

# Uso do TLD Aplicado ao CARMEN para Rastreamento de Veículos

Jessica Oliveira Brito e Vinicius Cardoso

**Resumo** — Neste trabalho é apresentada a implementação do framework TLD (Tracking-Learning-Detection) para o rastreamento de veículos em um fluxo de vídeo. O framework foi aplicado ao CARMEN (Carnegie Mellon Robot Navigation Toolkit), e os testes foram realizados usando logs da IARA (Intelligent Autonomous Robotic Automobile). O algoritmo seguiu um veículo a frente da IARA por todo o anel viário da Universidade Federal do Espírito Santo (UFES), apresentando bons resultados.

## I. INTRODUÇÃO

Rastreamento de objetos é uma tarefa complexa. De acordo com [1] isso se dá devido a duas características principais. A primeira delas descreve as situações relacionadas com o objeto, como formas, variações de escala, oclusões, aparições e reaparições durante o processo. Enquanto que a segunda está relacionada à cena, já que podem ocorrer diversas alterações em seus elementos, como alterações de iluminação e cenário, distrações e até mesmo os próprios movimentos da câmera.

Este trabalho usa um algoritmo conhecido como TLD (Tracking-Learning-Detection) proposto por [4], que busca resolver o problema de rastreamento a longo prazo combinando rastreamento e detecção, de maneira que o rastreamento contínuo de um objeto possa ser realizado.

O TLD foi aplicado em um módulo do CARMEN (Carnegie Mellon Robot Navigation Toolkit)<sup>1</sup> que poderá ser usado para o rastreamento de objetos aplicado em um carro autônomo. Uma aplicação é o rastreamento de um veículo à frente, permitindo um robô segui-lo autonomamente.

## II. TRABALHOS RELACIONADOS

Em [2] é proposto um método que apresenta um desempenho que é preciso, mesmo sofrendo mudanças na translação, escala e rotação e sendo aplicável a diferentes classes de objetos. É um método robusto para oclusão parcial e completa. Usa combinação e rastreamento de pontos-chave para definir o modelo do objeto. Através de um sistema de votos, cada ponto-chave vota na posição que ele acredita ser o centro do objeto. Posteriormente esse conjunto de votos é computado usando distância Euclidiana entre o par coordenado, e essas distâncias são comparadas as distâncias dos pontos-chave correspondentes. Se um objeto sofre alguma rotação, os votos precisam sofrer a mesma rotação. Um consenso dentre esses votos é calculado. Assim, a posição do objeto tende a ser mais precisa, e relativamente flexível, já que os pontos-chave possuem uma taxa de

flexibilidade definida por parâmetro.

[3] sugere uma abordagem sinérgica que integra rastreamento, detecção e localização tanto da pista quanto do veículo. O rastreamento da pista é feito usando filtros direcionáveis a partir da construção de uma imagem IPM (Inverse-Perspective-Mapped) e posteriormente aplicando um conjunto de filtros direcionáveis a imagem. Já a detecção e rastreamento do veículo consistem em um detector de veículos baseado em aprendizagem ativa. Essa abordagem melhora o desempenho de um sistema de rastreamento de pistas, a precisão de um sistema de detecção de veículos, além disso, sugere uma nova abordagem para localizar e rastrear outros veículos na pista.

Uma extensão para o TLD é proposta em [5], usando o conceito de um detector genérico e um validador projetado para rastreamento de faces em tempo real, sendo resistente a oclusões e mudanças de aparência. O detector off-line localiza imagens frontais de faces, enquanto que o validador que é treinado online decide qual face corresponde ao objeto (face) sendo rastreado. Eles avaliam quantitativamente diversas estratégias para a construção desse validador durante o rastreio. Usando dois vídeos diferentes, o sistema é capaz de detectar e rastrear a face, e aprender automaticamente um modelo de múltiplas visualizações partindo de um único exemplo frontal e um vídeo não rotulado.

[6] implementa o algoritmo do TLD[4], reduzindo o número de frames, assumindo que o tamanho e a posição dos objetos não varia tanto entre frames. Usando restrições para filtrar as escolhas das bounding boxes, essa implementação limita a amostra a 2.000. O algoritmo faz uso de classificadores, como por exemplo, o Fern, esse classificador permite que os melhores conjuntos de configurações sejam encontrados.

Berger et al. [7] apresentam um sistema de rastreamento de objeto biologicamente inspirado baseado em uma técnica de aprendizagem de máquina conhecida como Virtual Generalizing Random Access Memory (VG-RAM) Weightless Neural Networks (WNN). VG-RAM WNN é uma técnica de aprendizagem de máquina eficaz que provê uma implementação simples e um treinamento rápido. O sistema deles modela o movimento de sacada feito pelo olho, a transformação sofrida pelas imagens capturadas pelos olhos a partir da retina até o Colículo Superior (Superior Colliculus - SC) e a resposta dos neurônios do SC para padrões visualizados anteriormente. Para avaliar o desempenho do sistema eles usaram um banco de dados para rastreamento visual. Os resultados obtidos por eles mostram que a abordagem desenvolvida rastreia um objeto de interesse em um vídeo de maneira confiável e eficiente, e com acurácia

<sup>1</sup> <http://carmen.sourceforge.net/home.html>

igual ou superior aos trabalhos relacionados.

Em [9], a abordagem do TLD é usada para o rastreamento recursivo. E o detector de objetos com uma abordagem em cascata foi estendido usando características diferentes como parâmetro com a finalidade de reduzir o tempo de execução. Este detector é composto por quatro componentes, sendo eles um detector de primeiro plano, um filtro de variância, um classificador fern aleatório baseado em características específicas e um método de correspondência de templates. Não é aplicado nenhum tipo de distorção na imagem, diferentemente do que acontece no TLD. Essa abordagem foi implementada em C++, e difere pouco da original, sendo uma versão otimizada e que teve sua performance comparada com a de métodos existentes. Essa foi a implementação usada neste trabalho e encontra-se disponível online<sup>2</sup>.

### III. METODOLOGIA

A base para esse trabalho é o TLD, que é um framework proposto por Kalal et al. [4] para rastreamento, decompondo explicitamente a tarefa de rastreamento a longo prazo em três partes, rastreamento, aprendizagem e detecção. O rastreador segue o objeto desconhecido em um fluxo de vídeo frame a frame, com o objetivo de identificar o objeto ou se o objeto está presente em cada frame do vídeo. O detector encontra todas as aparições daquele objeto até o momento e corrige o rastreador caso seja necessário. A parte da aprendizagem faz uma estimativa dos erros do detector e o atualiza para prevenir que esses erros aconteçam no futuro. Também foi desenvolvido um método de aprendizagem (P-N learning) que estima os erros usando dois critérios, o primeiro é chamado de P-expert e faz uma estimativa das detecções erradas, e o segundo é chamado de N-expert e faz uma estimativa dos alarmes falsos. Esse algoritmo consegue lidar com mudanças de escala na imagem. O TLD usa a abordagem *Scan-Window* como base para o detector, para analisar a imagem de entrada e decidir sobre a presença ou a ausência do objeto.

Em [9] é usada uma abordagem de janela deslizante. Como pode ser visto em [9], o espaço de busca precisa ser reduzido a um subespaço, devido ao grande número de possíveis sub-janelas existentes para cada imagem. Essa redução é feita usando as seguintes restrições. Primeiro, é assumido que o objeto de interesse mantém a sua proporção de aspecto (o objeto mantém as suas características mesmo sofrendo variações de escala). Posteriormente, são introduzidas as margens  $d_x$  e  $d_y$  entre duas sub-janelas adjacentes, e os valores de  $d_x$  e  $d_y$  são definidos como um décimo dos valores da boundingbox original. Para que a busca em múltiplas escalas possa ser empregada é usado um fator de escala. Além disso, também são consideradas as sub-janelas com uma área mínima de apenas 25 pixels.

Neste trabalho foi feita a implementação do rastreador usando o TLD aplicado ao CARMEN, uma coleção de softwares (módulos) para controle de robôs móveis com suporte a C/C++ e Java. A comunicação entre os módulos é feita através de mensagens, utilizando o protocolo de

comunicação IPC (do inglês: Inter-Process Communication), cada módulo pode publicar uma mensagem e/ou se inscrever para receber as mensagens de outro módulo, a distribuição das mensagens é feita por um processo central.

O CARMEN é utilizado na plataforma Intelligent Autonomous Robotic Automobile (IARA, **Figura 1**), um carro autônomo desenvolvido pelo Laboratório de Computação de Alto Desempenho (LCAD)<sup>3</sup>. A IARA foi utilizada para os experimentos neste trabalho.



Figura 1. Intelligent Autonomous Robotic Automobile (IARA).

Assim como requisito para essa implementação, esse rastreador será usado em aplicações no mundo real, processando imagens a 16 frames por segundo (número máximo de frames enviados pela câmera utilizada).

Para o desenvolvimento deste trabalho primeiramente foi feita uma pesquisa na literatura sobre o TLD, como requisitada, a implementação seria feita usando a linguagem C++ para que fosse possível a integração com o Carmen. Já que a implementação original foi feita em MATLAB<sup>4</sup>, outras implementações da biblioteca foram pesquisadas. Entre elas, a biblioteca do OpenCV<sup>5</sup>, que conforme consulta a alguns desenvolvedores, foi considerada muito lenta para aplicações no mundo real, além de ser necessária a utilização da versão 3.0, no CARMEN atualmente é utilizada a versão 2.4.9 do OpenCV. Desta forma a implementação proposta por [9] cumpriu com os requisitos, sendo desenvolvida em C++, conseguindo processar 16 frames/s (FPS), além de ser de código aberto, fácil manipulação e utilizar a biblioteca do OpenCV já utilizada pelos módulos do CARMEN. Porém, originalmente o pacote OpenTLD foi construído para ser instalado usando cmake. Já que o CARMEN utiliza compilação usando Makefile, foi preciso verificar os arquivos necessários e transformar a compilação para Makefile.

Feita a conversão do modo de compilação, foi criado um módulo chamado `tracker_opentld` para receber as imagens da câmera Bumblebee® instalada na IARA (**Figura 2**). Essas imagens são enviadas pelo módulo do driver da câmera, frame por frame, por meio de mensagens, através do protocolo IPC.

<sup>3</sup> <http://www.lcad.inf.ufes.br/>

<sup>4</sup> <https://github.com/zk00006/OpenTLD>

<sup>5</sup> <http://opencv.org/>

<sup>2</sup> <https://github.com/gnebehay/OpenTLD>

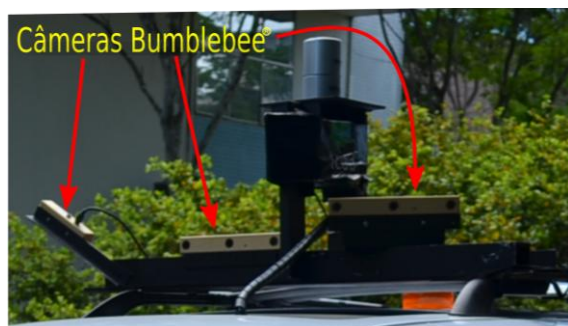


Figura 2. Câmeras instaladas na IARA.

Sempre que uma nova mensagem chega, o programa é notificado e uma função para tratar a mensagem é acionada, dentro desta função é feito o pré-processamento da imagem, como por exemplo, redimensionamento e mudança do padrão de cor de RGB (utilizado no CARMEN) para BGR (utilizado pelo OpenCV), e o processamento usando o TLD.

Através de uma interface gráfica, o usuário delimita uma área na imagem selecionando uma boundingbox (quadrado delimitando o objeto a ser rastreado), como mostrado na Figura 3. Dada a boundingbox com objeto a ser rastreado, algoritmo do TLD processa essa imagem e as próximas recebidas, fazendo o processo de rastreamento. A cada futuro frame da sequência de imagens o algoritmo retorna um boundingbox com a posição x, y, altura, comprimento e confiança do objeto na imagem. O módulo publica uma mensagem contendo essas informações e o timestamp da mensagem da imagem recebida também utilizando o IPC para a utilização por outros módulos.

Algumas possíveis configurações podem ser feitas tanto na interface como nas configurações iniciais do algoritmo.

Na interface é possível:

- Ativar/desativar visualização da trajetória do objeto;
- Ativar/desativar a aprendizagem do algoritmo;

Nas configurações iniciais pode-se configurar:

- Altura e largura da imagem
- Threshold de confiança do rastreamento
- Número da menor e maior escala a partir do tamanho do objeto inicial
- Número de Features extraídas
- Menor tamanho da *Scan-Window*
- Valor do parâmetro P-expert (chamado de thetaP)
- Valor do parâmetro N-expert (chamado de thetaN)

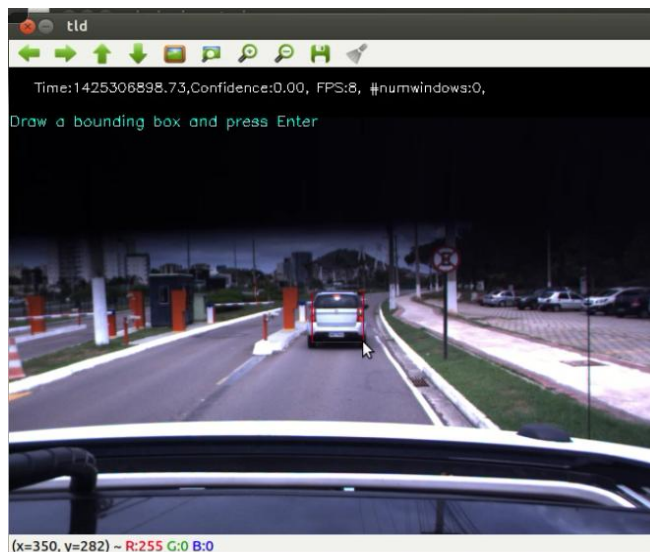


Figura 3. Interface para seleção da boundingbox.

#### IV. EXPERIMENTOS E RESULTADOS

O principal uso dessa implementação é o rastreamento de um carro para que futuramente a IARA possa, de forma autônoma, seguir um veículo à frente. Partindo deste objetivo buscando avaliar o sistema foram feitos experimentos utilizando logs de uma câmera instalada na IARA (Figura 2), em ambiente real.

No primeiro experimento foi solicitado a um motorista que conduzisse um carro a frente da IARA. Este carro deve ser rastreado durante o percurso em torno do anel viário da Universidade Federal do Espírito Santo (UFES), já que este é o percurso principal de testes da plataforma. Um vídeo demonstrando o experimento está disponível online<sup>6</sup>.

O percurso é dinâmico, tendo variações de iluminação, passagem de outros veículos e pessoas. Além disso, duas cancelas, seis quebra-molas, e pavimentação que varia entre paralelepípedo e asfalto, fazendo a imagem capturada pela câmera sofrer tremulações e ser afetada por movimentos bruscos.

Para diminuir a interferência do sistema de adaptação de iluminação da câmera, que causa grande variação na coloração da imagem, a parte superior da lente da câmera foi fechada reduzindo a influência da luz solar diretamente na câmera.

Para avaliar os resultados desta volta foi computada a confiança do rastreamento a cada frame e calculada a média e o desvio padrão ao longo do trajeto.

O gráfico na Figura 4 mostra a variação da confiança do rastreamento durante o percurso, onde 0 é a perda do rastreamento e 1 é a confiança total.

<sup>6</sup> <https://youtu.be/fThYL5N-aAw>

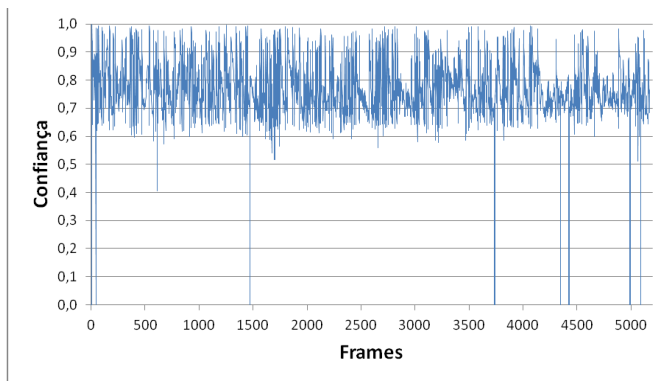


Figura 4. Gráfico de confiança para cada frame do experimento da volta da UFES.

Neste experimento a confiança teve uma média de 0.77 dentro do threshold definido (0.5) e com um desvio padrão de 0.09.

Os pontos aonde a confiança chegou a zero são causados pelo movimento brusco da câmera quando a IARA passa pelos quebra-molas (seis no total) e na primeira cancela que obstrui parcialmente a imagem do carro da frente. O **Anexo 1** mostra os exemplos da situação em um dos quebra-molas e o **Anexo 2** mostra a passagem pela primeira cancela.

Outra situação relevante no experimento, ilustrada pelo **Anexo 3**, foi a perda de precisão e correção da detecção ocorridas em curvas, onde o algoritmo aumentou o tamanho da bounding box para recuperar o objeto rastreado. A causa foi avaliada como sendo a mudança da posição do veículo na imagem, diferente da esperada, além do fato de que a marcação da bounding box para rastreamento é em 2D e foi feita da traseira do veículo e na curva ela sofre uma distorção.

## V. CONCLUSÕES

Como observado, o TLD tem dificuldades em situações de grande variação de iluminação e sofre influência do plano de fundo da imagem. Além de situações como quebra-molas, pelo movimento acentuado da câmera, e em cancelas onde há a perda da detecção, o que mostra que para essa aplicação, determinados ambientes ainda precisam ser tratados. Entretanto, o TLD tem uma boa recuperação e mesmo com a baixa confiança manteve a posição muito próxima da posição real quando há o tratamento de iluminação da câmera.

A integração foi bem sucedida, e já está disponível para uso na IARA, através do módulo disponível no GitHub do LCAD<sup>7</sup>.

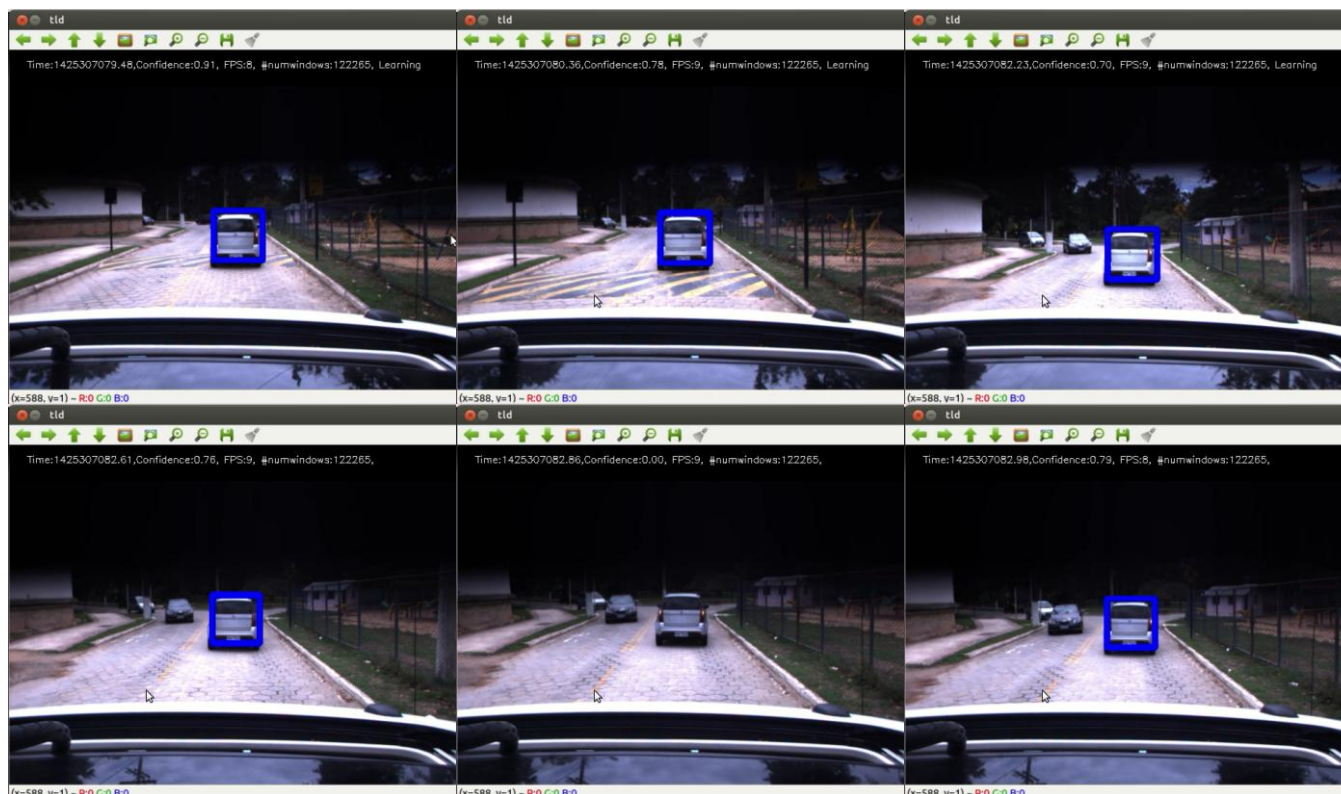
## BIBLIOGRAFIA

- [1] Lu, Y., Wu, T., & Chun Zhu, S. (2014). Online object tracking, learning and parsing with and-or graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3462-3469).
- [2] Nebehay, G., & Pflugfelder, R. (2014, March). Consensus-based matching and tracking of keypoints for object tracking. In *IEEE Winter Conference on Applications of Computer Vision* (pp. 862-869). IEEE.
- [3] Sivaraman, S., & Trivedi, M. M. (2013). Integrated lane and vehicle detection, localization, and tracking: A synergistic approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 906-917.
- [4] Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7), 1409-1422.
- [5] Kalal, Z., Mikolajczyk, K., & Matas, J. (2010, September). Face-tld: Tracking-learning-detection applied to faces. In *2010 IEEE International Conference on Image Processing* (pp. 3789-3792). IEEE.
- [6] Anenberg, B., & Meister, M. Tracking-Learning-Detection.
- [7] Berger, M., De Souza, A. F., de Oliveira Neto, J., de Aguiar, E., & Oliveira-Santos, T. (2016). Visual tracking with VG-RAM Weightless Neural Networks. *Neurocomputing*, 183, 90-105.
- [8] Kalal, Z., Mikolajczyk, K., & Matas, J. (2010, August). Forward-backward error: Automatic detection of tracking failures. In *Pattern recognition (ICPR), 2010 20th international conference on* (pp. 2756-2759). IEEE.
- [9] Nebehay, G. (2012). Robust object tracking based on tracking-learning-detection. na.

<sup>7</sup><https://goo.gl/tajZU1>



## ANEXOS



Anexo 1. Exemplo da perda do rastreamento na passagem da IARA pelo quebra-molas - Visualização em vídeo disponível em <https://youtu.be/fThYL5N-aAw?t=180>.



Anexo 2. Exemplo da perda do rastreamento na passagem da IARA pela cancela - Visualização em vídeo disponível em <https://youtu.be/fThYL5N-aAw?t=5>.



Anexo 3. Curva onde o algoritmo faz a recuperação da posição do objeto - Visualização em vídeo disponível em <https://youtu.be/fThYL5N-aAw?t=206>.