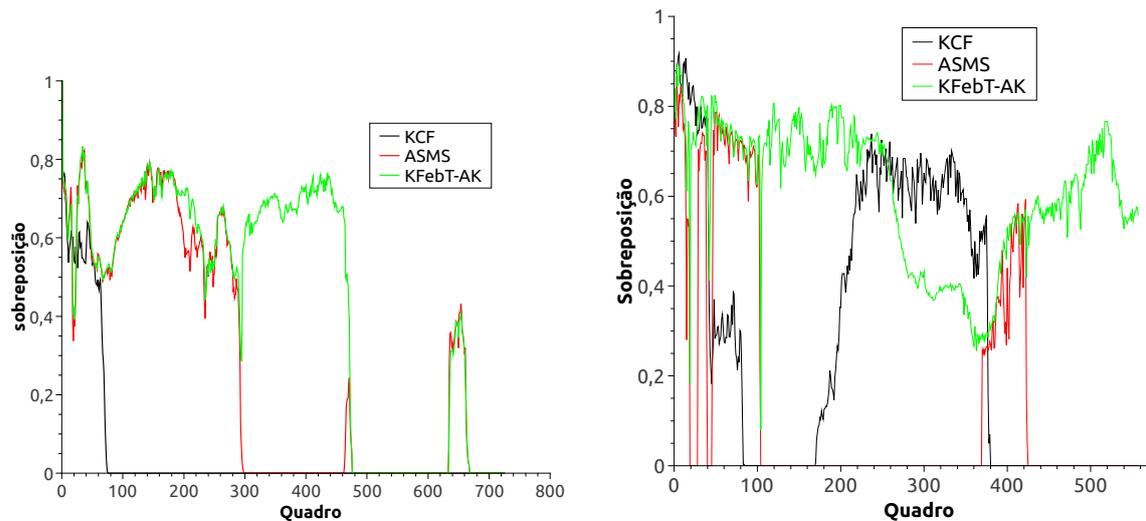


Figura 28: Gráficos da sobreposição do resultado com a anotação dos rastreadores KFebT-AK, ASMS e KCF para cada quadro das sequências *basketball* (esquerda) e *road* (direita) no teste sem supervisão.

Analisando a melhor combinação, o KFebT-ACK obteve uma sobreposição esperada 13% maior do que o ASMS e 15% maior que o KCF. Avaliando a sobreposição, o KFebT-ACK obteve resultados 12% e 45% melhores que o ASMS e o KCF respectivamente.

As demais combinações também obtiveram resultados melhores do que todos os rastreadores utilizados na comparação. Alguns resultados visuais do teste sem supervisão são apresentados na Figura 29. Nela são apresentados os resultados do KFebT-AK, ASMS e KCF para alguns quadros de seis sequências. Foram utilizados apenas esses rastreadores para que a imagem apresentasse um resultado mais nítidos.

Para as sequências *basketball* e *road* foram gerados gráficos apresentando a sobreposição obtida em cada quadro pelos rastreadores. Esses gráficos estão na Figura 28. Nos gráficos pode ser visto que o KFebT-AK falha menos, e em geral obtém uma sobreposição maior que os outros rastreadores.

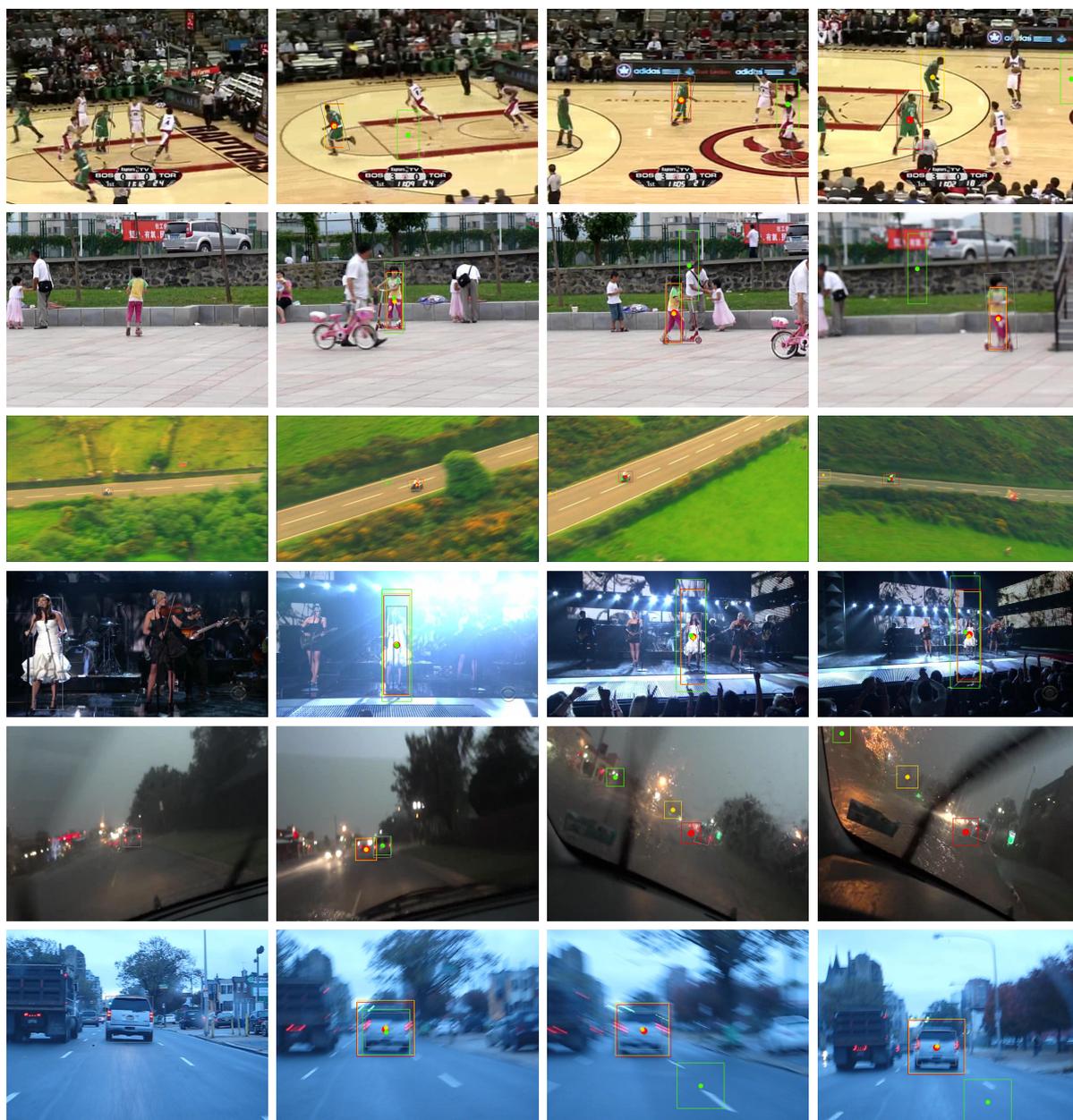


Figura 29: Resultado qualitativo do teste sem supervisão. São apresentados os quadros 1, 100, 200 e 300 das seqüências *basketball*, *girl*, *road*, *singer1*, *wiper* e *car1*. No quadro 1 é apresentada apenas a anotação do objeto de interesse. Nos outros quadros são apresentados os resultados do KFebT-AK (vermelho), do ASMS (amarelo), do KCF(verde) e a anotação do objeto de interesse (cinza).

6 Conclusão e trabalhos futuros

6.1 Conclusão

Neste trabalho foi discutido o problema de rastreamento de objetos livre de modelo prévio. Este é um problema desafiador pela quantidade de dificuldades que podem aparecer no decorrer do rastreamento, como oclusões ou mudança de iluminação, e pelo fato de ser utilizada apenas uma amostra do objeto de interesse para iniciar o rastreador. Alguns métodos propostos para resolver este problema focam em solucionar algum dos desafios, como variação de escala ou rastreamento do objetos com deformação, mas costumam obter resultados ruins em outras situações.

Este trabalho apresentou um novo método de combinação de resultados de rastreadores utilizando um Filtro de Kalman chamado KFebT. Este método é capaz de combinar o resultado de diversos rastreadores de forma simples e rápida.

O KFebT foi testado na base de dados do VOT2015 e avaliado em termos de expectativa de sobreposição, sobreposição, robustez e velocidade de execução. O KFebT supera os resultados dos métodos que o compõe (ASMS, KCF e uma simplificação do CMT) em relação à sobreposição esperada, precisão e robustez por uma boa margem, mostrando a eficiência do método proposto em realizar a combinação desses rastreadores.

Comparado com os métodos de tempo real submetidos no VOT2015, o método proposto obteve expectativa de sobreposição, precisão e robustez superiores a todas as técnicas comparadas. Quando comparado com os melhores rastreadores submetidos no VOT2015, o KFebT conseguiu atingir bons resultados, principalmente considerando que nenhum dos melhores rastreadores são capazes de executar em tempo real, e a combinação AK roda acima de 100 quadros por segundo em um computador móvel e as combinações ACK e AKV rodam acima de 60 quadros por segundo.

Um dos objetivos deste trabalho é que o KFebT seja capaz de ser executado em tempo real. Como as operações realizadas pelo KFebT são leves, o tempo de execução do método vai depender majoritariamente da complexidade dos rastreadores utilizados na combinação. Desta forma, neste trabalho foram utilizados rastreadores e métodos especialistas que são executados em tempo real, resultando em um método capaz de ser executado em tempo real.

A implementação do método proposto foi realizada em C++ e foi criada uma classe abstrata para realizar a interface com os métodos de rastreamento, facilitando a

integração de novos rastreadores. O código é mantido em um repositório público¹ e será atualizado à medida que a técnica for aprimorada.

A maior contribuição do KFebT é a possibilidade de utilizar métodos especialistas nas combinações, utilizando a mesma interface genérica utilizada para rastreadores. Desta forma, os métodos especialistas podem ser facilmente integrados no método proposto. Outros métodos que realizam a combinação de resultados de rastreadores costumam trabalhar com rastreadores completos.

6.2 Trabalhos futuros

O método proposto pode ser expandido das seguintes maneiras:

- Adicionando mais rastreadores na combinação. O KFebT foi implementado em C/C++ pelo fato dessa linguagem ser compilada, orientada a objetos e possuir bibliotecas de código livre voltadas para a visão computacional. Uma dificuldade para integrar mais rastreadores é que o rastreador deve ser implementado na mesma linguagem. No VOT2015, 9 dentre os 10 rastreadores melhor ranqueados são implementados, pelo menos em parte, na linguagem do Matlab e muitas vezes dependem de módulos do Matlab. A plataforma do Matlab e seus módulos são pagos, o que dificultaria o acesso de pessoas que não os possuem ao método proposto. Outra dificuldade é a obtenção do código fonte dos rastreadores que, muitas vezes, não é disponibilizado pelos autores.
- Incluindo uma estratégia de reiniciação dos rastreadores com resultados ruins. O resultado da combinação pode ser utilizado para realizar a reiniciação de rastreadores que obtenham resultados com muita incerteza por um certo período de tempo. Essa característica pode ajudar rastreadores cuja modelagem do objeto de interesse não consiga identificar alguma mudança na sua aparência no decorrer do rastreamento.
- Adicionando a estimativa de rotação no método proposto para melhorar a precisão do rastreador. Para isso, é necessário encontrar métodos de rastreamento que realizem a estimativa da rotação do objeto de interesse. O desafio VOT utiliza retângulos rotacionados na anotação da sua base, possibilitando testar métodos que estimam a rotação.
- Integrando o método proposto em uma aplicação móvel para testar o *feedback* das informações do controle no filtro de Kalman.

¹<https://github.com/psenna/KF-EBT>

Referências

- ALI, A. et al. Correlation, kalman filter and adaptive fast mean shift based heuristic approach for robust visual tracking. *Signal, Image and Video Processing*, Springer London, v. 9, n. 7, p. 1567–1585, 2015. Citado na página 30.
- ANDRILUKA, M.; ROTH, S.; SCHIELE, B. People-tracking-by-detection and people-detection-by-tracking. In: IEEE. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.], 2008. p. 1–8. Citado na página 20.
- BABENKO, B.; YANG, M. H.; BELONGIE, S. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 8, p. 1619–1632, Aug 2011. ISSN 0162-8828. Citado na página 48.
- BAILER, C.; PAGANI, A.; STRICKER, D. A superior tracking approach: Building a strong tracker through fusion. In: *Computer Vision–ECCV 2014*. [S.l.]: Springer, 2014. p. 170–185. Citado 5 vezes nas páginas 16, 30, 31, 32 e 62.
- BOLME, D. S. et al. Visual object tracking using adaptive correlation filters. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. [S.l.], 2010. p. 2544–2550. Citado 3 vezes nas páginas 15, 22 e 25.
- BOLME, D. S.; DRAPER, B. A.; BEVERIDGE, J. R. Average of synthetic exact filters. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.], 2009. p. 2105–2112. Citado na página 25.
- BRIECHLE, K.; HANEBECK, U. D. Template matching using fast normalized cross correlation. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Aerospace/Defense Sensing, Simulation, and Controls*. [S.l.], 2001. p. 95–102. Citado na página 66.
- ČEHOVIN, L. TraX: The visual Tracking eXchange Protocol and Library. *Neurocomputing*, 2017. ISSN 0925-2312. Citado 2 vezes nas páginas 41 e 54.
- ČEHOVIN, L.; LEONARDIS, A.; KRISTAN, M. Visual object tracking performance measures revisited. *arXiv preprint arXiv:1502.05803*, 2015. Citado na página 49.
- COCHRAN, W. T. et al. What is the fast fourier transform? *Proceedings of the IEEE*, IEEE, v. 55, n. 10, p. 1664–1674, 1967. Citado na página 26.
- COMANICIU, D.; RAMESH, V.; MEER, P. Real-time tracking of non-rigid objects using mean shift. In: IEEE. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. [S.l.], 2000. v. 2, p. 142–149. Citado 3 vezes nas páginas 16, 22 e 24.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. [S.l.], 2005. v. 1, p. 886–893. Citado na página 25.

- DANELLJAN, M. et al. Accurate scale estimation for robust visual tracking. In: BMVA PRESS. *British Machine Vision Conference, Nottingham, September 1-5, 2014*. [S.l.], 2014. p. 1–11. Citado 2 vezes nas páginas 22 e 25.
- DANELLJAN, M. et al. Learning spatially regularized correlation filters for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 4310–4318. Citado na página 66.
- EKIN, A.; TEKALP, A. M.; MEHROTRA, R. Automatic soccer video analysis and summarization. *IEEE Transactions on Image processing*, IEEE, v. 12, n. 7, p. 796–807, 2003. Citado na página 15.
- FUKUNAGA, K.; HOSTETLER, L. D. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, IEEE, v. 21, n. 1, p. 32–40, 1975. Citado na página 24.
- GAVRILA, D. M. The visual analysis of human movement: A survey. *Computer vision and image understanding*, Elsevier, v. 73, n. 1, p. 82–98, 1999. Citado na página 20.
- GONZÁLEZ, A. et al. Single object long-term tracker for smart control of a ptz camera. In: ACM. *Proceedings of the International Conference on Distributed Smart Cameras*. [S.l.], 2014. p. 39. Citado na página 66.
- GRABNER, H. et al. Tracking the invisible: Learning where the object might be. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. [S.l.], 2010. p. 1285–1292. Citado na página 15.
- HARE, S.; SAFFARI, A.; TORR, P. H. Struck: Structured output tracking with kernels. In: IEEE. *2011 International Conference on Computer Vision*. [S.l.], 2011. p. 263–270. Citado na página 66.
- HARITAOGU, I.; HARWOOD, D.; DAVIS, L. S. W 4: Real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 22, n. 8, p. 809–830, 2000. Citado na página 15.
- HENRIQUES, J. F. et al. Exploiting the circulant structure of tracking-by-detection with kernels. In: *Computer Vision–ECCV 2012*. [S.l.]: Springer, 2012. p. 702–715. Citado na página 25.
- HENRIQUES, J. F. et al. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 37, n. 3, p. 583–596, 2015. Citado 2 vezes nas páginas 16 e 25.
- HUA, Y.; ALAHARI, K.; SCHMID, C. Online object tracking with proposal selection. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 3092–3100. Citado na página 66.
- HWANG, I. et al. Multiple-target tracking and identity management in clutter, with application to aircraft tracking. In: IEEE. *American Control Conference, 2004. Proceedings of the 2004*. [S.l.], 2004. v. 4, p. 3422–3428. Citado na página 15.
- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Forward-backward error: Automatic detection of tracking failures. In: IEEE. *Pattern recognition (ICPR), 2010 20th international conference on*. [S.l.], 2010. p. 2756–2759. Citado na página 28.

- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 34, n. 7, p. 1409–1422, 2012. Citado 3 vezes nas páginas 15, 16 e 26.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Citado 2 vezes nas páginas 16 e 32.
- KOLLER, D.; WEBER, J.; MALIK, J. Robust multiple car tracking with occlusion reasoning. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 1994. p. 189–196. Citado na página 20.
- KRISTAN, M. et al. The visual object tracking vot2016 challenge results. In: _____. *Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*. Cham: Springer International Publishing, 2016. p. 777–823. ISBN 978-3-319-48881-3. Disponível em: <http://dx.doi.org/10.1007/978-3-319-48881-3_54>. Citado na página 48.
- KRISTAN, M. et al. The visual object tracking vot2015 challenge results. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. [S.l.: s.n.], 2015. p. 1–23. Citado 7 vezes nas páginas 15, 16, 25, 47, 50, 51 e 52.
- KRISTAN, M. et al. The visual object tracking vot2013 challenge results. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. [S.l.: s.n.], 2013. p. 98–111. Citado na página 47.
- KRISTAN, M. et al. The visual object tracking vot2014 challenge results. *ECCV2014 Workshops, Workshop on visual object tracking challenge*, 2014. Citado 2 vezes nas páginas 47 e 68.
- KWON, J.; LEE, K. M. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.], 2009. p. 1208–1215. Citado na página 16.
- LEE, J.-Y.; YU, W. Visual tracking by partition-based histogram backprojection and maximum support criteria. In: IEEE. *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. [S.l.], 2011. p. 2860–2865. Citado na página 66.
- LEPETIT, V.; FUA, P. *Monocular model-based 3D tracking of rigid objects*. [S.l.]: Now Publishers Inc, 2005. Citado na página 18.
- LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. Brisk: Binary robust invariant scalable keypoints. In: IEEE. *Computer Vision (ICCV), 2011 IEEE International Conference on*. [S.l.], 2011. p. 2548–2555. Citado na página 27.
- LI, Y.; ZHU, J. A scale adaptive kernel correlation filter tracker with feature integration. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2014. p. 254–265. Citado na página 66.
- LUCAS, B. D.; KANADE, T. et al. An iterative image registration technique with an application to stereo vision. In: *IJCAI*. [S.l.: s.n.], 1981. v. 81, n. 1, p. 674–679. Citado 2 vezes nas páginas 28 e 43.

- LUKEŽIČ, A.; ČEHOVIN, L.; KRISTAN, M. Deformable parts correlation filters for robust visual tracking. *arXiv preprint arXiv:1605.03720*, 2016. Citado na página 66.
- MAGGIO, E.; CAVALLARO, A. *Video tracking: theory and practice*. [S.l.]: John Wiley & Sons, 2011. Citado 3 vezes nas páginas 18, 20 e 21.
- MAHALANOBIS, A. et al. Unconstrained correlation filters. *Applied Optics*, Optical Society of America, v. 33, n. 17, p. 3751–3759, 1994. Citado na página 25.
- MARESCA, M. E.; PETROSINO, A. Matrioska: A multi-level approach to fast tracking by learning. In: *Image Analysis and Processing-ICIAP 2013*. [S.l.]: Springer, 2013. p. 419–428. Citado na página 66.
- MARESCA, M. E.; PETROSINO, A. Clustering local motion estimates for robust and efficient object tracking. In: SPRINGER. *Computer Vision-ECCV 2014 Workshops*. [S.l.], 2014. p. 244–253. Citado na página 66.
- MONTERO, A.; LANG, J.; LAGANIERE, R. Scalable kernel correlation filter with sparse feature integration. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. [S.l.: s.n.], 2015. p. 24–31. Citado 5 vezes nas páginas 15, 22, 25, 32 e 66.
- NAM, H.; HAN, B. Learning multi-domain convolutional neural networks for visual tracking. *arXiv preprint arXiv:1510.07945*, 2015. Citado na página 30.
- NEBEHAY, G. *A Deformable Part Model for One-Shot Object Tracking*. Tese (Doutorado) — Graz University of Technology, set. 2016. Citado na página 18.
- NEBEHAY, G.; PFLUGFELDER, R. Consensus-based matching and tracking of keypoints for object tracking. In: *Winter Conference on Applications of Computer Vision*. [S.l.]: IEEE, 2014. p. 862–869. Citado 4 vezes nas páginas 15, 16, 20 e 30.
- NEBEHAY, G.; PFLUGFELDER, R. Clustering of Static-Adaptive correspondences for deformable object tracking. In: *Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2015. Citado 5 vezes nas páginas 27, 28, 36, 43 e 44.
- PANG, Y.; LING, H. Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 2784–2791. Citado na página 66.
- PAPANIKOLOPOULOS, N. P.; KHOSLA, P. K.; KANADE, T. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 9, n. 1, p. 14–35, 1993. Citado na página 15.
- POSSEGGGER, H.; MAUTHNER, T.; BISCHOF, H. In defense of color-based model-free tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 2113–2120. Citado na página 66.
- ROSS, D. A. et al. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, v. 77, n. 1, p. 125–141, 2008. ISSN 1573-1405. Disponível em: <<http://dx.doi.org/10.1007/s11263-007-0075-7>>. Citado na página 48.

- ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. *Computer vision—ECCV 2006*, Springer, p. 430–443, 2006. Citado na página 27.
- SANTNER, J. et al. Prost: Parallel robust online simple tracking. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. [S.l.], 2010. p. 723–730. Citado na página 30.
- SHI, J.; TOMASI, C. Good features to track. In: IEEE. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. [S.l.], 1994. p. 593–600. Citado 2 vezes nas páginas 43 e 44.
- SHU, G. et al. Part-based multiple-person tracking with partial occlusion handling. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. [S.l.], 2012. p. 1815–1821. Citado na página 20.
- SMEULDERS, A. W. et al. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 36, n. 7, p. 1442–1468, 2014. Citado 2 vezes nas páginas 49 e 52.
- VARFOLOMIEIEV, A.; LYSENKO, O. An improved algorithm of median flow for visual object tracking and its implementation on arm platform. *Journal of Real-Time Image Processing*, Springer, v. 11, n. 3, p. 527–534, 2016. Citado na página 66.
- VOJÍŘ, T.; MATAS, J. The enhanced flock of trackers. In: *Registration and Recognition in Images and Videos*. [S.l.]: Springer, 2014. p. 113–136. Citado na página 66.
- VOJIR, T.; MATAS, J.; NOSKOVA, J. Online adaptive hidden markov model for multi-tracker fusion. *arXiv preprint arXiv:1504.06103*, 2015. Citado 3 vezes nas páginas 16, 31 e 46.
- VOJIR, T.; NOSKOVA, J.; MATAS, J. Robust scale-adaptive mean-shift for tracking. In: SPRINGER. *Scandinavian Conference on Image Analysis*. [S.l.], 2013. p. 652–663. Citado na página 52.
- VOJIR, T.; NOSKOVA, J.; MATAS, J. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, Elsevier, v. 49, p. 250–258, 2014. Citado 6 vezes nas páginas 15, 16, 22, 24, 42 e 66.
- WANG, N. et al. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015. Citado na página 66.
- WANG, N.; YEUNG, D.-Y. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. [S.l.: s.n.], 2014. p. 1107–1115. Citado 2 vezes nas páginas 16 e 66.
- WU, Y.; LIM, J.; YANG, M.-H. Online object tracking: A benchmark. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2013. p. 2411–2418. Citado 2 vezes nas páginas 49 e 52.
- YAN, F. et al. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In: IEEE. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. [S.l.], 2006. v. 1, p. 634–641. Citado na página 15.

- YU, X. et al. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: ACM. *Proceedings of the eleventh ACM international conference on Multimedia*. [S.l.], 2003. p. 11–20. Citado na página 15.
- ZHANG, J.; MA, S.; SCLAROFF, S. Meem: Robust tracking via multiple experts using entropy minimization. In: *Computer Vision–ECCV 2014*. [S.l.]: Springer, 2014. p. 188–203. Citado 2 vezes nas páginas 31 e 66.
- ZHANG, K.; ZHANG, L.; YANG, M.-H. Real-time compressive tracking. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2012. p. 864–877. Citado na página 15.
- ZHANG, M. et al. Robust visual tracking using joint scale-spatial correlation filters. In: IEEE. *Image Processing (ICIP), 2015 IEEE International Conference on*. [S.l.], 2015. p. 1468–1472. Citado na página 66.
- ZHU, G.; PORIKLI, F.; LI, H. Tracking randomly moving objects on edge box proposals. *arXiv preprint arXiv:1507.08085*, 2015. Citado na página 66.