

# Uma implementação do jogo pedra, papel e tesoura utilizando visão computacional

Ezequiel França dos Santos<sup>1</sup>, Gabriel Fontenelle<sup>1</sup>

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

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

**Abstract.** *Space for translation, when resume is final version*

**Resumo.** *Este trabalho apresenta um jogo, controlado por um algoritmo de visão computacional de identificação 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 sua área e quantidade de vértices.*

## 1. Introdução

A busca por meios que tornem os jogos mais interativos tem sido muito explorada na indústria atual de games. Entre muitos meios e dispositivos a visão computacional tem se destacado por pode ser utilizada para capturar gestos e criar interfaces de interação e controle. Os dispositivos comerciais atuais de interação baseados em gestos utilizam equipamentos caros e muitas vezes requerem ambientes especiais que dificultam a difusão destas interfaces de controle. Este trabalho apresenta um estudo sobre a viabilidade de utilizarmos uma webcam como dispositivo de interação baseado em gestos da mão, especificamente para o jogo pedra, papel e tesoura.

## 2. título provisório Elaboração técnica

Para reconhecer os gestos da mão foi necessário submeter a imagem da camera a algumas fases de pré-processamento. O diagrama 1, mostra a sequência adotada e cada etapa será brevemente descrita no decorrer deste trabalho.

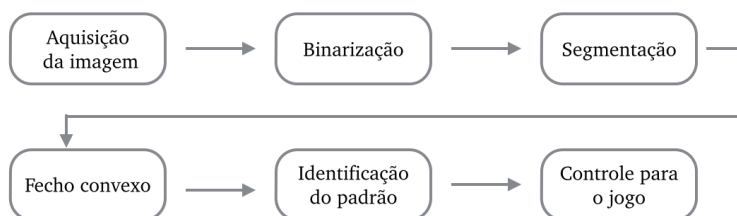


Figura 1. Fluxo no processamento da imagem

### 2.1. Aquisição da imagem

Falar da aquisição da imagem, opencv, etc.

## 2.2. Segmentação da imagem

### 2.2.1. 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.

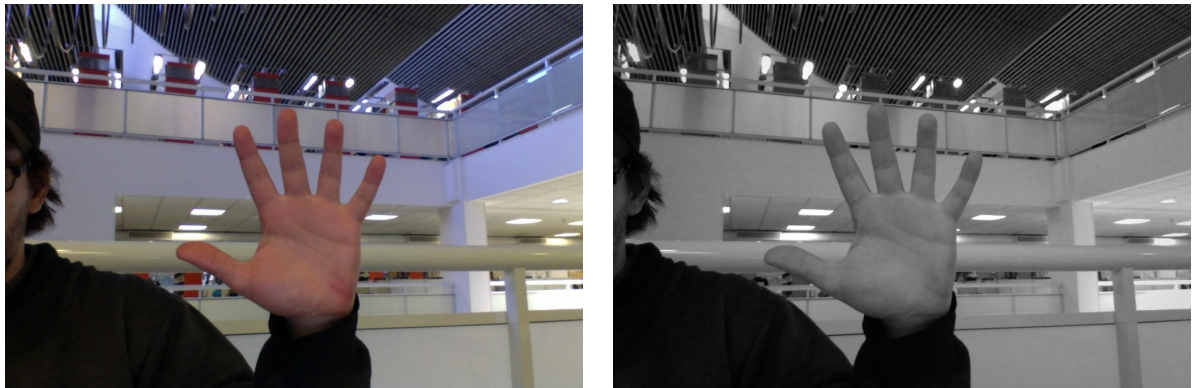


Figura 2. Imagem normal e imagem normalizada em cinza

### 2.2.2. 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 algoritmo de Otsu encontra um nível de cinza  $t$  tal que a soma ponderada da variância dentro das classes. Uma desvantagem de utilizar este algoritmo é a influência da média de todo os pixels da imagem. Isto pode fazer com que o limiar ótimo para a página toda não seja o mesmo que o de dentro de uma janela.

## 2.3. Implementação

### 2.3.1. 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:

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

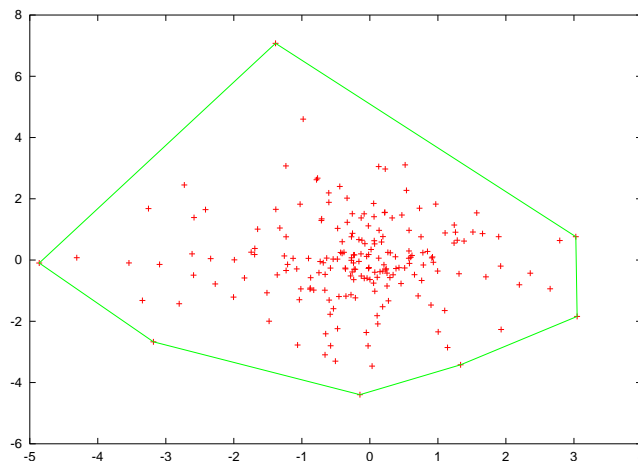
## 2.4. Reconhecimento de padrões

Falar brevemente, e apresentar a ideia utilizada

### 2.4.1. 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 convexos  $\mathcal{S}$  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:

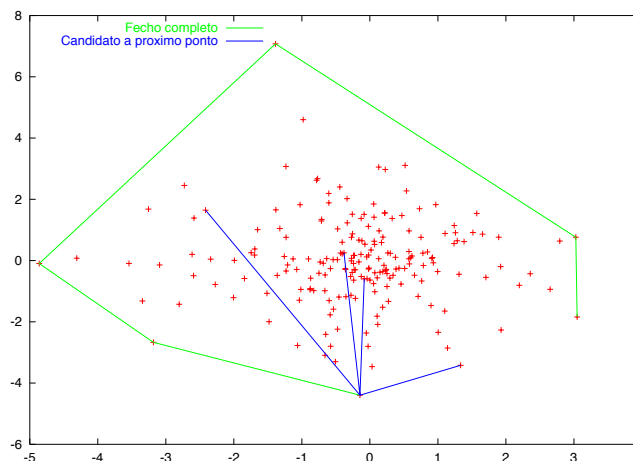
$$\mathcal{S} = \left\{ \sum_{i=1}^n \lambda_i z_i \mid 0 \leq \lambda_i \leq 1, \sum \lambda_i = 1 \right\}.$$



- 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.

### 2.4.2. 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  $O(nh)$ .

### 2.4.3. 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.

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

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

## 2.5. Biblioteca Allegro5

nem sei se precisa

## 2.6. Desenvolvimento de uma Engine utilizando structs

dasdasdasdasdasdas

## 3. Imagens do Jogo

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 4. 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.

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

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

## 4. Conclusão

gkgkjgkjkhkjhk hhghgjhghghb

## 5. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references, and [Backes e Schmid 2011] [Barbosa 1997] e [Maria 2011]. [Kishimoto 2008]

SÓ LEMBRE DE ADICIONAR A BIBLIOGRAFIA CERTINHA NO ARQUIVO

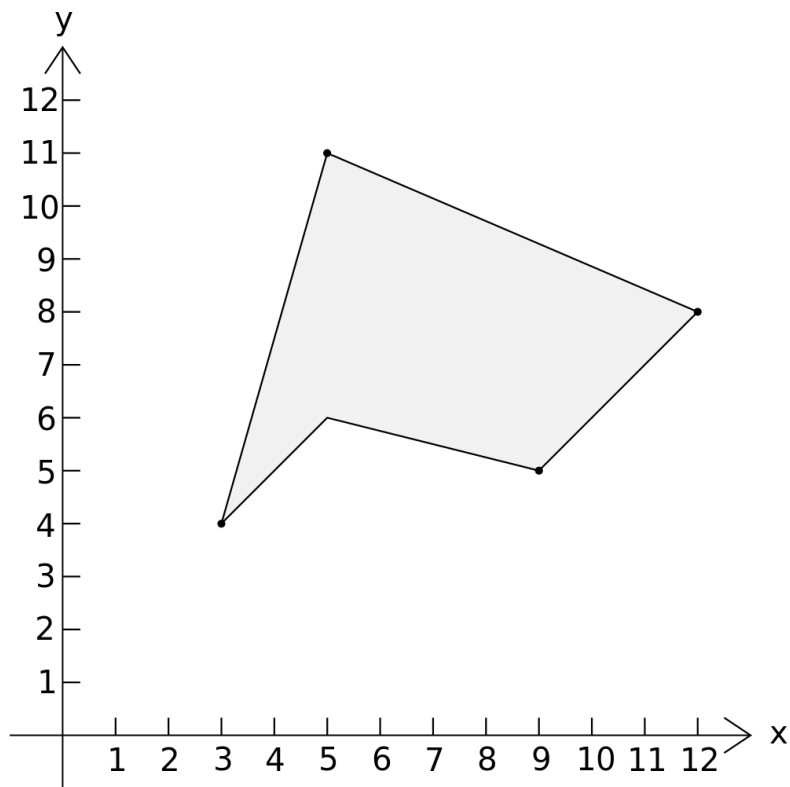
\*.bib

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

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

## Referências

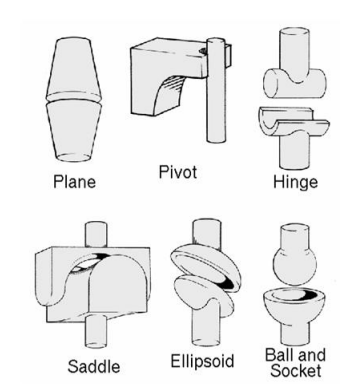
- Backes, I. M. e Schmid, A. B. (2011). Álgebra booleana no processo ensino aprendizagem da matemática. Em *Fundação Universitária do Desenvolvimento do Oeste [online]*. UNOCHAPECÓ.
- Barbosa, M. C. S. (1997). Jogo, brinquedo, brincadeira e a educação. Em *Educ. Soc. [online]*, pages 398–404. Scielo.
- Kishimoto, T. (2008). *O jogo e a educação infantil*. Cengage Learning, São Paulo, SP.
- Maria, R. (2011). A ludicidade na educação: uma atitude pedagógica. Em *Curitiba- PR, 2º edição*. IBPEX.



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



**Figura 3. A typical figure**



**Figura 4. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 3.**