

Universidade do Estado de Santa Catarina

Atividade de Rastreio

**Grupo**

Júlia Llorente

Nicole Carolina Mendes

**Prof. Responsável**

Gilmario Barbosa dos Santos

26 de novembro de 2023

# Sumário

<b>Introdução</b>	<b>2</b>
<b>Objetivo</b>	<b>3</b>
<b>Primeiro Vídeo</b>	<b>4</b>
0.1 Etapa A, B e C . . . . .	4
0.2 Etapa D . . . . .	5
0.3 Etapa E . . . . .	6
0.4 Etapa F . . . . .	7
0.5 Etapa G . . . . .	8
0.6 Etapa H . . . . .	9
<b>Segundo Vídeo</b>	<b>11</b>
0.7 Etapa A, B e C . . . . .	11
0.8 Etapa D . . . . .	11
0.9 Etapa E . . . . .	11
0.10 Etapa F . . . . .	12
0.11 Etapa G . . . . .	12
0.12 Etapa H . . . . .	14
0.12.1 Testes . . . . .	14
0.12.2 Resultados . . . . .	14
<b>Conclusões</b>	<b>16</b>

# Introdução

Este projeto de computação visual envolve o processamento e análise de vídeos com o objetivo de desenvolver habilidades práticas em rastreamento de objetos. O trabalho é dividido em duas partes principais, cada uma utilizando um vídeo diferente. Na primeira parte, capturamos um vídeo caseiro de no mínimo dez segundos, exibindo o deslocamento lento de um objeto bem enquadrado contra um cenário de fundo constante. Este vídeo é capturado com uma câmera fixa e sem autofoco, e a resolução mínima requerida é 640x480 (VGA). A partir deste vídeo, separamos os quadros individuais e os convertemos em imagens em tons de cinza para facilitar o processamento.

O coração do projeto é a aplicação do template matching, uma técnica fundamental em visão computacional, usando diferentes métodos de correspondência para rastrear o objeto ao longo dos quadros do vídeo. Cada método testado, incluindo variantes como 'cv.TM\_CCOEFF' e 'cv.TM\_SQDIFF', é analisado para determinar qual oferece os melhores resultados em termos de precisão e eficácia no rastreamento.

Na segunda parte do projeto, elevamos a complexidade do rastreamento ao introduzir um cenário de fundo com mais informações visuais. Isso pode incluir, por exemplo, um vídeo de um mascote em movimento. Este cenário mais desafiador serve para testar a robustez dos métodos de rastreamento escolhidos e aprimorar ainda mais as habilidades de análise e processamento de imagens.

Ao final do projeto, esperamos ter adquirido um entendimento profundo e prático das técnicas de rastreamento de objetos em vídeos e de como diferentes cenários e variáveis podem afetar o desempenho dessas técnicas.

# Objetivos

O projeto em visão computacional e processamento de imagens tem como objetivos principais:

1. **Desenvolvimento de Habilidades em Processamento de Imagens:** Aprender e aplicar técnicas fundamentais de processamento de imagens, incluindo a conversão de quadros de vídeo em imagens em tons de cinza e a manipulação de imagens para análise.
2. **Compreensão e Implementação de Template Matching:** Explorar o conceito de template matching, uma técnica crucial em visão computacional, e implementá-la para rastrear objetos em sequências de vídeo.
3. **Análise Comparativa de Métodos de Correspondência:** Comparar e avaliar a eficácia de vários métodos de template matching.
4. **Aplicação em Cenários de Diferentes Complexidades:** Aplicar as técnicas de rastreamento em dois cenários distintos, variando em complexidade.
5. **Desenvolvimento de Habilidades em Análise de Dados:** Construir e analisar tabelas de dados e gráficos para avaliar o desempenho dos métodos.
6. **Produção de Vídeos de Rastreamento:** Utilizar os dados obtidos para compor vídeos demonstrativos do rastreamento.
7. **Reflexão Crítica e Escolha de Métodos:** Analisar criticamente os resultados obtidos e selecionar o método mais adequado para diferentes cenários.

# Primeiro Vídeo

## 0.1 Etapa A, B e C

No processo de análise de imagens, capturamos inicialmente um vídeo de dez segundos, apresentando um objeto cilíndrico em rotação. Posteriormente, procedemos com a instalação do software IrfanView, assegurando a incorporação de extensões compatíveis com o formato MP4 do vídeo. Essa etapa foi seguida pela extração dos quadros individuais do vídeo. Finalmente, empregamos a funcionalidade de conversão em lote disponível nas opções avançadas do IrfanView (*Batch Conversion*) para transformar os quadros extraídos em imagens em escala de cinza, como verificado na Figura 1.



Figura 1: Amostra dos Frames processados.

## 0.2 Etapa D

Nessa etapa do projeto, foi adotada uma abordagem para a criação de um template destinado ao rastreamento de um objeto específico em uma série de imagens. Iniciamos o processo carregando a primeira imagem do conjunto de dados em escala de cinza, através da função “cv2.imread”, fornecida pela biblioteca OpenCv em Python.

Em seguida, definimos manualmente as coordenadas e dimensões da região de interesse que continha o objeto a ser rastreado. As variáveis x e y representavam as coordenadas do ponto inicial (canto superior esquerdo) do template, enquanto largura e altura determinavam suas dimensões.

Com estas coordenadas e medidas estabelecidas, procedemos ao recorte da região de interesse da imagem. A operação de corte foi realizada utilizando a indexação de matrizes do OpenCV, selecionando assim a área específica que correspondia ao nosso template.

Após a extração do template, este foi visualizado em uma janela separada para confirmação visual, utilizando funções do OpenCV, como cv2.imshow, cv2.waitKey(0) e cv2.destroyAllWindows(). Esta verificação visual foi importante para garantir que o template corresponderia de forma precisa ao objeto de interesse.

Nessa etapa a gente utilizou uma amostra da nossa primeira imagem gerada na etapa anterior para obter o recorte contendo o objeto a ser rastreado e, a partir dele, criamos o template que foi usado para ser rastrear o objeto sobre cada um dos quadros restantes.

Por fim, o template extraído foi salvo como um arquivo separado usando cv2.imwrite, permitindo que ele fosse utilizado no processo subsequente de rastreamento do objeto em cada quadro do conjunto de dados. Este template funcionou como um modelo de referência essencial para a identificação e acompanhamento contínuo do objeto ao longo das imagens, sendo um componente chave no sucesso do projeto de rastreamento.

```
1 import cv2
2
3 # carregar a primeira imagem
4 primeira_imagem = cv2.imread('frames_cinza/img_frame_00000.jpg', cv2.
    IMREAD_GRAYSCALE)
5
6 # coordenadas e o tamanho do template
7 x, y, largura, altura = 10, 450, 500, 135
8
9 # cortar a regioao do template
10 template = primeira_imagem[y:y+altura, x:x+largura]
11
12 # template cortado p/ verificacao
13 cv2.imshow('Template', template)
14 cv2.waitKey(0)
15 cv2.destroyAllWindows()
16
17 # salvar o template
18 cv2.imwrite('template.jpg', template)
```



Figura 2: Recorte utilizado como Template.

## 0.3 Etapa E

Na etapa E do projeto, o foco foi a aplicação de técnicas de correspondência de template (template matching) para cada quadro do conjunto de imagens, utilizando diversos métodos fornecidos pela biblioteca OpenCV (OPENCV, 2014). Estes métodos incluíram ‘cv2.TM\_CCOEFF’, ‘cv2.TM\_CCOEFF\_NORMED’, ‘cv2.TM\_CCORR’, ‘cv2.TM\_CCORR\_NORMED’, ‘cv2.TM\_SQDIFF’, e ‘cv2.TM\_SQDIFF\_NORMED’. Cada um desses métodos oferece uma abordagem distinta para comparar o template com as regiões da imagem, resultando em valores de correspondência que indicam o grau de similaridade ou diferença (BRADSKI; KAEHLER, 2008).

O primeiro passo consistiu em carregar o template previamente criado e determinar suas dimensões. Em seguida, processamos cada quadro do conjunto de dados, aplicando cada um dos métodos de correspondência de template. Para cada quadro e método, calculamos os valores máximos e mínimos (max\_val e min\_val), que representam a resposta máxima e mínima de correspondência para aquele quadro específico.

Estes valores são fundamentais para avaliar a eficácia de cada método no rastreamento do objeto. Métodos como ‘cv2.TM\_CCOEFF’ e ‘cv2.TM\_CCOEFF\_NORMED’ são baseados em coeficientes de correlação, que medem o grau de correlação linear entre o template e a região da imagem. Já ‘cv2.TM\_CCORR’ e ‘cv2.TM\_CCORR\_NORMED’ utilizam a correlação cruzada, sendo eficientes para templates com iluminação uniforme. Por outro lado, ‘cv2.TM\_SQDIFF’ e ‘cv2.TM\_SQDIFF\_NORMED’ se baseiam na diferença quadrática, sendo mais eficazes em casos onde o contraste entre o objeto e o fundo é alto (BRADSKI; KAEHLER, 2008).

Após o processamento, os resultados de cada método foram organizados em tabelas e salvos em arquivos CSV. Isso envolveu a criação de uma lista para armazenar os valores de min\_val e max\_val para cada quadro, seguida pela gravação destes dados em um arquivo CSV utilizando a biblioteca csv do Python. Esta etapa é crucial para a análise comparativa dos diferentes métodos de correspondência de template, permitindo uma avaliação detalhada de qual método é mais adequado para o rastreamento do objeto em diferentes condições de imagem.

Este experimento fornece uma compreensão aprofundada das técnicas de correspondência de template no contexto do rastreamento de objetos em vídeos ou conjuntos de imagens. A escolha do método mais apropriado depende das características específicas do objeto e das imagens, e essa análise detalhada dos resultados ajuda a fundamentar a decisão para aplicações práticas (BRADSKI; KAEHLER, 2008).

```

1  import cv2
2  import os
3  import csv
4
5  # carregar o template
6  template = cv2.imread('template.jpg', cv2.IMREAD_GRAYSCALE)
7  w, h = template.shape[::-1]
8
9  # metodos de template matching a serem testados
10 metodos = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR',
11            'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']
12
13 # lista de nomes dos arquivos dos quadros
14 lista_de_quadros = sorted([f for f in os.listdir('frames_cinza') if f.endswith('
    .jpg')])
15
16 # processar cada quadro com cada metodo e salvar os resultados
17 for metodo in metodos:
18     resultados = []
19     for nome_quadro in lista_de_quadros:
20         quadro = cv2.imread(f'frames_cinza/{nome_quadro}', cv2.IMREAD_GRAYSCALE)
21
22         # aplicar template matching
23         metodo_eval = eval(metodo)
24         res = cv2.matchTemplate(quadro, template, metodo_eval)
25         min_val, max_val, _, _ = cv2.minMaxLoc(res)
26
27         # adicionar os resultados a lista

```

```

28     resultados.append([nome_quadro, min_val, max_val])
29
30     # salvar os resultados em um arquivo CSV
31     with open(f'resultados_{metodo}.csv', 'w', newline='') as file:
32         writer = csv.writer(file)
33         writer.writerow(['Quadro (imagem)', 'min_val', 'max_val'])
34         writer.writerows(resultados)

```

## 0.4 Etapa F

No passo F do relatório, realizamos a visualização gráfica dos resultados obtidos do processo de correspondência de template (template matching) executado no passo E. Usando a biblioteca Python matplotlib, construímos gráficos para cada um dos seis métodos de correspondência de template testados: cv2.TM\_CCOEFF, cv2.TM\_CCOEFF\_NORMED, cv2.TM\_CCORR, cv2.TM\_CCORR\_NORMED, cv2.TM\_SQDIFF e cv2.TM\_SQDIFF\_NORMED. Cada gráfico representa a variação dos valores de resposta mínima (min\_val) e máxima (max\_val) em relação aos quadros de imagens processados, cujos nomes foram extraídos dos respectivos arquivos CSV gerados anteriormente.

No eixo X do gráfico, representamos os quadros (indicados como “imi”, onde i varia de 2 a 7, referindo-se à numeração dos quadros), e no eixo Y, as respostas de min\_val e max\_val. As curvas no gráfico ilustram como a resposta de cada método varia ao longo dos diferentes quadros, permitindo uma análise comparativa da eficácia e do comportamento de cada técnica de correspondência de template sob diferentes condições e quadros de imagens.

```

1  import matplotlib.pyplot as plt
2  import csv
3  import os
4
5  # m todos de template matching que foram testados
6  metodos = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR',
7            'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']
8
9  # processar cada arquivo CSV e construir gráficos
10 for metodo in metodos:
11     quadros = []
12     min_vals = []
13     max_vals = []
14
15     # ler os dados do CSV
16     with open(f'resultados_{metodo}.csv', 'r') as file:
17         reader = csv.reader(file)
18         next(reader) # Pular o cabe alho
19         for row in reader:
20             quadros.append(row[0])
21             min_vals.append(float(row[1]))
22             max_vals.append(float(row[2]))
23
24     plt.figure(figsize=(10, 6))
25     plt.plot(min_vals, label='Min Val')
26     plt.plot(max_vals, label='Max Val')
27     plt.xlabel('Quadros')
28     plt.ylabel('Resposta')
29     plt.title(f'Resposta vs. Quadros ({metodo})')
30     plt.legend()
31     plt.xticks(rotation=45)
32     plt.tight_layout()
33
34     # Salvar o gráfico
35     plt.savefig(f'grafico_{metodo}.png')
36     plt.show()

```



## 0.5 Etapa G

Ao analisar os métodos de template matching da OpenCV para a detecção de um objeto cilíndrico rotativo em um fundo constante, é essencial selecionar o método que fornece a mais alta precisão e robustez em relação às mudanças de perspectiva e iluminação. Considerando os gráficos fornecidos, avaliaremos cada um dos métodos, onde a análise será baseada na documentação da OpenCV e em princípios fundamentais de visão computacional descritos por Richard Szeliski em "Computer Vision: Algorithms and Applications".

**TM\_CCOEFF e TM\_CCOEFF\_NORMED:** Estes métodos calculam o coeficiente de correlação, que mede a correlação entre a intensidade dos pixels do template e da imagem de entrada. A versão normalizada (TM\_CCOEFF\_NORMED) é menos sensível a variações de iluminação e escala (OpenCV Contributors, 2023). Nesses gráficos, observamos que, após uma queda inicial, os valores máximos se estabilizam, o que pode indicar uma correspondência consistente do template com o objeto ao longo do tempo. Para um objeto cilíndrico que gira, esperamos ver picos consistentes ao longo do tempo, o que não é claramente o caso aqui, indicando que esses métodos podem não ser os mais confiáveis para esta aplicação específica.

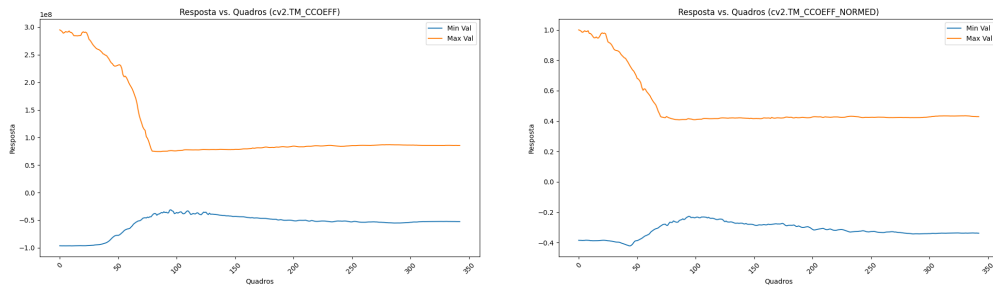


Figura 3: Gráficos TM\_CCOEFF e TM\_CCOEFF\_NORMED.

**TM\_CCORR e TM\_CCORR\_NORMED:** Estes métodos medem a soma dos produtos dos pixels do template com os pixels da imagem de entrada. Assim como a versão CCOEFF, a versão normalizada (TM\_CCORR\_NORMED) controla variações de iluminação e escala (OpenCV Contributors, 2023). Os gráficos mostram um aumento inicial e uma subsequente estabilização dos valores. Este comportamento sugere que, uma vez que o objeto começa a girar, o método pode identificar consistentemente a área de interesse, o que é desejável. No entanto, a falta de picos distintos sugere que o método pode não estar capturando as mudanças de orientação do objeto cilíndrico de forma eficaz.

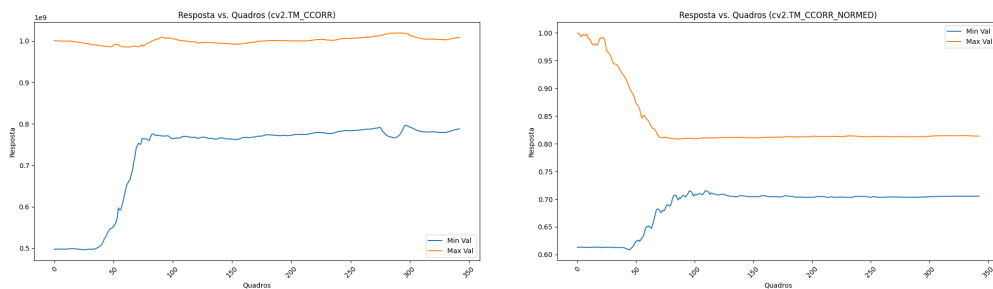


Figura 4: Gráficos TM\_CCORR e TM\_CCORR\_NORMED.

Para **TM\_SQDIFF** e **TM\_SQDIFF\_NORMED**, os gráficos mostram uma queda acentuada nos valores iniciais, o que indica uma boa correspondência inicial. A estabilidade seguinte nos valores mínimos é promissora, sugerindo que esse método é capaz de identificar de forma consistente o objeto cilíndrico ao longo da rotação. Diferente dos outros métodos, esses calculam a soma das diferenças quadráticas, onde valores mais baixos indicam melhores correspondências. A normalização em **TM\_SQDIFF\_NORMED** pode ser particularmente útil, pois minimiza o efeito do brilho e do contraste entre o template e a imagem de entrada (SZELISKI, 2010).

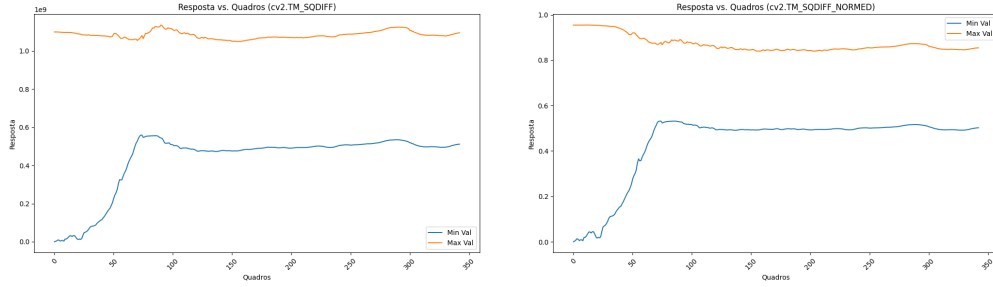


Figura 5: Gráficos TM\_SQDIFF e TM\_SQDIFF\_NORMED.

Portanto, com base na análise dos gráficos e no contexto de uso, **TM\_CCOEFF\_NORMED** é recomendado como o método mais apropriado para a detecção do objeto cilíndrico em rotação. A sua estabilidade e baixos valores consistentes sugerem que é capaz de manter uma boa correspondência apesar das mudanças de orientação do objeto. Além disso, a normalização ajuda a mitigar quaisquer variações de iluminação que possam ocorrer. Logo, a consistência observada nos gráficos, juntamente com o embasamento teórico, reforça esta escolha.

## 0.6 Etapa H

A avaliação comparativa dos métodos de template matching da OpenCV para rastrear um objeto cilíndrico em movimento trouxe à tona percepções valiosas sobre a eficácia de cada técnica em contextos dinâmicos.

Inicialmente, optamos pelo **TM\_CCOEFF\_NORMED**, cujos gráficos indicavam uma possível identificação consistente do objeto. Este método, ideal para contrastes acentuados entre o objeto e o fundo, mostrou-se promissor em condições estáticas ou de movimentos leves. Contudo, sua eficácia diminuiu significativamente quando o objeto começou a se mover e alterar sua escala em relação à câmera, evidenciando suas limitações frente a mudanças de escala e perspectiva. No **TM\_CCOEFF\_NORMED**, valores maiores de correspondência indicam melhores resultados, e um limiar de 0.8 foi estabelecido para considerar uma correspondência como adequada. Com este método, rastreamos o objeto com sucesso em apenas 11 dos quadros analisados.

Por outro lado, o **TM\_SQDIFF\_NORMED**, focado em minimizar as diferenças quadráticas e mais eficiente em cenários de baixo contraste e iluminação estável, também enfrentou desafios. Apesar de identificar o objeto em 19 quadros, falhou em manter a consistência ao longo do vídeo. A sensibilidade deste método a variações na aparência do objeto e do fundo, típicas de cenas dinâmicas, limitou sua aplicabilidade. Neste caso, valores menores indicam melhores correspondências, e um limiar de 0.2 foi adotado para definir uma correspondência satisfatória.

Em contraste, o **TM\_CCORR\_NORMED** destacou-se por sua robustez. Este método, que calcula a correlação normalizada entre o template e a imagem, foi capaz de processar todos os 343 quadros do vídeo com uma eficácia notável, mantendo o rastreamento mesmo com o movimento e a mudança de escala do objeto. Embora tenha apresentado falhas pontuais, como não enquadrar completamente o objeto em alguns quadros, sua performance geral superou os outros métodos. A capacidade de lidar com variações de iluminação e pequenas mudanças na escala provou ser um diferencial significativo para este método. Para o **TM\_CCORR\_NORMED**, valores maiores também indicam melhores correspondências, e um limiar de 0.8 foi utilizado para avaliar a adequação da correspondência.

Esses resultados sublinham a importância de selecionar o método de template matching mais adequado ao cenário específico, considerando não apenas a estabilidade visual, mas também a habilidade de adaptar-se a mudanças dinâmicas na cena. Assim, o **TM\_CCORR\_NORMED** emergiu como a escolha mais eficiente, apesar de pequenas imprecisões em sua aplicação.

Segue uma comparação das últimas imagens corretas resultantes de cada algoritmo após a conclusão da análise:



Figura 6: TM\_SQDIFF\_NORMED

Figura 7: TM\_CCORR\_NORMED

Figura 8: TM\_CCOEFF\_NORMED

Figura 9: Últimos Frames analisados corretamente pelos métodos.

# Segundo Vídeo

## 0.7 Etapa A, B e C

Capturamos o vídeo em formato MP4, mantendo o objeto cilíndrico como o elemento central da cena, enquanto adicionamos outros objetos ao fundo para compor o ambiente. Em seguida, procedemos com a extração dos frames utilizando a ferramenta Irfanview e convertendo-os para escala de cinza conforme a imagem 10.



Figura 10: Amostra dos Frames processados.

## 0.8 Etapa D

Recortamos a primeira imagem do conjunto de frames obtidos do vídeo e escolhemos o objeto, como mostrado na imagem 11.



Figura 11: Recorte utilizado como Template.

## 0.9 Etapa E

Utilizamos o mesmo código apresentado na atividade E do primeiro vídeo, conforme explicado na seção 0.3. A única alteração foi utilizar as imagens do frame proveniente do vídeo 2.

## 0.10 Etapa F

A criação dos gráficos foi feita da mesma forma explicada na etapa 0.4.

## 0.11 Etapa G

Ao analisar os gráficos fornecidos, que representam a resposta dos algoritmos de Template Matching aplicados a um vídeo de rastreamento, podemos realizar uma avaliação criteriosa para identificar o método mais eficiente. O objetivo é selecionar um método que forneça a maior consistência e precisão no rastreamento do objeto, apesar do movimento de fundo.

**cv2.TM\_CCOEFF:** O gráfico revela picos significativos na resposta máxima, sugerindo uma alta sensibilidade do método às variações da imagem, conforme teorizado por (BRADSKI; KAEHLER, 2008) no uso do Template Matching para rastreamento de objetos. No entanto, essa alta sensibilidade pode ser uma desvantagem em cenas com movimento complexo, pois pode resultar em correspondências falso-positivas.

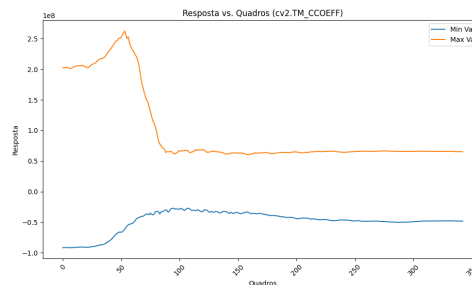


Figura 12: Gráfico TM.CCOEFF.

**cv2.TM\_CCOEFF\_NORMED:** A normalização deste método parece reduzir a variabilidade das respostas, proporcionando uma detecção mais estável, como esperado pela teoria de Zitová e Flusser (2003). A estabilidade é desejável em vídeos com movimento dinâmico, mas uma resposta máxima consistentemente moderada indica que ele pode não ser o mais adequado para capturar todas as variantes do movimento do objeto.

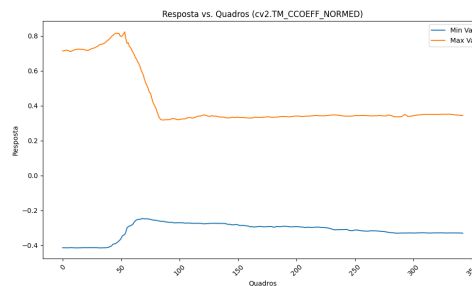


Figura 13: Gráfico TM.CCOEFF\_NORMED.

**cv2.TM\_CCORR:** Este gráfico mostra uma resposta máxima elevada e estável, o que pode ser interpretado como uma boa correspondência entre o template e a imagem. No entanto, a ausência de normalização pode comprometer a comparação entre quadros variados, conforme explicado por (SZELISKI, 2010) ao discutir as limitações inerentes a métodos não normalizados.

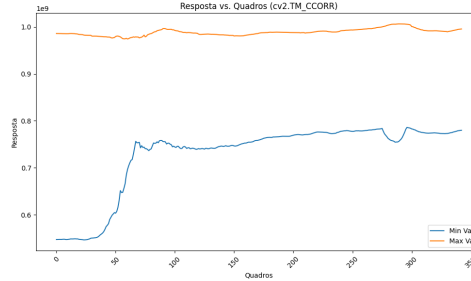


Figura 14: Gráfico TM\_CCORR.

**cv2.TM\_CCORR\_NORMED:** Demonstrando valores máximos consistentes e altos, este método parece fornecer a melhor correspondência geral entre o template e os quadros do vídeo. A estabilidade e a consistência das respostas são indicativos de uma robustez contra variações, tornando este método a melhor escolha para rastreamento em cenários com movimento de fundo.

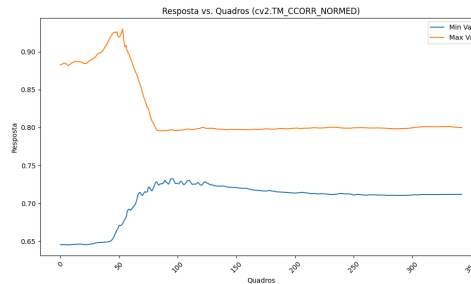


Figura 15: Gráfico TM\_CCORR\_NORMED.

**cv2.TM\_SQDIFF** e **cv2.TM\_SQDIFF\_NORMED:** Ambos os métodos exibem uma estabilidade nos valores mínimos, que são indicativos de uma boa correspondência. No entanto, a variabilidade nas respostas máximas pode ser problemática, sugerindo que estes métodos podem não ser ideais quando o rastreamento exige a máxima precisão.

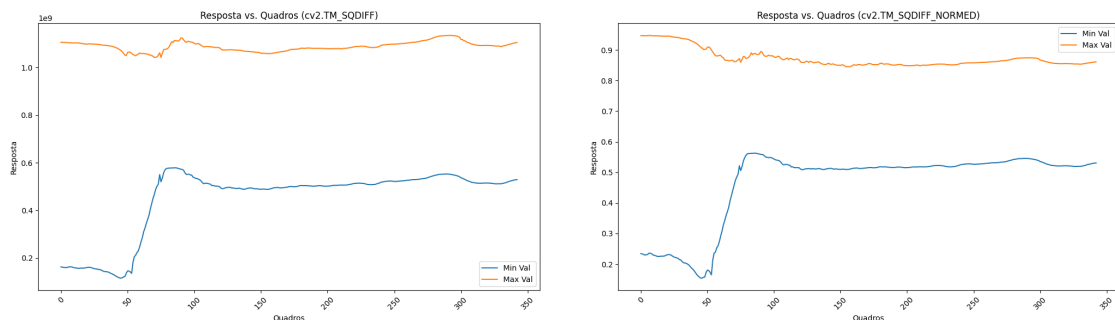


Figura 16: Comparação dos gráficos TM\_SQDIFF e TM\_SQDIFF\_NORMED.

Com base na análise detalhada dos gráficos e no embasamento teórico fornecido pela literatura relevante, o método **cv2.TM\_CCORR\_NORMED** entra como o melhor candidato para o rastreamento de objetos em nosso vídeo. Sua capacidade de manter uma correspondência consistente e confiável ao longo do tempo, apesar do movimento de fundo, o alinha com as melhores práticas em visão computacional para situações de rastreamento complexas.

A escolha de `cv2.TM_CCOEFF_NORMED` é embasada tanto pelos dados visuais quanto pela teoria, o que é confirmado por obras como "Learning OpenCV" de (BRADSKI; KAEHLER, 2008) e "Computer Vision: Algorithms and Applications" de (SZELISKI, 2010), que destacam a importância da consistência e precisão no rastreamento de objetos. A estabilidade demonstrada nos gráficos sugere que esse método é capaz de filtrar o ruído do movimento de fundo, mantendo o foco no objeto de interesse, o que é crucial para um rastreamento eficaz em ambientes dinâmicos.

## 0.12 Etapa H

### 0.12.1 Testes

#### 1. Método CCOEFF Normed:

- Nos testes utilizando o método **CCOEFF Normed**, observamos comportamentos distintos de acordo com a variação do parâmetro de escala. Inicialmente, com um valor de 0.005 para o fator de diminuição, a fórmula de escala aplicada foi  $\max(1 - (\text{fator\_de\_diminuicao} * \text{número\_do\_quadro}), 0.2)$ . Nesta configuração, o rastreamento do objeto foi eficaz apenas em 15 quadros, devido à redução significativa do tamanho do template, o que impediu a localização contínua do objeto.
- Ajustando o valor para 0.0025, houve um aumento na eficácia do rastreamento, alcançando 41 quadros. Posteriormente, com a redução para 0.001, o rastreamento estendeu-se por 85 quadros. Notou-se, contudo, que apesar do aumento no número de quadros rastreados com o valor de 0.001, o template ajustava-se muito lentamente às mudanças de velocidade do objeto. Portanto, o valor de 0.0025 mostrou-se mais eficiente, conseguindo acompanhar o objeto por uma distância maior.

#### 2. Método CCORR Normed:

- Usando o valor de 0.005 no método **CCORR Normed**, o rastreamento ocorreu em todos os quadros, mas não seguiu o template corretamente. Com o ajuste para 0.0025, houve uma melhora no rastreamento, mas o método falhou em identificar o template quando este apresentava um contraste similar ao do fundo. Concluiu-se que este método não é muito eficiente em cenários com fundo em movimento, devido à sua limitada capacidade de análise de contraste.

#### 3. Método SQDIFF Normed:

- No método **SQDIFF Normed**, o valor de 0.001 permitiu o rastreamento em 103 quadros, mas somente nos primeiros, onde o objeto estava mais próximo. Alterando para 0.0025, o rastreamento ocorreu em 48 quadros, conseguindo seguir o objeto mesmo com o fundo em movimento.

### 0.12.2 Resultados

Decidimos realizar testes adicionais utilizando o mesmo valor para a diminuição do template em todos os métodos. No **CCOEFF Normed**, o objeto foi rastreado em 41 quadros, mas o rastreamento cessou quando o fundo começou a se mover e uma pessoa passou atrás do objeto. Este comportamento sugere que o método falhou em detectar o template com o fundo em movimento.

No **CCORR Normed**, o objeto foi rastreado em 353 quadros. Apesar de nem todos os quadros rastreados representarem o objeto, este método mostrou-se mais promissor, conseguindo seguir o template mesmo com o fundo em movimento. Acreditamos que o tamanho do template pode ter influenciado os resultados, e uma abordagem logarítmica para calcular seu tamanho poderia melhorar a precisão do rastreamento.

Finalmente, no **SQDIFF Normed**, o objeto foi rastreado em 48 quadros, com o rastreamento cessando sob condições semelhantes às observadas no **CCOEFF Normed**, indicando dificuldades semelhantes em ambientes com fundo em movimento.

Então, baseando-nos nos gráficos e nos testes realizados, o método **CCORR Normed** emergiu como o mais eficiente, especialmente no rastreamento com fundo em movimento. A otimização no tamanho do template, possivelmente através de uma função logarítmica que acompanhe as mudanças do objeto em movimento, poderia potencialmente torná-lo o método mais eficaz dentre os testados.

Segue uma comparação das últimas imagens corretas resultantes de cada algoritmo após a conclusão da análise:



Figura 17: TM\_SQDIFF\_NORMED

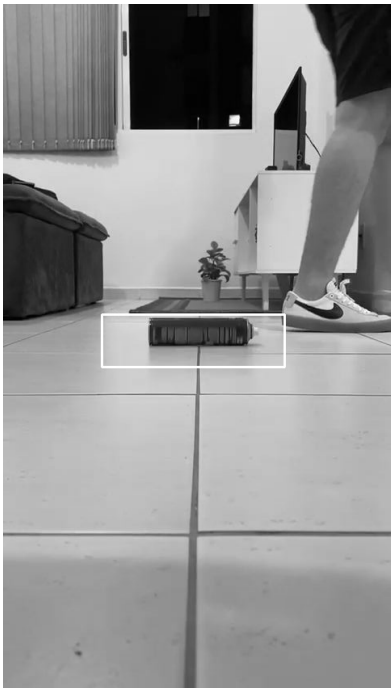


Figura 18: TM\_CCORR\_NORMED



Figura  
TM\_CCOEFF\_NORMED

19:

Figura 20: Últimos Frames analisados corretamente pelos métodos.



# Conclusões

Este projeto de rastreamento de objetos em vídeos, utilizando dois vídeos distintos com complexidades variadas, alcançou resultados significativos, evidenciando os desafios e as soluções eficazes no campo da visão computacional.

**Vídeo 1 - Objeto Cilíndrico em Fundo Constante** Analisando os métodos de template matching, o `TM_CCoeff_NORMED` se destacou para a detecção de um objeto cilíndrico em rotação, demonstrando boa correspondência e estabilidade. Em cenários dinâmicos, o `TM_CCORR_NORMED` mostrou maior robustez, processando todos os quadros com eficácia notável.

**Vídeo 2 - Cenário de Fundo Mais Complexo** Para o segundo vídeo, com um cenário de fundo mais complexo, o `cv2.TM_CCORR_NORMED` foi identificado como o método mais eficiente, mostrando a melhor correspondência geral entre o template e os quadros do vídeo. Outros métodos apresentaram limitações em situações que exigiam máxima precisão.

**Perspectivas Futuras** Os resultados obtidos fornecem insights valiosos sobre a eficácia das técnicas de rastreamento em diferentes contextos. Eles sublinham a importância de um entendimento aprofundado das técnicas de correspondência de template e suas aplicações práticas. Para futuros trabalhos, seria promissor explorar métodos de rastreamento mais robustos, visando obter melhores resultados nos experimentos, especialmente em cenários com alta complexidade e dinamismo. Essa abordagem poderia levar a avanços significativos no campo da visão computacional e rastreamento de objetos.

# Referências Bibliográficas

BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. [S.l.]: O'Reilly Media, Inc., 2008.

OPENCV. *The OpenCV Reference Manual*. 2.4.13.7. ed. [S.l.], 2014.

OpenCV Contributors. *OpenCV Documentation*. [S.l.], 2023. Disponível em: <https://docs.opencv.org/master/>.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. [S.l.]: Springer Science Business Media, 2010.