

## COMPARAÇÃO ENTRE TEMPLATE MATCHING E FEATURE MATCHING

FELIPE MACHADO<sup>1</sup>; EDUARDO VARGAS ZUMMACH<sup>2</sup>; JOAB TAVARES  
FAGUNDES<sup>3</sup>; FELIPE DIAS LOPES<sup>4</sup>; MARCELO LEMOS ROSSI<sup>5</sup>

<sup>1</sup>Universidade Federal de Pelotas – [fmachado@ufpel.edu.br](mailto:fmachado@ufpel.edu.br)

<sup>2</sup>Universidade Federal de Pelotas – [eduardo.zummach@hotmail.com](mailto:eduardo.zummach@hotmail.com)

<sup>3</sup>Universidade Federal de Pelotas – [joabtavaresf02@gmail.com](mailto:joabtavaresf02@gmail.com)

<sup>4</sup>Universidade Federal de Pelotas – [felopes\\_7@hotmail.com](mailto:felopes_7@hotmail.com)

<sup>5</sup>Universidade Federal de Pelotas – [marcelo.rossi@ufpel.edu.br](mailto:marcelo.rossi@ufpel.edu.br)

### 1. INTRODUÇÃO

De acordo com o IBGE de 2013, atualizado em 2015, o Brasil tem cerca de 7,28 milhões de pessoas portadoras de deficiência visual, sendo 952 mil com idade entre 10 e 29 anos. Com o aumento da população, estes casos aumentam e torna-se, cada vez mais, imprescindível a inclusão social dos mesmos perante a sociedade.

Dando atenção especial nas gerações mais recentes, por se tratar de uma fase formadora, segundo CARVALHO (2005), “Como processo contínuo, dialético e complexo diz respeito a qualquer aluno que, por direito de cidadania, deve frequentar escolas de boa qualidade, onde aprenda a aprender, a fazer, a ser e onde participem ativamente”. Logo precisamos de ferramentas de ensino que garantam este aprendizado.

Algumas ferramentas já foram elaboradas, a citar exemplo do *Talking Braille* (KUMAR 2007), um tutor eletrônico que ensina braille fornecendo uma resposta audiofônica ao que foi escrito, através de cliques em até 6 pinos na máquina. Sendo assim notado uma melhora na velocidade do aprendizado.

O presente trabalho faz parte de um projeto que busca encontrar formas de conectar a visão computacional com ensino de eletrônica digital, já que se trata de uma área relativamente complexa no tocante ao ensino de esquemas de circuitos lógicos, devido a representação gráfica. O objetivo então é o entendimento da identificação de campos em imagens utilizando duas ferramentas da biblioteca OpenCV: *Template Matching* e *Feature Matching*.

### 2. METODOLOGIA

*Template Matching* é uma técnica de processamento digital de imagem, onde, dado um pedaço de uma imagem (*template*), busca “casar” com a própria imagem. Este casamento ocorre comparando o template em cada parte da imagem original. Como resultado temos uma matriz do tamanho da imagem contendo valores, segundo uma métrica determinada, de melhor ou pior casamento.

Neste trabalho será apresentado o uso da métrica *Cross Correlation Normalized*, pois apresentou bons resultados. A Equação 1 descreve a *Cross Correlation* na qual a matriz resultado (R) é encontrada através do somatório dos valores do template (T) multiplicados pelos da imagem original (I). No nosso caso os valores dos pixels da imagem são de 8 bits, o que permite indicar valores entre 0 a 255, que correspondem ao brilho da imagem, ao fazermos a

conversão de uma imagem colorida para escalas de cinza. Então normalizamos com a raiz do somatório de  $T^2$  multiplicado pelo somatório dos valores de  $I^2$ . Sendo  $x$  (altura) e  $y$  (largura), a localização na imagem.

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

Como resultado, a matriz  $R$  é uma imagem em tons de cinza onde quanto mais branco for um pixel (maior o seu valor), maior é a chance do template corresponder a aquela região na imagem.

Para uma alternativa a fins de comparação, buscou-se outra técnica mais robusta: o ORB (*Oriented FAST and Rotated BRIEF*) utilizando *Feature Matching*.

Composto por FAST (*Features from Accelerated Segment Test*): tem-se uma matriz com centro em  $P$  cercada por 16 pixels, apresentada na Figura 1, estes pixels são, então, divididos em 3 grupos, igual, mais claro ou escuro que  $P$ . Caso mais de 8 pixels façam parte do grupo mais claro ou mais escuro, esta matriz é então selecionada como um ponto de interesse (*Keypoints* ou *Feature*).

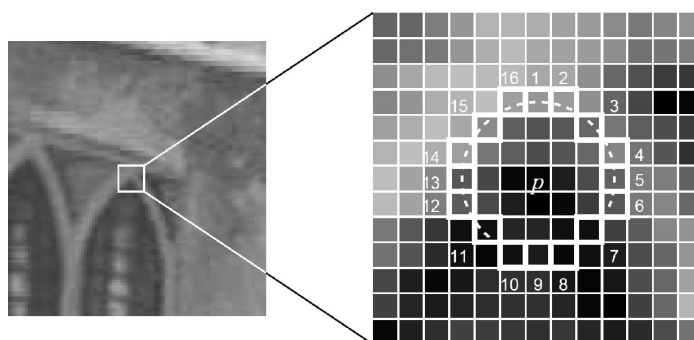


Figura 1: Matriz de 16 pixels utilizada pelo FAST

E por BRIEF (*Binary Robust Independent Elementary Features*): Usando estes pontos de interesse para criar descritores, que são vetores binários de 256 bits. Primeiramente, o algoritmo seleciona um par de pontos aleatórios e equidistantes de  $P$  dentro da vizinhança do *Keypoint*, sendo que se o segundo ponto for mais escuro que o primeiro, ele adiciona 1 ao vetor, fazendo isso 256 vezes.

ORB também aplica o algoritmo FAST em escalas cada vez menores da imagem e consegue, através da orientação, girar os *Keypoints*. Essa orientação se dá a partir do cálculo de uma centróide ( $C$ ), que é equivalente ao “centro de massa” do *Keypoint*, e de uma aresta ( $O$ ), onde através do uso do descritor de cantos Harris (similar ao próprio FAST) encontra tendências de direção a determinado canto. Por fim, um vetor  $CO$  que vai do centróide à aresta fornece a indicação da direção deste *Keypoint*.

### 3. RESULTADOS E DISCUSSÃO

Utilizando como material base os Boletins coronavírus nº 140 e 148 de Pelotas, objetivando, através do experimentalismo no uso de técnicas de

detecção de objetos com OpenCV, encontrar os campos “Confirmados”, “Recuperados”, “Isolados”, “Internados” e “Óbitos”. Fez-se primeiramente tentativas com o Template Matching utilizando os templates apresentados na Figura 2.



Figura 2: Templates retirados do boletim n° 148.

Temos como resultado apresentado na Figura 3, onde a imagem da esquerda é a matriz R do resultado do processo de *Template Matching*. Nota-se os pontos mais claros como os *matches* e sendo mais nítido no template “óbitos” já que possui um fundo preto que se destaca dos outros. Na imagem da direita há a indicação dos casamentos, onde repara-se os primeiros problemas com este método: a escala, novamente mais nítido no template “óbitos”.



Figura 3: Template Matching no boletim n° 140.

Outro problema apresentado, mas não percebido neste exemplo, é a rotacionalidade. Ou seja, se o template não estiver na mesma orientação haverá erro na matriz R já que os pontos mais claros não serão os que procuramos.

Como concorrente, e possível solução a estes dois problemas, há o ORB. Utilizando o mesmo material, obtivemos os resultados apresentados na Figura 4. Nota-se os *Keypoints* (pontas das linhas coloridas) sendo correspondidos na imagem da esquerda, mas na direita há alguns erros devido a própria natureza da ferramenta. No template “internados” há cerca de 1000 Keypoints, e no boletim aproximadamente 40000, consequentemente o algoritmo erra ao confundir as letras finais já que tem descritores muito parecidos nos campos “isolados” e “confirmados”.

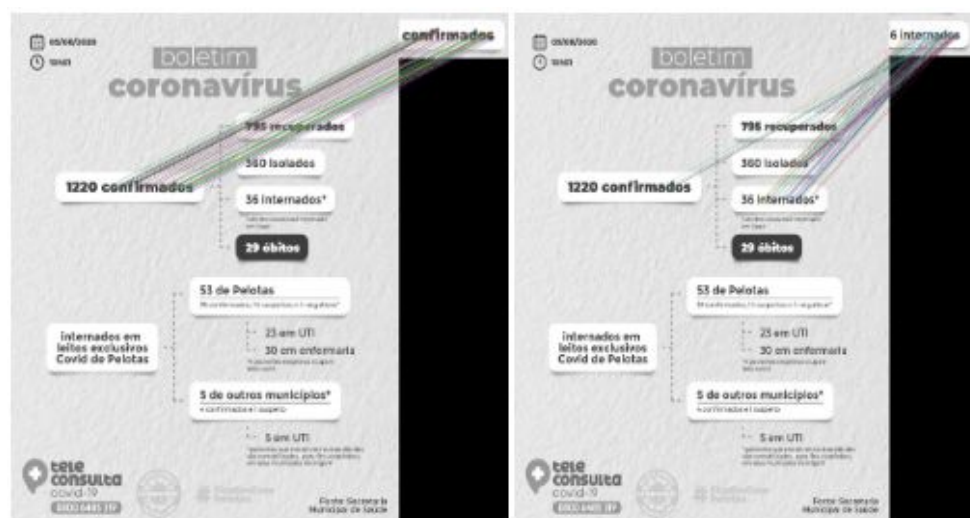


Figura 4: Feature Matching no boletim nº 140.

Porém caso os templates estivessem rotacionados de outra forma, que é o objetivo da pesquisa, o algoritmo encontraria o mesmo resultado. Neste exemplo utilizou-se os 100 descritores mais parecidos.

#### 4. CONCLUSÕES

Conclui-se então que, como início de pesquisa, através do experimentalismo alcançamos bons resultados na detecção de objetos em imagens e espera-se em trabalhos futuros aperfeiçoamentos já em situações mais práticas e complexas como um vídeo por exemplo, onde seja possível até embarcar tal processamento.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

Sidra IBGE, pessoas com deficiência visual. Acessado em 15 Set. 2020. Disponível em: <https://sidra.ibge.gov.br/tabela/5753#resultado>

Carvalho, R.E. Educação inclusiva: do que estamos falando?. **Revista Educação Especial**. n. 26. 2005

Kumar, S. Talking Braille: A new tool to teach blind children. **Research information Ltd**. Burnham, v.34, n.4, p.56-57, 2007

E. Rublee, V. Rabaud, K. Konolige and G. Bradski. ORB: An efficient alternative to SIFT or SURF. **2011 International Conference on Computer Vision**, Barcelona, p.2564-2571, 2011.

Documentação sobre Template Matching. Acessado em Set. 2020. Disponível em: [https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template\\_matching/template\\_matching.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html)

Documentação sobre ORB. Acessado em Set. 2020. Disponível em: [https://docs.opencv.org/3.4/d1/d89/tutorial\\_py\\_orb.html](https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html)