

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

Ezequiel França dos Santos¹, Gabriel Fontenelle¹

¹Centro Universitário Senac - Campus Santo Amaro (SENAC-SP)
Av. Engenheiro Eusébio Stevaux, 823 – São Paulo – CEP 04696-000 – SP – Brasil

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

Abstract. *This paper presents a game, controlled by computer vision, in identification of hand gestures (hand-tracking). The proposed work is based on image segmentation and construction of a convex hull with Jarvis's Algorithm, and determination of the pattern based on the extraction of area characteristics in the convex hull.*

Resumo. *Este trabalho apresenta um jogo, controlado através de visão computacional, na identificação de gestos da mão (hand-tracking). O trabalho proposto baseia-se na segmentação da imagem e construção de um fecho convexo com algoritmo de Jarvis e determinação do padrão com base na extração de 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.

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 necessário um pré-processamento da imagem.

Primeiramente capturamos a imagem da camera, 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. Em seguida, obtemos a determinação do padrão com base na extração de características da área do fecho convexo.

2. Desenvolvimento

2.1. Controle do Jogo

O controle do jogo foi feito totalmente utilizando visão computacional. O diagrama 1, mostra a sequência adotada e cada etapa será brevemente descrita no decorrer deste trabalho.

2.1.1. Reconhecimento de padrões

Em uma primeira etapa deste

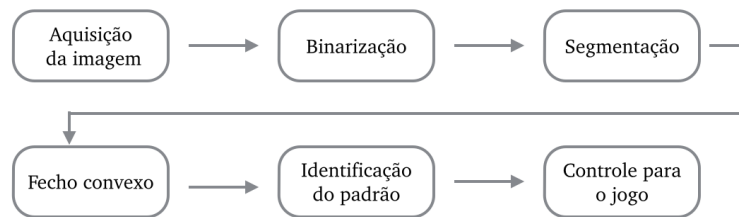


Figura 1. Fluxo no processamento da imagem

2.1.2. Aquisição da imagem

Neste trabalho utilizamos uma biblioteca que permite o acesso a imagens de câmeras através da plataforma OpenCV.

A inclusão desta biblioteca adiciona o tipo de dados camera, composto por uma matriz de tamanho largura x altura x 3.

Os campos largura e altura representam a resolução em pixels da imagem de uma câmera. Esta matriz representa a imagem como uma matriz tridimensional de caracteres sem sinal. São esses três campos que podem ser analisados pelo programa.

A primeira dimensão da matriz representa a altura, a segunda dimensão representa a largura e a terceira representa os canais vermelho, verde e azul (RGB) de cada pixel.

Para ilustrar essa definição, considere um apontador camera *cam.

```
cam->quadro[5][9][0];  
cam->quadro[5][9][1];  
cam->quadro[5][9][2];
```

Os valores representam respectivamente a quantidade de vermelho (R), verde (G) e azul (B) do pixel na posição vertical 5 e posição horizontal 9. Cada quantidade, sendo um caractere sem sinal, está entre 0 e 255.

2.1.3. Segmentação da imagem

2.1.4. 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 2 apresenta o resultado deste processo.

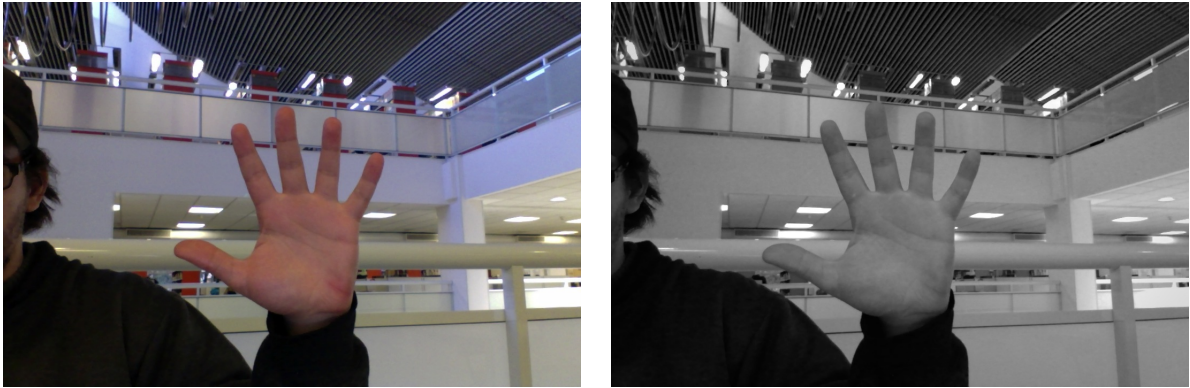


Figura 2. Imagem normal e imagem normalizada em cinza

2.1.5. 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 [2], 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 método de Otsu escolhe o limiar de valor k , tal que k é um nível de intensidade que divide o histograma em duas classes $C_0 = [0, 1, \dots, k - 1]$ e $C_1 = [k, k + 1, \dots, L - 1]$, e que maximise a variância σ_B^2 definida como:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \quad (3)$$

Sendo:

$$\omega_0 = \sum_{q=0}^{k-1} p_q(r_q) \quad (4a)$$

$$\omega_1 = \sum_{q=k}^{L-1} p_q(r_q) \quad (4b)$$

$$\mu_0 = \sum_{q=0}^{k-1} \frac{qp_q(r_q)}{\omega_0} \quad (4c)$$

$$\mu_1 = \sum_{q=k}^{L-1} \frac{qp_q(r_q)}{\omega_1} \quad (4d)$$

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q) \quad (4e)$$

O resultado da binarização com limiar ajustado segundo o método de Otsu pode ser observado na Figura 3

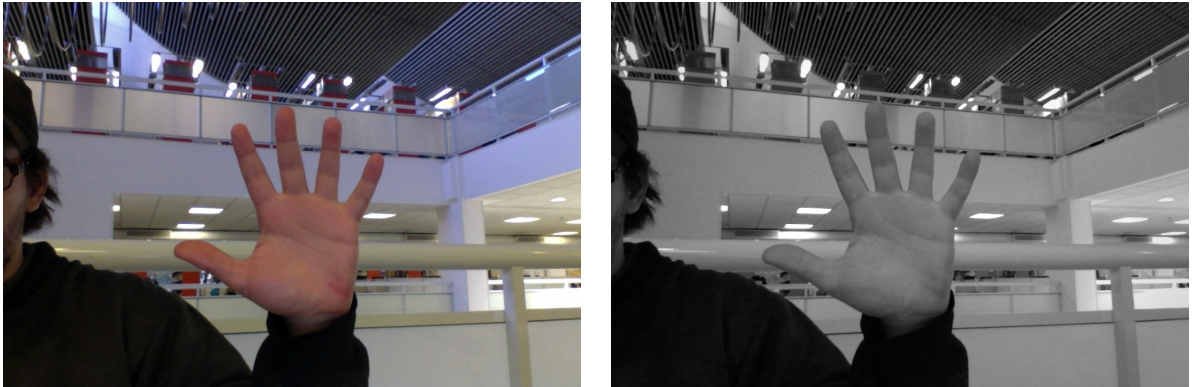


Figura 3. Imagem binarizada com limiar definido pelo método de Otsu.

2.1.6. Detecção de bordas com filtro Sobel

O filtro Sobel calcula o gradiente da intensidade da imagem em cada ponto, dando a direcção da maior variação de claro para escuro e a quantidade de variação nessa direcção, através de duas matrizes 3x3, que são convoluídas com a imagem original para

calcular aproximações das derivadas - uma para as variações horizontais Gx e uma para as verticais Gy .

Máscara de Sobel 3x3

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

A magnitude do gradiente é dado por:

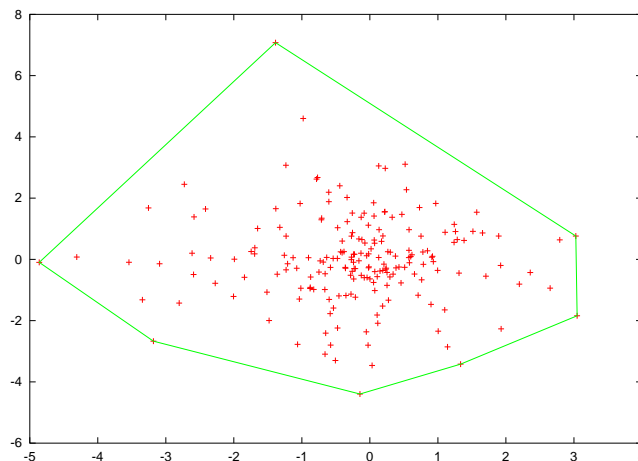
$$|G| = \sqrt{Gx^2 + Gy^2}$$

2.2. Reconhecimento dos padrões

2.2.1. Determinação do fecho convexo

- Um conjunto $\mathcal{S} \subset \mathcal{R}^d$ é *convexo* se $\lambda z_1 + (1 - \lambda)z_2 \in \mathcal{S}$ sempre $z_1, z_2 \in \mathcal{S}$, e $0 \leq \lambda \leq 1$. Resumidamente, \mathcal{S} contém todos os segmentos de linha que conectam pares de pontos em \mathcal{S} .
- O *fecho convexo* gerado por um conjunto de pontos \mathcal{P} é a intersecção de todos conjuntos convexas \mathcal{S} que contem \mathcal{P} . Se $\mathcal{P} = \{z_i \in \mathcal{R}^d, i = 1 \dots, n\}$ é finito, pode ser expresso da seguinte como:

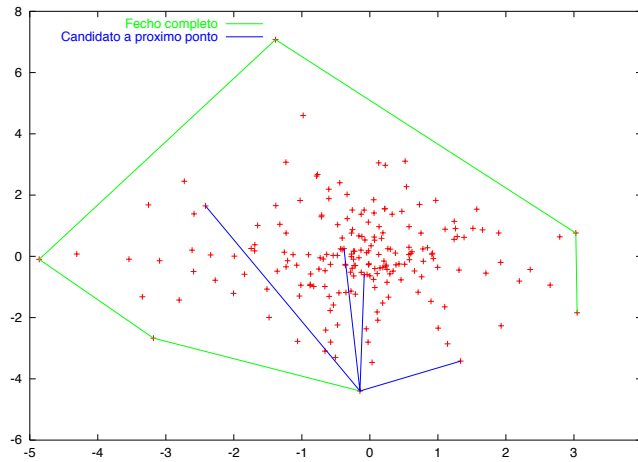
$$\mathcal{S} = \left\{ \sum_{i=1}^n \lambda_i z_i \mid 0 \leq \lambda_i \leq 1, \sum \lambda_i = 1 \right\}.$$



- Se \mathcal{P} é finito, existe um único subconjunto $\mathcal{P}^* \subset \mathcal{P}$ de tamanho mínimo de tal modo que o fecho convexo de \mathcal{P}^* é indentico ao fecho convexo de \mathcal{P} . O conjunto \mathcal{P}^* é chamado de conjunto *gerador de fecho convexo*.
- Encontrar o conjunto de gerador do fecho convexo de um conjunto finito ponto \mathcal{P} é um problema computacional difícil quando a dimensão d é maior que 2. Se a $d = 2$ existem vários algoritmos eficientes.

2.2.2. 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 $O(nh)$.

2.2.3. Calculo da área do fecho convexo

Descrever o que inicialmente estava sendo feito (uma busca em largura), ele ficou lenta, e por isso procuramos outra alternativa, sendo o *shoelace* perfeito para este caso.

$$A_{\text{tri.}} = \frac{1}{2} |x_1 y_2 + x_2 y_3 + x_3 y_1 - x_2 y_1 - x_3 y_2 - x_1 y_3|$$

$$A_{\text{pent.}} = \frac{1}{2} |x_1 y_2 + x_2 y_3 + x_3 y_4 + x_4 y_5 + x_5 y_1 - x_2 y_1 - x_3 y_2 - x_4 y_3 - x_5 y_4 - x_1 y_5|$$

$$A_{\text{quad.}} = \frac{1}{2} |x_1y_2 + x_2y_3 + x_3y_4 + x_4y_1 - x_2y_1 - x_3y_2 - x_4y_3 - x_1y_4|$$

$$A_{\text{quad.}} = \frac{1}{2} |x_1y_2 + x_2y_3 + x_3y_4 + x_4y_1 - x_2y_1 - x_3y_2 - x_4y_3 - x_1y_4|$$

2.2.4. Multilinguagem

Uso do Hash.

2.2.5. Probabilidade Inimigo

Minions.

2.2.6. Pontuação

Sobre a pontuação

3. Resultados

Imagens do jogo do controle (debug), imagens da pontuação e inimigos randomicos com probabilidade.

4. Imagens do Jogo

4.1. Biblioteca Allegro5

nem sei se precisa

4.2. Desenvolvimento de uma Engine utilizando structs

dasdasdasdasdas

Figure and table captions should be centered if less than one line (Figure ??), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 5. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

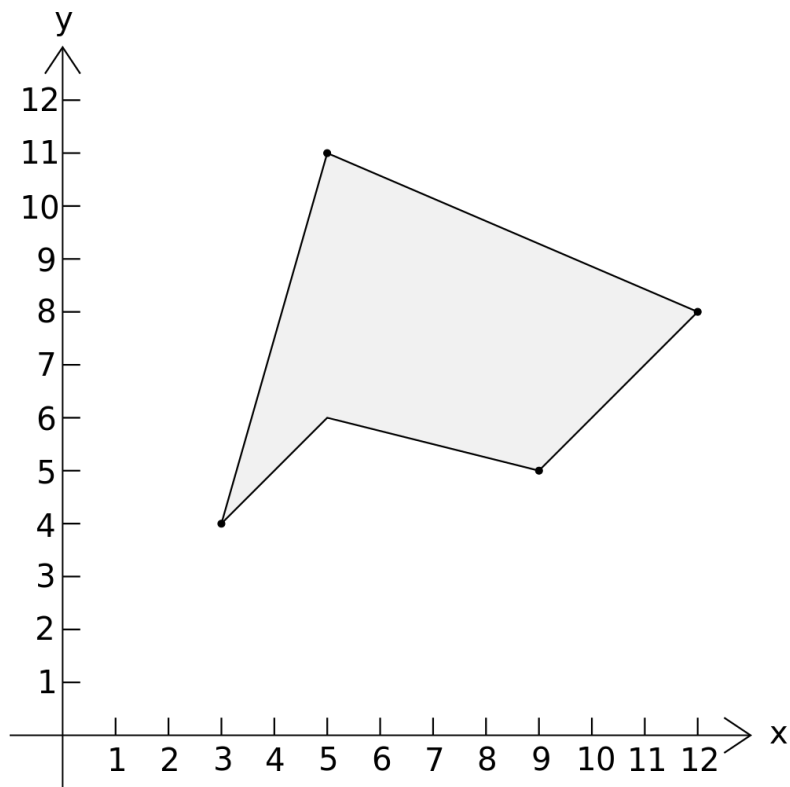
In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

5. Conclusão

gkgkjgkjkhk hhghgjhghghb

6. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references, and [?] [?] e [?]. [?]



$$\begin{aligned}
 A &= \frac{1}{2} | 3 \times 11 + 5 \times 8 + 12 \times 5 + 9 \times 6 + 5 \times 4 \\
 &\quad - 4 \times 5 - 11 \times 12 - 8 \times 9 - 5 \times 5 - 6 \times 3 | \\
 &= \frac{60}{2} = 30
 \end{aligned}$$



Figura 4. A typical figure

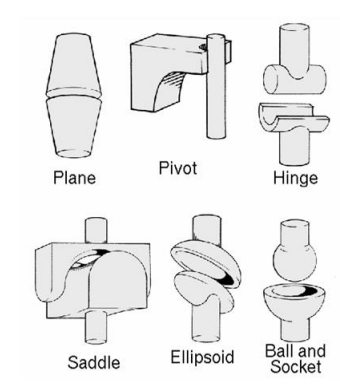


Figura 5. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 4.

Referências

- [1] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26:167–174, 1993. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.
- [2] Matteo Matteucci, Lecture 4 (2000), disponível em [http : //homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf).
- [3] Some useful links are given at .
- [4] A 3x3 isotropic gradient operator for image processing. Never published but presented at a talk at the Stanford Artificial Project, 1968.
- [5] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11:701–715, 1989.
- [6] 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.

Tabela 1. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 \pm 5.22	7.01 \pm 6.84
Selection with in- depth movements	6.29 \pm 4.99	12.22 \pm 11.33
Manipulation with side movements	4.66 \pm 4.94	3.47 \pm 2.20
Manipulation with in- depth movements	5.71 \pm 4.55	5.37 \pm 3.28

- [7] 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.