

Universidade do Estado de Santa Catarina

Template Matching

Draylon Vieira
Victor Bernardes

4 de abril de 2021

Sumário

Introdução	1
Objetivo	2
Experimento	3
Resultados	6

Resumo

Neste trabalho será abordado técnicas de template matching usando a biblioteca OpenCV.

Introdução

Este trabalho tem como objetivo a experimentação das diferentes técnicas de Template Matching disponíveis na biblioteca **opencv** bem como a utilização da ferramenta **ffmpeg** para recorte de frames dos vídeos. Com o avanço da disciplina, surgiu a necessidade de classificar imagens. O reconhecimento dos objetos dentro da imagem se faz necessário para categorizar as imagens dentro de suas características, esses dados futuramente podendo ser aplicados nos mais diversos campos de estudo.

Objetivo

O objetivo deste trabalho é utilizar as ferramentas e métodos da biblioteca **opencv** para identificar objetos dentro de imagens.

Experimento

Primeiramente fizemos alguns teste usando todos os métodos de template matching da função *matchTemplate(img,template,method)* disponíveis no **OpenCV**, lá estão disponíveis os seguintes métodos:

1. `TM_SQDIFF`
2. **`TM_SQDIFF_NORMED`**
3. `TM_CCORR`
4. `TM_CCORR_NORMED`
5. `TM_CCOEFF`
6. `TM_CCOEFF_NORMED`

Os métodos foram aplicado a um video de poucos segundos onde era um objeto (drone) em movimento por um fundo estático com a câmera fixa, percebemos que o método `TM_SQDIFF_NORMED` apresentou melhores resultados em relação aos demais.

$$R(x,y) = \frac{\sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (1)$$

$$R(x,y) = \frac{\sum_{x',y'} ((T(x',y') - I(x+x',y+y')) \cdot M(x',y'))^2}{\sqrt{\sum_{x',y'} (T(x',y') \cdot M(x',y'))^2 \cdot \sum_{x',y'} (I(x+x',y+y') \cdot M(x',y'))^2}} \quad (2)$$

Equações usadas pelo método `TM_SQDIFF_NORMED` onde a eq. (2) está com a máscara já aplicada na formula.

Abaixo a Figura 1 representa um frame de um video onde nosso objetivo é detectar a presença do drone. A Figura 2 demonstra o template recortado da Figura 1. A Figura 3 demonstra o resultado final.

Depois de testar qual seria o método a ser utilizado no nosso trabalho, separamos os frames do video utilizando um software chamado **FFmpeg** (FFMPEG.ORG,) onde separamos os frames com o seguinte comando no terminal:

```
$ ffmpeg -i <arquivo_entrada>.mp4 -vf fps=30 <arquivo_saida>%d.png
```

O **FFmpeg** irá transformar o video em um arquivo de imagem, onde serão montadas 30 imagens a cada segundo de video. Depois de geradas as imagens, nos pegamos o diretório e fazemos o método frame a frame para gerar os retângulos e salvamos em uma pasta as imagens resultantes. Por fim, para re-montar o video agora mostrando o objeto identificado iremos usar o seguinte comando:

```
$ ffmpeg -r 30 -i <arquivo_saida>%d.png -c:v libx264 -pix_fmt yuv420p <video_saida>.mp4
```



Figura 1: Primeiro frame do video



Figura 2: Template recortado do primeiro frame



Figura 3: Objeto detectado pelo método

Resultados

Para realizar testes mais complexos, usamos uma câmera em movimento e um objeto (gato) também em movimento, isso somado ao fato da iluminação variar bastante no ambiente deixou o teste do método com ainda mais variáveis, podendo demonstrar o potencial do nosso método escolhido. Observamos que o método depende do ângulo e da iluminação em alguns frames onde ele não consegue identificar o template dentro do frame porém foram poucos frames onde isso aconteceu, dando uma precisão muito boa ao método **TM_SQDIFF_NORMED**. Abaixo seguem imagens do gato passando pelo método. Lembrando que os videos resultantes bem como todas as imagens usadas se encontram na pasta **src** dentro do projeto.



Figura 4: Primeiro frame do video

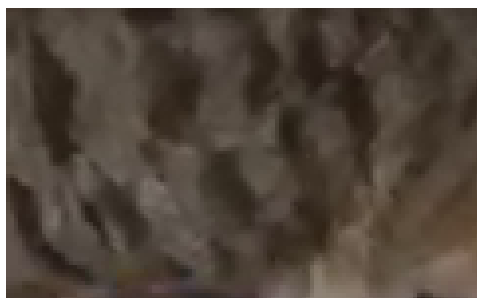


Figura 5: Template usado



Figura 6: Objeto detectado pelo método

Referências Bibliográficas

FFMPEG.ORG. *ffmpeg documentation*. Disponível em: <<https://www.ffmpeg.org/documentation.html>>.

OPENCV, o. *Template Matching*. Disponível em: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.htm>.

SOLOMON, C.; BRECKON, T. *Fundamentos de Processamento Digital de Imagens: Uma abordagem pratica com exemplos em matlab*. Ltc. São Paulo: [s.n.], 2013.