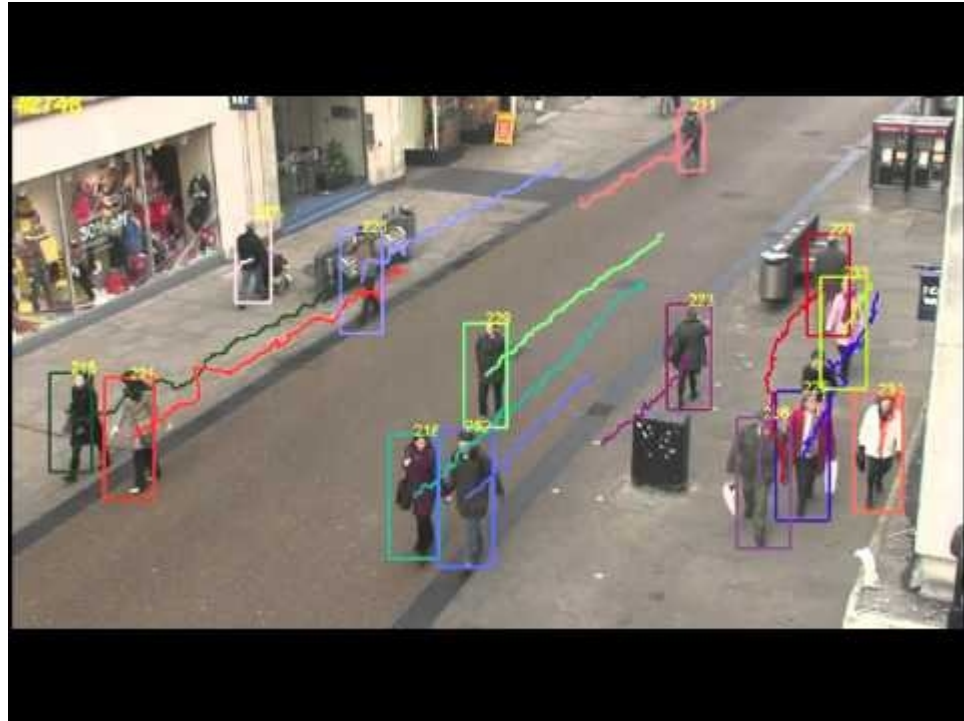
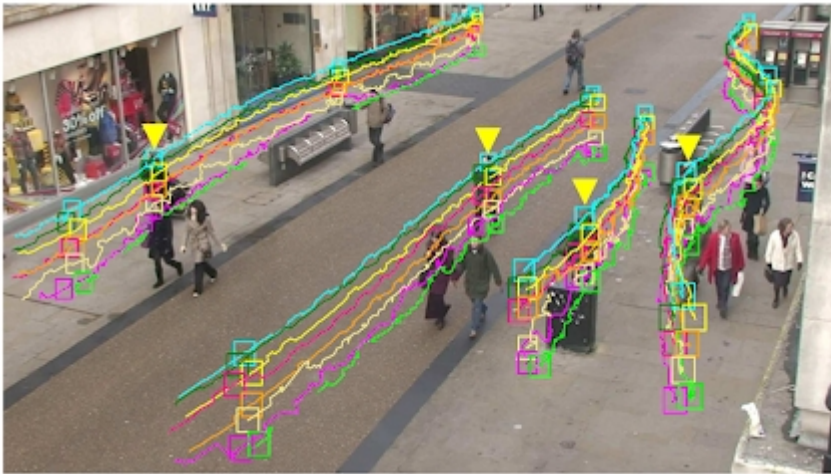


Rastreio/tracking

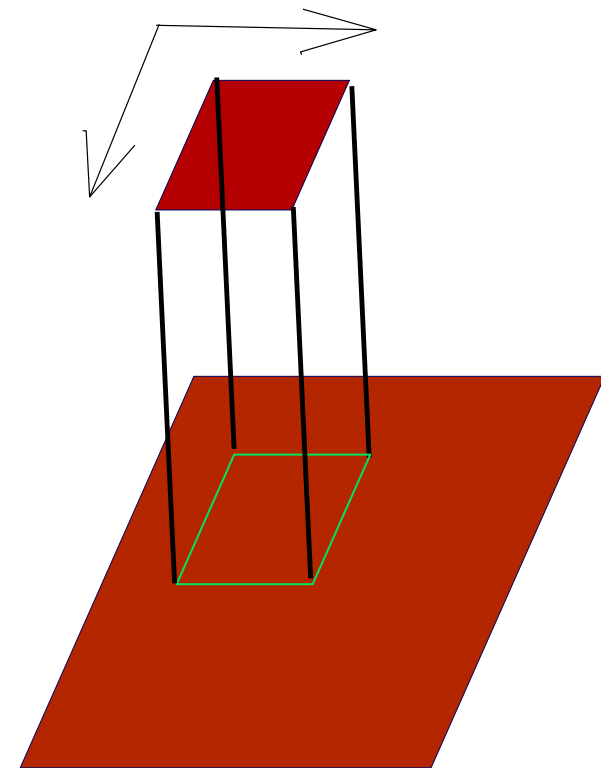
Rastrear é evidenciar o rastro ou a pista de um ou mais objetos em um vídeo, determinando a/as trajetória/as ao longo do tempo.

O rastreio de objetos em imagens é uma operação muito aplicada em sistemas de monitoramento (surveillance)



Template matching

Uma maneira de realizar rastreamento ou detecção de *regions of interest* (ROI's) em uma imagem é utilizar a avaliação do casamento (*matching*) de um padrão (*template*, subimagem) em relação a uma Imagem principal.

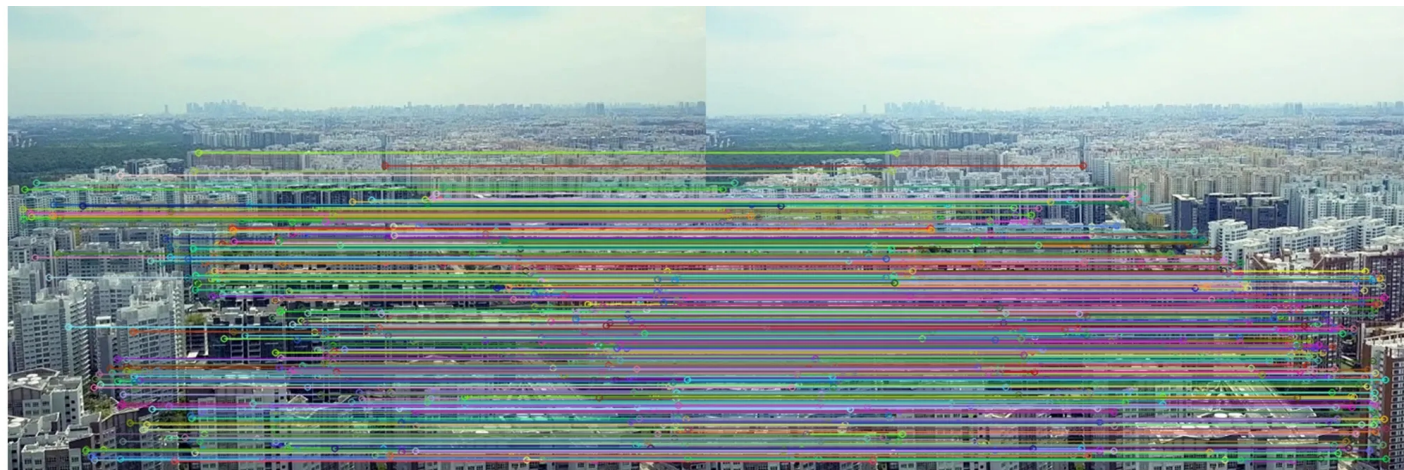


Existem métodos sofisticados e robustos a rotações e escala tais como SURF, ORB, BRIEF, etc, os quais podem ser usados para avaliar o casamento de *template matching*

Referências:

AQUI e **AQUI**

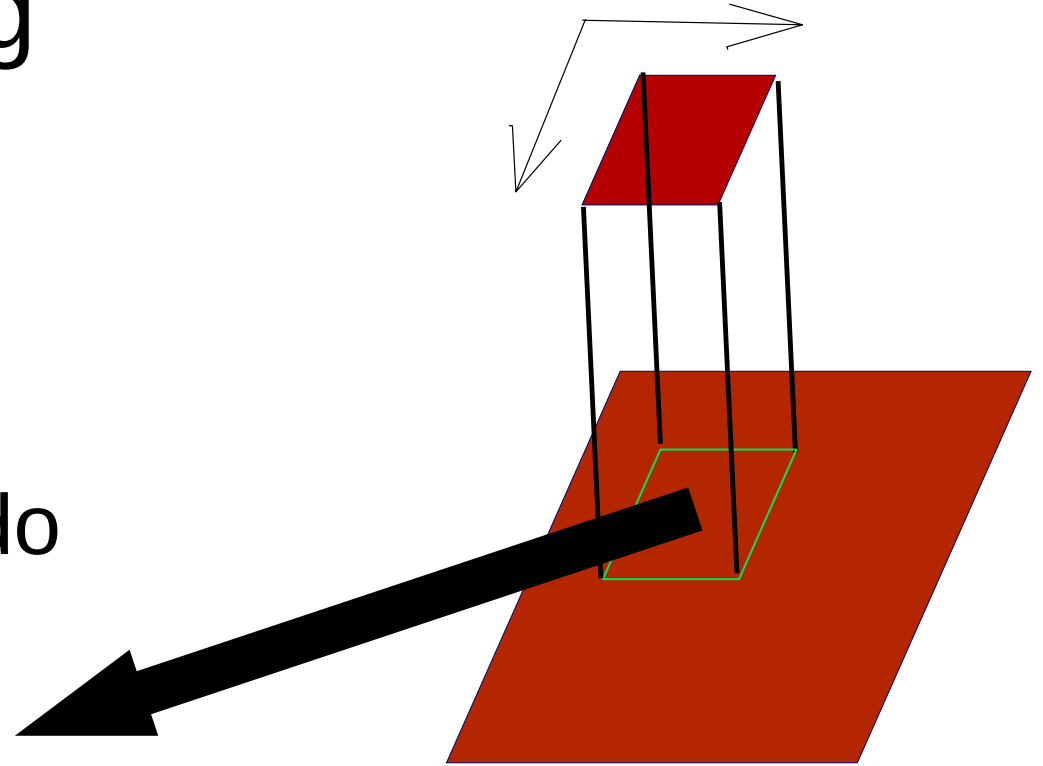
Aqui iremos discutir métodos mais simples!



Template matching

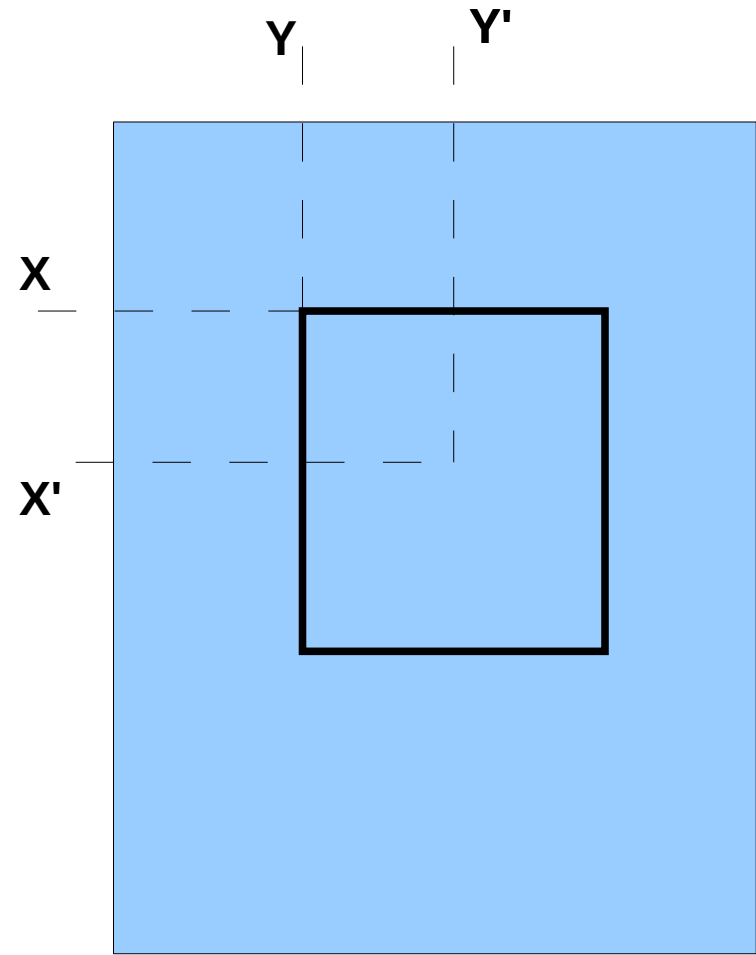
O casamento é avaliado por meio de cálculos realizados para cada posicionamento do *template*;

Existem diversas estratégias para o cálculo do casamento: diferença quadrada, coeficiente de correlação de Pearson, etc.



Template matching

→ Diferença Quadrada

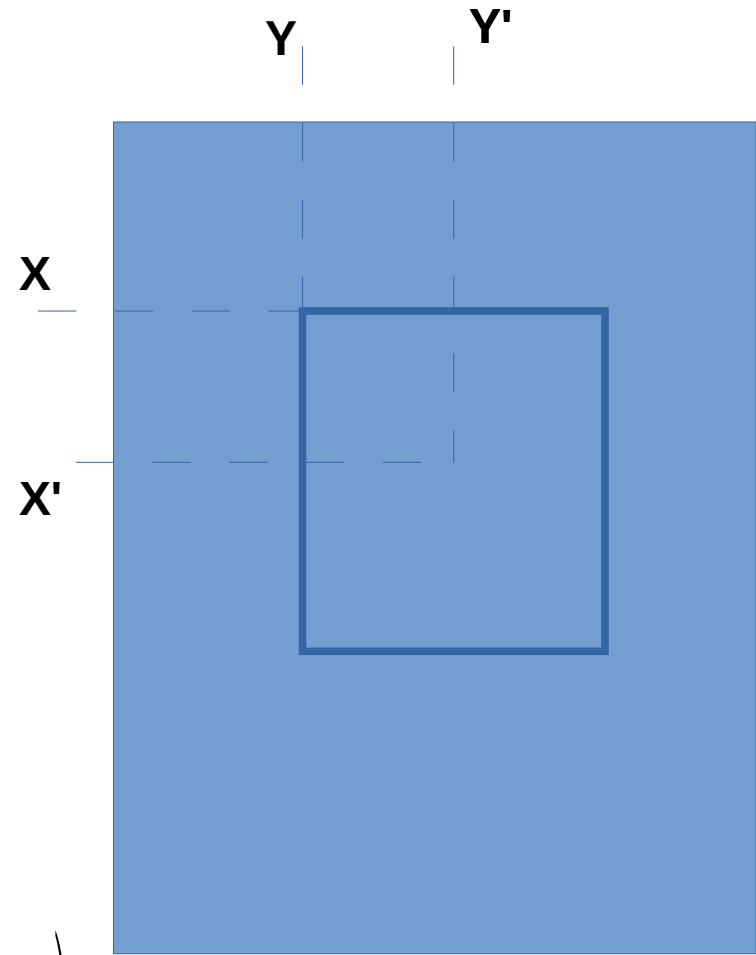


$$R_{SQDIFF} = \sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2$$

Template matching

Existem diversas estratégias para o cálculo do casamento:

→ Coeficiente de Correlação



$$R_{CCOEFF} = \sum_{x', y'} [T'(x', y') - I'(x + x', y + y')]^2$$

$$T'(x', y') = T(x', y') - \left(\frac{1}{(w * h) \sum_{x'', y''} T(x'', y'')} \right)$$

$$I'(x + x', y + y') = I(x + x', y + y') - \left(\frac{1}{(w * h) \sum_{x'', y''} I(x'', y'')} \right)$$

A biblioteca OpenCV

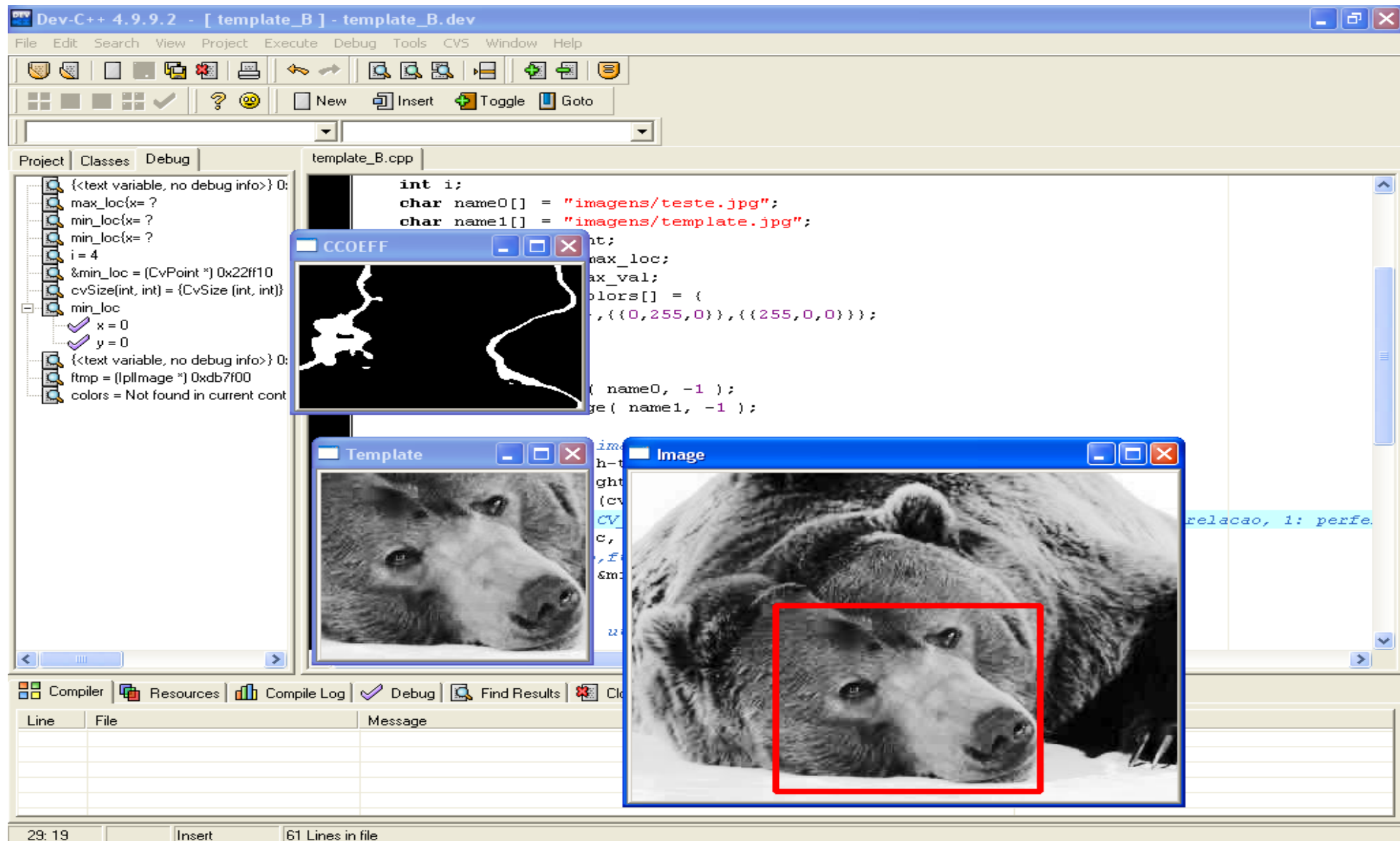
OpenCV (Open Source Computer Vision Library):

cvMatchTemplate(src,temp,resp,método);

Parâmetros:

- src: imagem de entrada;
- temp: template (subimagem);
- resp:
subimagem com a resposta do casamento do template para certa posição do template sobre a imagem de entrada;
- método:
CV_TM_SQDIFF,
CV_TM_CCORR,
CV_TM_CCOEF,
CV_TM_SQDIFF_NORMED,
CV_TM_CCORR_NORMED,
CV_TM_CCOEF_NORMED.

A biblioteca OpenCV



Proposta de tarefa de Rastreio (simplório)

A proposta é exercitar o *Template Matching* por meio da biblioteca OpenCv `cvMatchTemplate()` e da aplicação de outros métodos básicos disponíveis no OpenCV-Python.

`cvLoadImage`, `cvCreateImage`, `cvMatchTemplate`
`cvMinMaxLoc`, `cvRectangle`, `cvNamedWindow`
`cvShowImage`, etc...

Deseja-se a aplicação do melhor método de casamento de template, dentre os acima listados, como ferramenta para uma estratégia funcional (ainda que simplória) de rastreamento de objeto na cena.

Tutoriais do OpenCV:

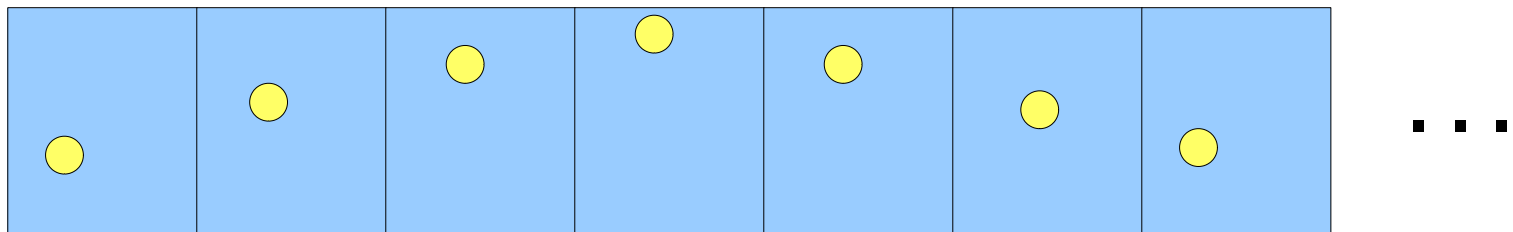
https://docs.opencv.org/4.5.2/d4/dc6/tutorial_py_template_matching.html
docs.opencv.org/3.4/de/da9/tutorial_template_matching.html

A ideia é realizar uma estratégia que aplica o *Template Matching* do OpenCV para simular um rastreamento.

Passos da Tarefa:

A) Faça um vídeo “caseiro” de no mínimo 10 segundos (ou seja **300** quadros) com sua webcam ou celular. Esse vídeo deve exibir o deslocamento lento de um objeto (bem enquadrado na cena) sobre um cenário de fundo constante, com a câmera fixada/estática, sem autofoco (função de foco automático desligada).

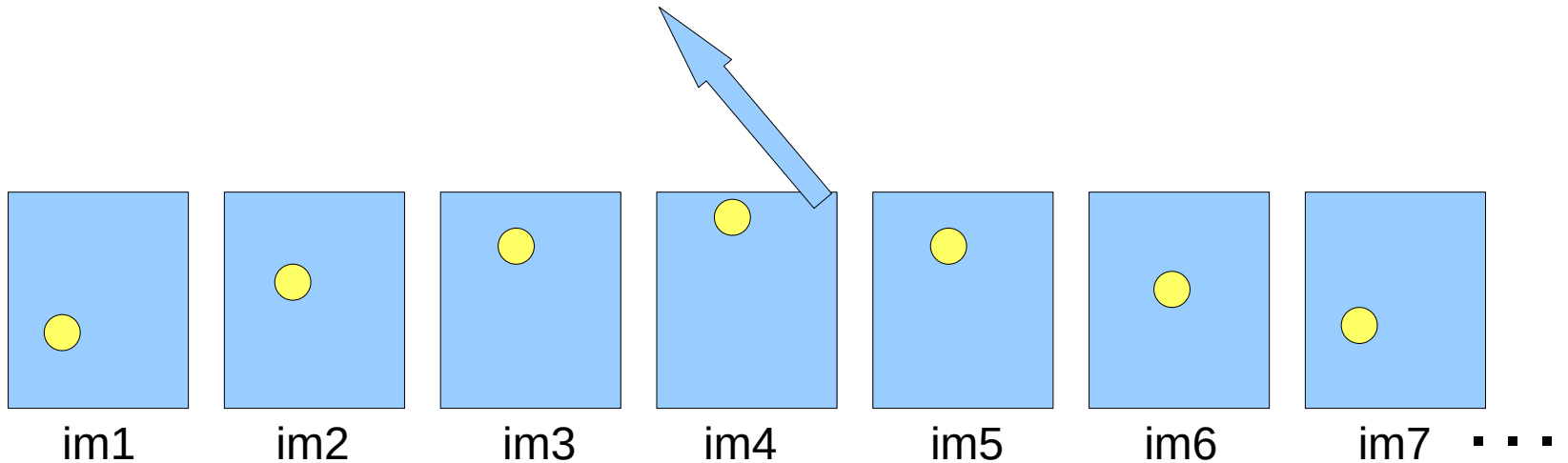
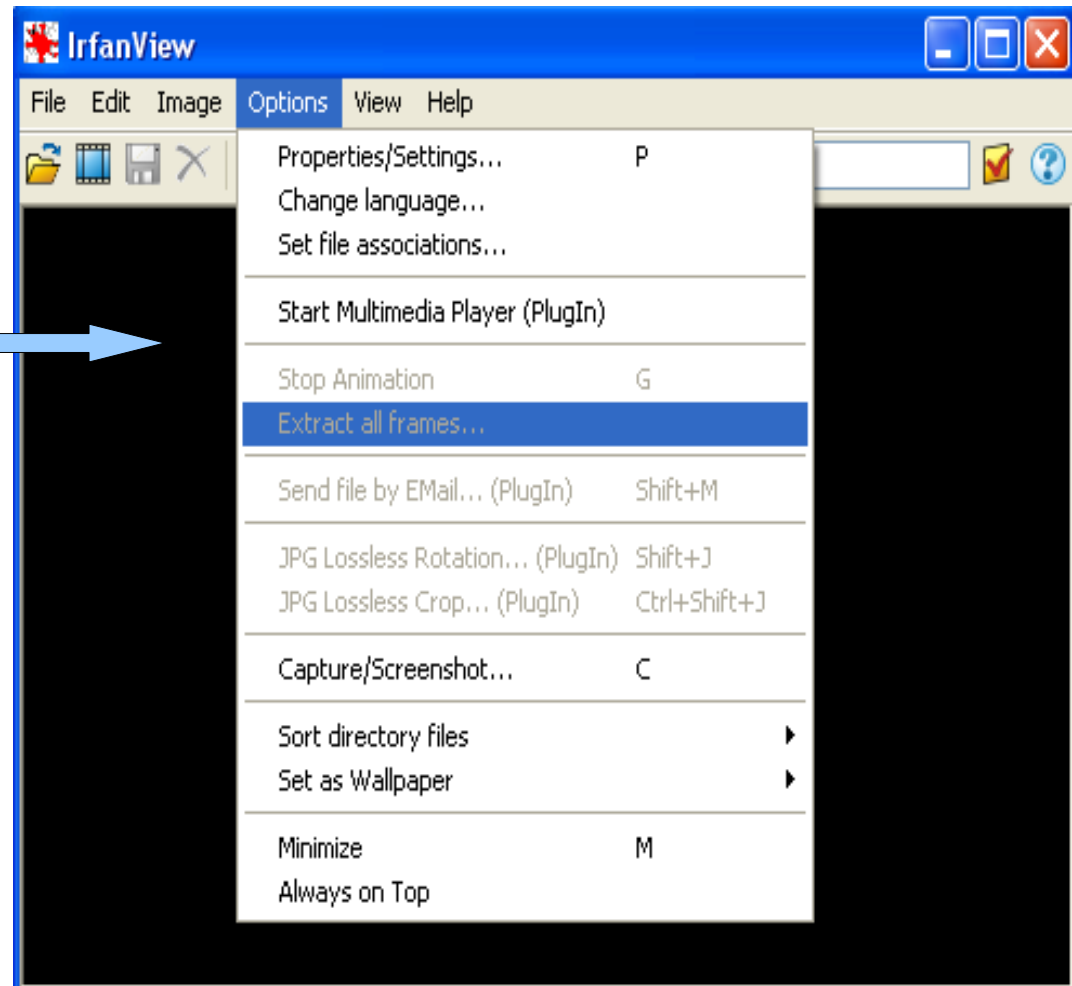
Utilize uma resolução mínima de 640X480 (VGA);



B) Separe os quadros utilizando, por exemplo, o software gratuito *Irfanview*;

C) Salve esses quadros como imagens em tons de cinza.

Esses quadros corresponderão ao conjunto de imagens: im1, im2, im3,...



D) Utilize uma amostra da primeira imagem *im1* para obter um recorte contendo o objeto a ser rastreado, o qual será o *template* a ser rastreado sobre cada um dos quadros restantes (*im2*, *im3*,...).

E) Para cada quadro (*im_i*) aplique *cvMatchTemplate* usando cada um dos métodos:

'cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED',
'cv.TM_CCORR', 'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF' e
'cv.TM_SQDIFF_NORMED'

Cada método responde com um par *max_val* e *min_val*, respectivamente a resposta máxima e mínima para cada quadro (imagem) avaliado.

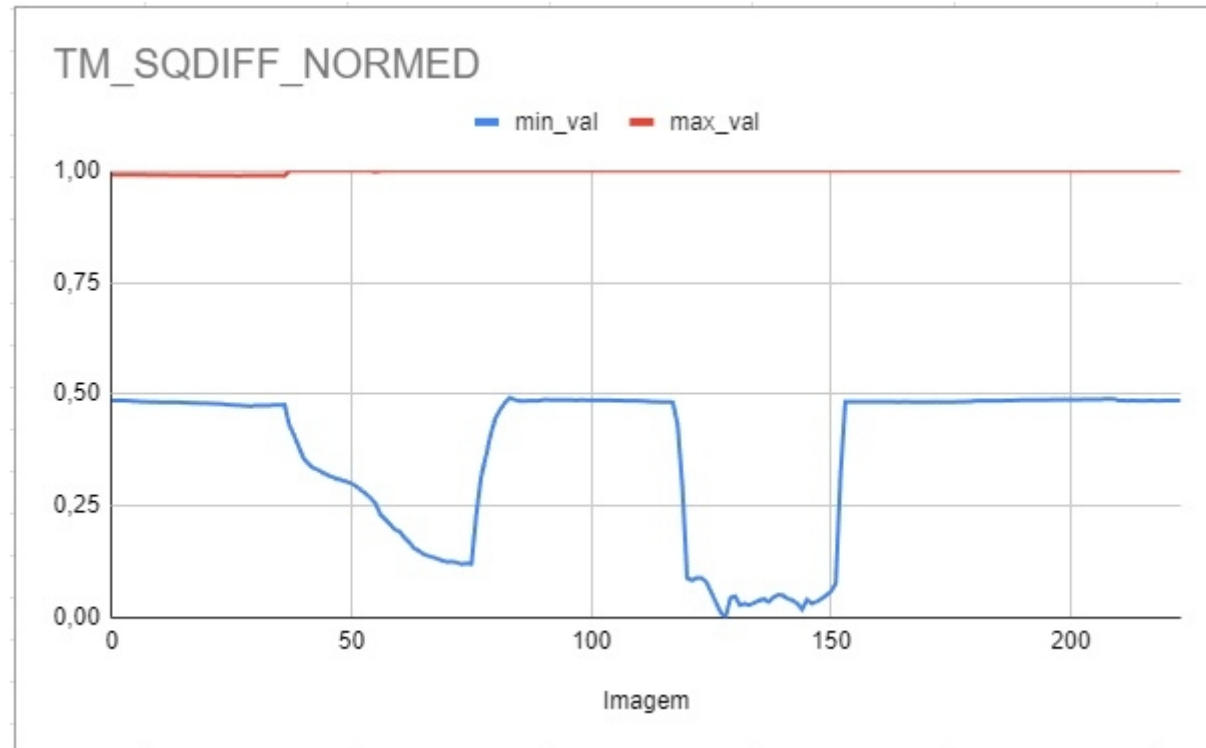
Construa a seguinte tabela para cada método testado. Salve cada tabela em um arquivo csv:

| Método | | |
|-----------------|---------|---------|
| Quadro (imagem) | min_val | max_val |
| im2 | | |
| im3 | | |
| ... | | |
| im300 | | |

F) Para cada tabela do passo E, construa o gráfico:

resposta (min_val e max_val) versus im_i (onde $i=2,3,4,5,6,7\dots$)

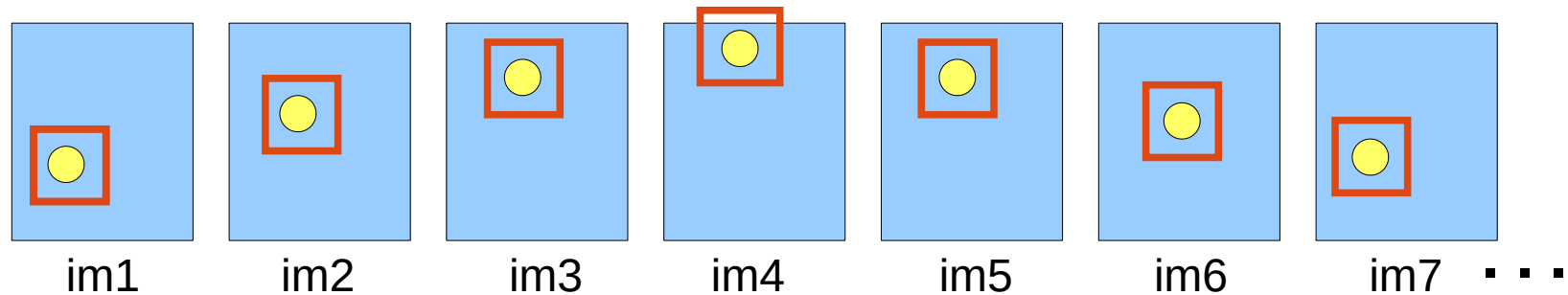
Exemplo:



G) Analise os dados obtidos até aqui e, com base nessa análise, de forma justificada, escolha o melhor método dentre os testados. Talvez mais de um seja ótimo, discuta sobre isso.

H) Tendo selecionado o melhor método, utilize os resultados gerados por ele para compor um vídeo de saída identificando o objeto rastreado a cada quadro.

Isso pode ser feito via *Irfanview*.



I) Execute os mesmos passos para uma sequência de quadros com uma cena mais complexa. Exemplo: o seu mascote se movendo em um cenário de fundo com mais informações visuais.

Entregue um relatório via Moodle contendo:

- Identificador da equipe (dupla);
- A solução da tarefa em todas as suas etapas;
- Identificação da equipe (Máximo 2 membros);
- Scripts (terminação ".py") com comentários explicativos nos trechos importantes;
- **Não** envie links para o Google **Colab**, a correção será realizada no Ubuntu dos laboratórios do CCT.

O relatório precisa ser convincente, contendo algum embasamento teórico, descrevendo a metodologia do experimento, resultados, análise, conclusão e bibliografia. Insira ilustrações e gráficos no relatório, analise e discuta os resultados.

Dados mínimos a serem fornecidos/exibidos:

vídeo de entrada, vídeo de saída (rastreio marcado nos frames), arquivos csv das tabelas, gráficos solicitados, códigos-fontes.

Referências:

Tutoriais sobre OpenCV, Python, bibliografia da disciplina

Sites:

docs.opencv.org/4.5.2/d4/dc6/tutorial_py_template_matching.html

docs.opencv.org/3.4/de/da9/tutorial_template_matching.html