

# Uma implementação do jogo Pedra, Papel e Tesoura utilizando Visão Computacional

Ezequiel França dos Santos, Gabriel Fontenelle Senno Silva

Bacharelado em Ciência da Computação

Centro Universitário SENAC - Campus Santo Amaro (SENAC-SP)

Av. Engenheiro Eusébio Stevaux, 823 – Santo Amaro, São Paulo – CEP 04696-000 – SP – Brasil

ezefranca.br, colecionador.gabriel, (@gmail.com)



## Resumo

Este trabalho apresenta um jogo, controlado por um algoritmo de Visão Computacional de identificação de gestos da mão (hand-tracking). O algoritmo proposto baseia-se na segmentação da imagem e construção de um fecho convexo com algoritmo de Jarvis e determinação do padrão com base em características de sua área.

## 1. Introdução

A busca por meios que tornem os jogos mais interativos tem sido muito explorada. Muitos destes novos meios envolvem a área de visão computacional. Este trabalho apresenta um estudo sobre a viabilidade de utilização de uma webcam como dispositivo de interação baseado em gestos da mão, especificamente para o jogo Pedra, Papel e Tesoura.

## 2. Objetivos

O Reconhecimento de padrões é uma das principais áreas da visão computacional. Neste trabalho, para o reconhecimento da mão e seus gestos utilizamos a combinação de algumas técnicas. Para um reconhecimento satisfatório, em geral, faz nescessário um pré-processamento da imagem. Primeiramente capturamos a imagem da câmera, após isto fazemos a subtração de fundo com intuito de isolar melhor a mão para a captura de seus gestos. A próxima etapa é a normalização da imagem para tons de cinza, seguida da sua binarização (elementos preto e branco). Com a imagem binarizada, começamos o processo de reconhecimento, primeiramente reconhecendo as bordas da mão, e calculando o fecho convexo em torno da mesma. Com estes valores reconhecemos padrões nos valores da área do fecho.

## 3. Metodologia

Neste trabalho utilizamos uma biblioteca que permite o acesso a imagens de câmeras através da plataforma OpenCV e adiciona o tipo de dados camera, composto por uma matriz de tamanho  $A \times B \times 3$ . Os valores representam respectivamente a quantidade de vermelho (R), verde (G) e azul (B) do pixel na posição vertical  $A$  e posição horizontal  $B$ . Cada quantidade, sendo um caractere sem sinal, está entre 0 e 255.

### Segmentação da imagem

#### Normalização para escala de cinza

A primeira fase de pré-processamento trata-se da normalização da imagem colorida para tons de cinza. A normalização foi feita com base no valor médio dos canais de cores da imagem, conforme a equação 1.

$$Valor_{cinza,i,j} = \sum_1^n \left| \frac{R + G + B}{3} \right|_{i,j} \quad (1)$$

Onde:

- $Valor_{cinza}$  - valor entre 0 - 255 para a escala de cinza
- $R$  - valor vermelho do ponto
- $G$  - valor verde do ponto
- $B$  - valor azul do ponto
- $n$  - quantidade de pontos da imagem
- $i, j$  - coordenadas (x,y) do ponto na imagem

A Figura 1 apresenta o resultado deste processo.

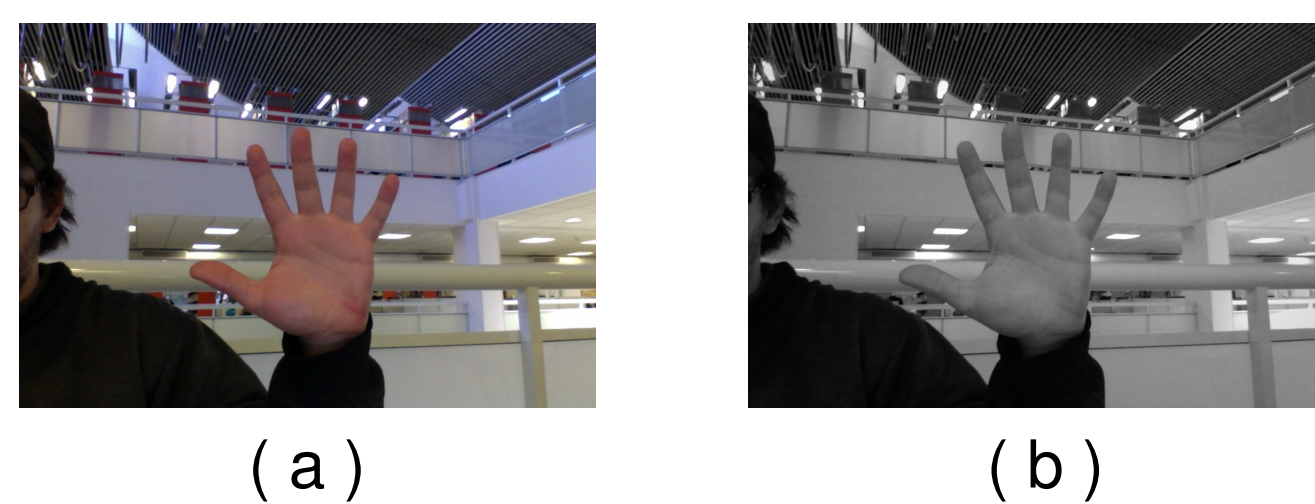


Figura 1: Imagem normal e imagem normalizada em cinza

### 3.0.1 Binarização da imagem

Existem diversos algoritmos para binarização de imagens, dentre a lista de soluções para este o algoritmo de Otsu, por ser de fácil implementação e apresentar resultados satisfatórios nos experimentos realizados.

O método de Otsu é um método de *thresholding* global, isto é, o valor obtido é uma constante, para escolha do melhor limiar. A base deste método é sua interpretação do histograma como uma função de densidade de probabilidade discreta [?], da seguinte maneira:

$$p_r(r_q) = \frac{n_q}{n}, q = 0, 1, 2, \dots, L - 1 \quad (2)$$

Onde:

- $n$  é o total de *pixels* da imagem;
- $n_q$  é o total de *pixels* que tem intensidade  $r_q$  e
- $L$  é o total de níveis de intensidade na imagem.

O resultado da binarização com limiar ajustado segundo o método de Otsu pode ser observado na Figura 2

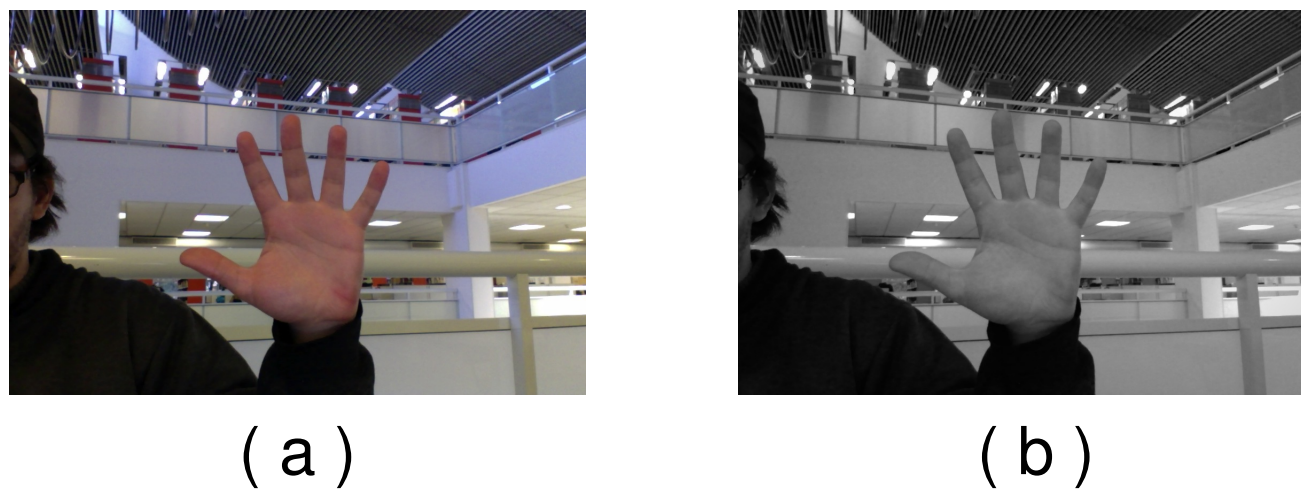


Figura 2: Imagem binarizada com limiar definido pelo método de Otsu.

### Deteção de bordas com filtro Sobel

O filtro Sobel calcula o gradiente da intensidade da imagem em cada ponto, dando a direção da maior variação de claro para escuro e a quantidade de variação nessa direção, através de duas matrizes 3x3, que são convoluídas com a imagem original para calcular aproximações das derivadas - uma para as variações horizontais  $G_x$  e uma para as verticais  $G_y$ .

Máscara de Sobel 3x3

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

A magnitude do gradiente é dado por:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

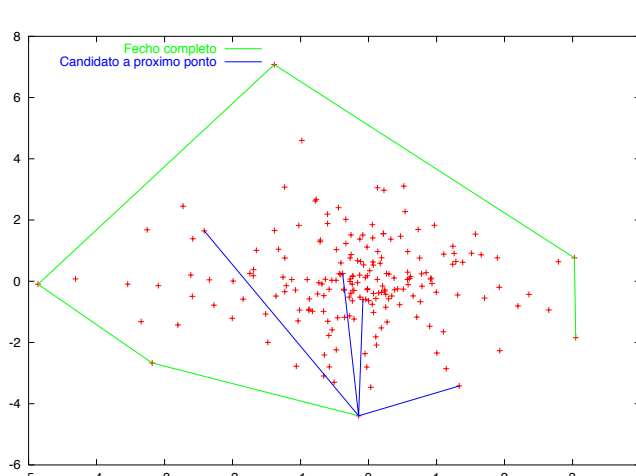
### Reconhecimento dos padrões

#### Determinação do fecho convexo

#### Algoritmo da embrulho de presente

- O *algoritmo da marcha de Jarvis*, popularmente conhecido como *gift wrapping algorithm* / *algoritmo do embrulho de presente*, visita os pontos do fecho convexo de maneira ordenada.

1. Começamos com qualquer ponto do fecho. O ponto com maior coordenada em  $x$  é uma escolha natural. Chamamos esse ponto de  $(X_0, Y_0)$ .
2. Varremos ("marchamos") através de todos os pontos  $(X_i, Y_i)$  e localizamos o ponto tal que o ângulo a partir da coordenada  $(1, 0)$  para  $(X_i - X_0, Y_i - Y_0)$  é mínimo. Este é o próximo ponto de sentido anti-horário a partir de  $(X_0, Y_0)$  no fecho, chamamos-o de  $(X_1, Y_1)$ .
3. Suponhamos que tenhamos localizado o ponto  $(X_i, Y_i)$ ,  $i = 1, \dots, m$  que ocorrem no sentido anti-horário ao fecho onde  $m \geq 2$ . Calculamos todos os ângulos entre os vetores  $(X_i - X_m, Y_i - Y_m)$  e  $(X_{m-1} - X_m, Y_{m-1} - Y_m)$ , e procuramos o ponto  $i$  que tenha o menor ângulo positivo. Adicionamos este ponto ao fecho.
4. Retornamos a etapa 3 até que  $(X_m, Y_m) = (X_0, Y_0)$ .



- O algoritmo de marcha Jarvis tem no pior caso complexidade  $O(n^2)$ , o que ocorre se todos os pontos estão no fecho. Em geral, se  $h$  pontos estão no fecho, a complexidade é  $O(nh)$ .

## 4. Resultados e Discussão

Verificar os principais resultados obtidos de acordo com os objetivos propostos.

Nacken [5] derived an algorithm for computation of pattern spectra for granulometries based on openings by discs of increasing radius for various metrics, using the opening transform. After the opening transform has been computed, it is straightforward to compute the pattern spectrum:

- Set all elements of array  $s$  to zero
- For all  $x \in X$  increment  $s[\Omega_X(x)]$  by one.

To compute the pattern *moment* spectrum, the only thing that needs to be changed is the way  $s[\Omega_X(x)]$  is incremented. As shown in Algorithm 1.

- Set all elements of array  $s$  to zero
- For all  $(x, y) \in X$  increment  $s[\Omega_X(x, y)]$  by  $x^i y^j$ .

**Algorithm 1:** Algorithm for computation of pattern moment spectrum of order  $ij$ .

This algorithm can readily be adapted to other granulometries, simply by computing the appropriate opening transform.

Figura 3: The opening transform using city-block metric: (a) opening transform of Fig. 1(c); (b) pattern spectrum; (c) pattern variance- $x$ ; (d) variance- $y$  spectra.

Figura 4: Pattern mean- $x$  (top) and variance- $x$  (bottom) spectra: the three columns show spectra for Fig. 1(a), (b) and (c) from left to right respectively. Unlike the standard pattern spectra, these spatial pattern spectra can distinguish the three images.

## 5. Conclusão

Sitting on a corner all alone, staring from the bottom of his soul, watching the night come in from the window  
It'll all collapse tonight, the fullmoon is here again In sickness and in health, understanding so demanding It has no name, there's one for every season Makes him insane to know

## Referências

- [1] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26:167–174, 1993.
- [2] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.
- [3] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11:701–715, 1989.
- [4] A. Meijster and M. H. F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(4):484–494, 2002.
- [5] P. F. M. Nacken. *Image Analysis Methods Based on Hierarchies of Graphs and Multi-Scale Mathematical Morphology*. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, 1994.