

Processamento de Imagens

Rastreio de Imagem

Igor Froehner¹, Pedro Serpa¹, Rafael Granza¹,

¹Universidade Estadual de Santa Catarina (UDESC)

R. Paulo Malschitzki, 200 - Zona Industrial Norte, Joinville - SC, 89219-710

[igoor.sf14, pedroserpah, rafaelgranzademello]@gmail.com

1. Introdução

Durante os últimos anos, a Formula 1 vem sendo uma modalidade de automobilismo muito inovadora, não só quando a tecnologia dos carros e equipes, mas também quanto a tecnologia envolvida nas transmissões de TV. No ano de 2020 as transmissões receberam uma nova funcionalidade que impressionou muitos fãs: em alguns momentos da corrida, quando a câmera estava no ponto de vista do piloto, um efeito visual mostra o carro da frente ressaltado por uma marcação na pista e uma informação em cima do carro – que pode ser o nome do piloto, a velocidade, etc – isso pode ser visto na Figura 1. Tal funcionalidade envolve conceitos interessantes de processamento de imagem, entre eles o objetivo desse trabalho: o **rastreio**.



Figura 1. Exemplo F1

Sendo assim, este trabalho tem como objetivo realizar rastreio – que consiste de evidenciar o rastro – de um carro de formula 1 no ambiente de uma pista de corrida usando um método de *template matching*. Para tal, será necessário fazer um comparativo entre os métodos disponíveis na biblioteca OpenCV, e verificar qual melhor se aplicou ao cenário em questão. Após esse comparativo, o método deve ser aplicado, tendo como resultado um vídeo com o rastreio do objeto na cena.

1.1. Introdução Conceitual

O método **CCOEFF** utiliza como seletor o coeficiente de correlação, isto é, nível da correlação é dado conforme o movimento das variáveis de cor e intensidade. Já o método

CCORR leva em consideração a correlação cruzada. A correlação cruzada mede a similaridade entre duas ou mais séries. Por fim, também é testado o método **SQDIFF**, o qual mede a semelhança entre quadrados das variáveis, levando em conta cor e intensidade [2]. E para cada um desses métodos foi também testado suas versões normalizadas.

2. Experimentos

O trabalho foi feito utilizando a linguagem de programação Python (3.8) e a biblioteca OpenCV (4.5.5.62), executado em uma plataforma Ubuntu Linux. Como base de dados para esta etapa foram extraídos 250 frames de uma parte de 10 segundos de um vídeo do youtube, que mostra os carros percorrerem a icônica curva "Eau Rouge" no circuito de Spa - Belgica [1]. O vídeo usado pode ser encontrado nesse link: <https://www.youtube.com/watch?v=hpvuu5MfaSk>. Abaixo é possível ver alguns exemplos de imagens as quais os métodos foram aplicados.



Figura 2. Exemplos de Imagens

É importante perceber que algumas características do vídeo selecionado não facilitam o uso da técnica de *template matching*, a câmera não é estática, o carro pode ficar em posições diferentes no decorrer da passagem e a distância do carro pode variar bastante. Tendo isso em mente, talvez algumas técnicas adicionais podem ser

Para isso, comparou-se os seguintes métodos de template matching disponíveis na biblioteca OpenCV(cv2):

- cv.TM_CCOEFF_NORMED
- cv.TM_CCORR_NORMED
- cv.TM_SQDIFF_NORMED

Para analisar a precisão de cada um dos métodos de template matching, definiu-se uma limiar de 70% de acurácia e então calculou-se a Taxa de Verdadeiro Positivo (TPR) para cada uma dos métodos. Por último, foi usada uma tabela de confusão para clarificar os resultados e decidir qual é o melhor método para o problema proposto por este trabalho.

2.1. Cálculo da precisão de cada teste

Método	Acertos		Erros	
	Absoluto	%	Absoluto	%
cv.TM_CCOEFF_NORMED	10	59	7	41
cv.TM_CCORR_NORMED	8	47	9	53
cv.TM_SQDIFF_NORMED	3	18	14	82

Tabela 1. cv.TM_CCOEFF confusion table

Concluiu-se, portanto, que o método a ser utilizado seria cv.TM_CCOEFF_NORMED.

3. Resultados

A partir das informações obtidas na seção Experimentos, decidiu-se que seria utilizado o método cv.TM_CCOEFF_NORMED prosseguir com o estudo.

Criou-se então um vídeo contendo a predição do trajeto de carros de Formula 1 percorrendo a curva Eau Rouge. Agora pode-se obter resultados da predição e acompanhar o template matching aplicado em um vídeo verdadeiro.

Eis o resultado:

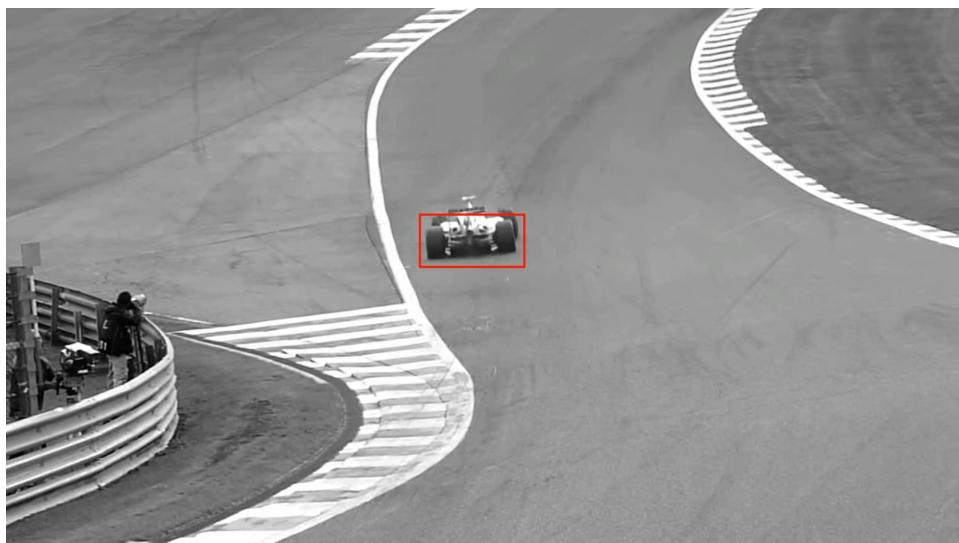


Figura 3. Resultado obtido a partir da aplicação do cv.TM_CCOEFF_NORMED em um vídeo com templates

Também houveram casos de falso positivo no vídeo gerado, mostrando que a acurácia do método não é completamente correta. Principalmente em um caso onde o plano de fundo é um pouco mais complexo, com bastante diferença na intensidade das cores.

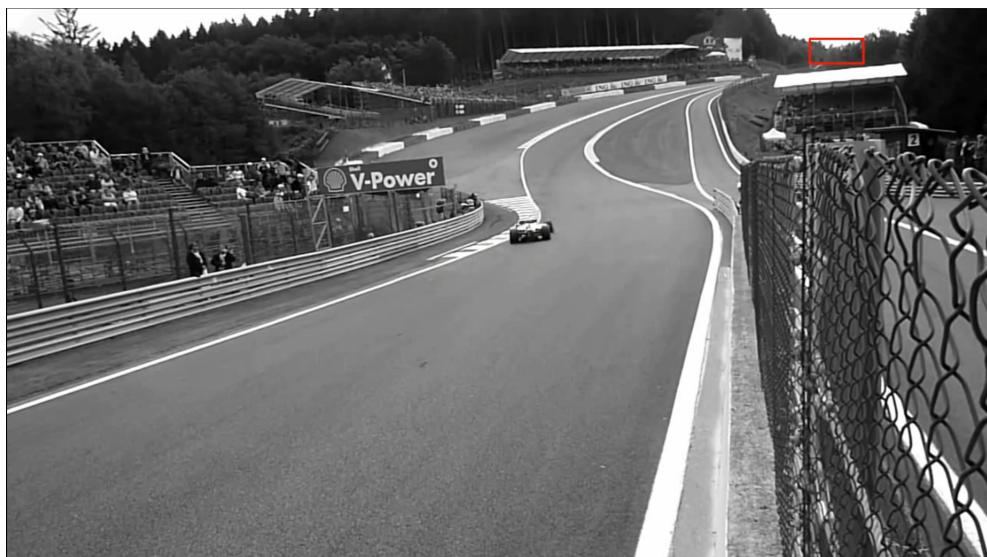


Figura 4. Erro na detecção do carro

4. Conclusão

Os métodos de detecção por template CCOEFF e CCORR se mostraram bastante eficientes quando bem calibrado o limiar. Apesar do método CCOEFF possuir taxa de verdadeiro positivo maior, alguns casos de positivo previsto se mostrou baixo. Já o método CCORR obteve taxa de verdadeiro positivo mais elevado e mesmo que o recall seja alto, isso foi contornado com melhor calibração do limiar. A estratégia de extrair os frames do vídeo e então rodar o algoritmo de detecção sob os quase 300 quadros se mostrou eficaz e válida.

Referências

- [1] FORMULA1. Belgian grand prix - f1 race - circuit de spa-francorchamps: Formula 1®, 2020.
- [2] KAEHLER, A., AND BRADSKI, G. Learning opencv 3: computer vision in c++ with the opencv library, 2016.