



# 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,coleccionador.gabriel, (@gmail.com)



## Resumo

Este projeto consiste em 2 jogos controlados por aglortimos de Visão Computacional Diferetnes. Um dos jogos tem como base a identificação das cores primarias e o outro em uma rustica detecção de face atraves de uma detecção de bordas sobre a imagem.

## 1. Introdução

Jogos interativos tem sido uma grande area a ser explorada a medida que a tecnologia e as formas de fazer jogos avançam. A partir disso e buscando uma interação com a camera , chegamos ao nosso trabalho , que tem como foco criar jogos que possibilitem uma interação agradável com a camera , sendo ela um dispositivo essecial para a jogabilidade e a ate para a diversao do usuario , de forma que ela nao atrapalhe ou se torne algo descartavel para o jogador.

## 2. Objetivos

Nesse projeto tivemos 2 objetivos diferentes a serem alcançados nos 2 jogos que criamos. Para o primeiro jogo , o principal objetivo era reconhecer com precisão 4 cores , verde , amarelo , vermelho e azul , assim conseguimos estruturar o classico jogo genius baseado nas imagens reconhecidas pela camera. O segundo jogo ja foi algo pouco mais complexo pois envolveu mais algoritimos como detecções de borda , e alguns filtros , e tem como objetivo uma rustica detecção de face , baseada nas bordas que podem ser reconhecidas na face como boca e os olhos , provendo um reconhecimento eficiente e com alguma precisão.

## 3. Metodologia

Para o inicio do trabalho fomos introduzidos a uma mini biblioteca , baseada em OpemCV que seria a nossa unica ferramenta de acesso a camera e ao tratamento das imagens , sendo ela bem simples a maior parte do trabalho de tratamento e reconhecimento nas imagens foi feito por nos mesmos. A biblioteca nos disponibilizou basicamente funções de acesso a imagem da camera , sendo a que principal delas era a que capturava a imagem da camera e a transformava em uma matriz tridimensional onde as colunas x e y representavam cada pixel da imagem e dentro de cada x e y existiam 3 campos que sao respectivamente os componetes R (vermelho) G (verde) e B(azul) de cada pixel .

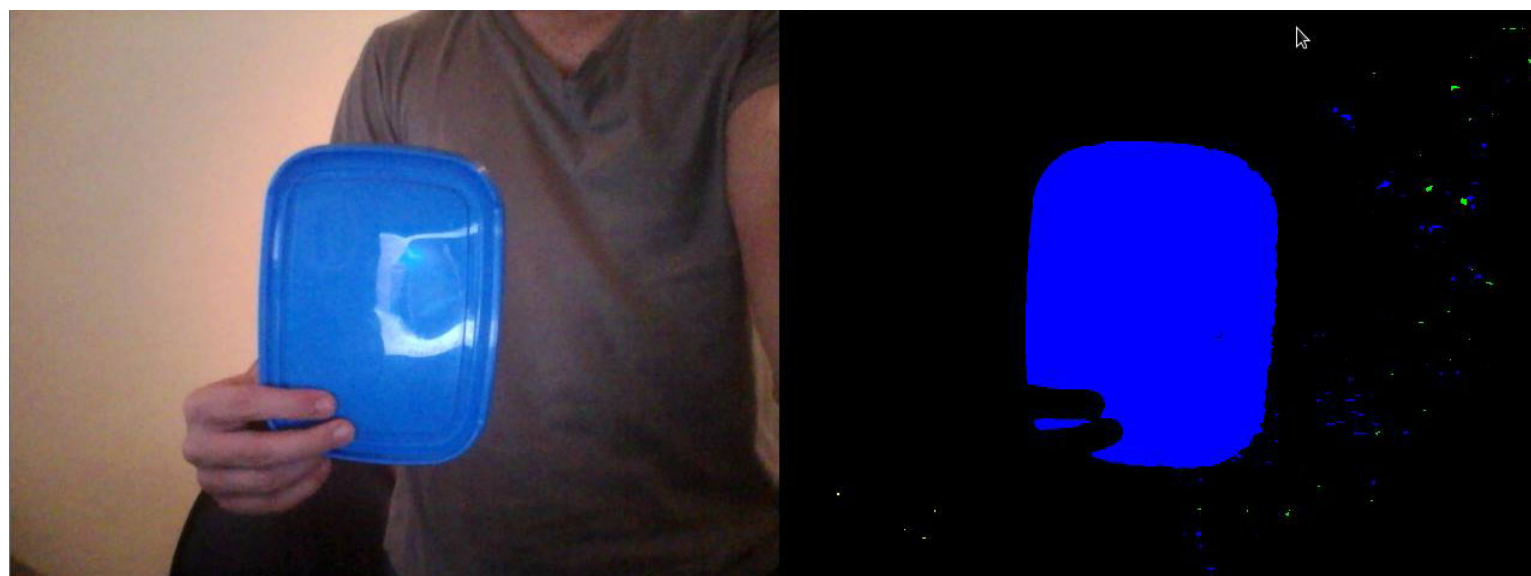
### Jogo Genius

A Principal ideia do reconhecimento das cores desse jogo foi baseada na basica detecção de vermelho , onde verificamos se o componente R do pixel é maior que a soma dos outros dois G e B. Esse metodo funciona bem para o vermelho , porem uma variante dele para as outras cores que precisavamos detectar nao funcionou muito bem ,devido a pequenas mudanças de luz e algumas interferencias , entao tivemos que partir para alguma outra soluçao. Depois de pesquisar , concluímos que mudando o espaço de cor R B G para o H S V, onde o H representa a cor em si , o S a saturação dessa cor e o V o brilho , com esse espaço de cor , diferente do RGB , conseguimos definir claramente um range para cada tipo de cor , podendo filtrar com sucesso as cores que nos desejavamos.

O algoritmo usado para a Conversão

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$
$$S = \frac{MAX - MIN}{MAX}$$
$$V = MAX$$

Um Exemplo de resultado com Azul



### Jogo do Vini

#### 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 como uma função de densidade de probabilidade discreta [?], da seguinte maneira:

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

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 3

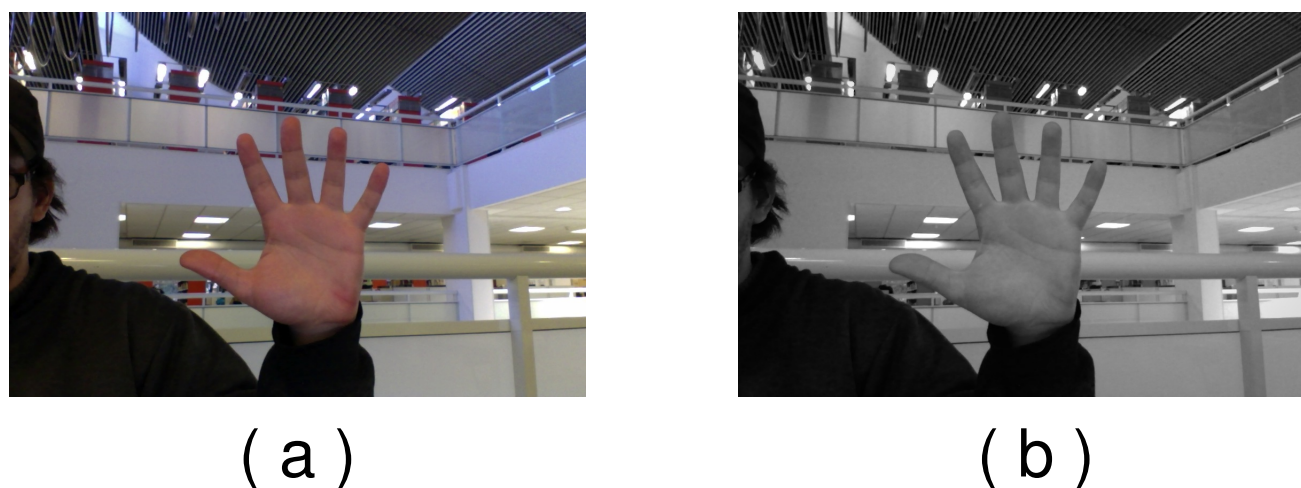


Figura 3: Imagem binarizada com limiar definido pelo método de Otsu.

#### Detecção 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:

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

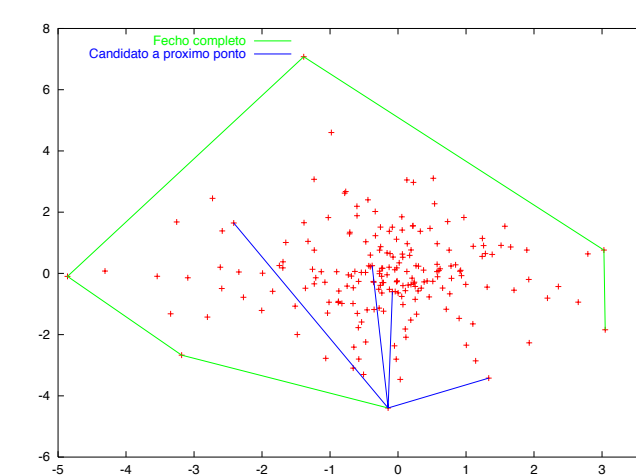
#### Reconhecimento dos padrões

##### Determinação do fecho convexo

##### Algortimo 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. TEste 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. Retornarmos 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 está 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 4: 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 5: 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.