

# Controle de Orientação e Velocidade de Cadeira de Rodas por Visão Computacional

Wanderson S. da Silva<sup>1</sup>, Paulo N. S. do Carmo<sup>2</sup>,  
José Roberto de A. Aranha Junior<sup>2</sup>, José Lucas C. Magalhães<sup>1</sup>,  
Thalles Jhonatan dos A. Silva<sup>2</sup>, Prof. Dr. Wener B. de Sampaio<sup>2</sup>,  
Prof. Dr. Paulo Rogério de A. Ribeiro<sup>2</sup>, Prof. Dr. Shigeaki L. de Lima<sup>2</sup>,  
Prof. Dr. Denisson Q. Oliveira<sup>2</sup>, Prof. Dr. Anselmo C. Paiva<sup>3</sup>

<sup>1</sup>Bacharelado Interdisciplinar em Ciência e Tecnologia

<sup>2</sup>Engenharia da Computação

<sup>3</sup>Núcleo de Computação Aplicada

Universidade Federal do Maranhão (UFMA) – São Luís, MA – Brazil

wankapef.silva@gmail.com, paulonayron.s@hotmail.com,  
{robertoaranhajr, lucasmagalhaesjc, thalles4njos}@gmail.com,  
wener.sampaio@ufma.br, paulo.ribeiro@ecp.ufma.br,  
{shigeaki.lima, dq.oliveira}@ufma.br, paiva@deinf.ufma.br

**Abstract.** *This work proposes a computer vision system for wheelchair control based on facial coordinates and head position estimation. This work comprises the design of a low-cost motorized wheelchair, which aims to promote independence and quality of life for people who have motor difficulties to manipulate a joystick. The tests performed on the built prototype resulted in an accuracy of 87.78%, an precision of 90.00% and a sensitivity of 87.70%. Tests performed in the motorized wheelchair resulted in an average accuracy of 94.69%, average precision of 91.04% and average sensitivity of 89.38%.*

**Resumo.** *Este trabalho propõe um sistema de visão computacional para controle de cadeira de rodas com base em coordenadas faciais e estimativa da posição da cabeça. Este trabalho compõe o projeto de uma cadeira de rodas motorizada de baixo custo, que tem como objetivo promover independência e qualidade de vida a pessoas que possuem dificuldades motoras para manipular um joystick. Os testes realizados no protótipo construído resultaram em uma acurácia de 87,78%, a precisão de 90,00% e uma sensibilidade de 87,70%. Os testes realizados na cadeira de rodas motorizada resultaram em uma acurácia média de 94,69%, precisão média de 91,04% e sensibilidade média de 89,38%.*

## 1. Introdução

Diversos são os problemas que podem acometer o sistema locomotor humano, sendo a tetraplegia um dos mais debilitantes. Esta se refere à paralisia parcial ou completa do tronco e músculos respiratórios e dos quatro membros, sendo resultado de lesões da medula cervical [Cerezetti et al. 2012].

As pessoas que possuem lesão medular acima do nível medular T1, onde é classificada a tetraplegia, têm a mobilidade funcional comprometida, por conta da diminuição ou perda da função motora e/ou sensória e/ou anatômica [Medola et al. 2009].

A mobilidade funcional (MF), por sua vez, é definida como a habilidade do indivíduo de se mover independentemente no ambiente, atingindo o destino desejado [Kleiner 2009].

Este trabalho propõe um sistema de Visão Computacional (VC), aliado a técnicas de Processamento Digital de Imagens (PDI), que compõem o projeto de uma cadeira de rodas motorizada, cujo objetivo é reestabelecer a MF, a fim de melhorar a qualidade de vida e promover maior independência às pessoas que possuem dificuldades de locomoção.

## 2. Trabalhos relacionados

Trabalhos vêm sendo desenvolvidos com a finalidade de promover melhor qualidade de vida e independência para o público alvo desse artigo.

Pinto (2017) desenvolveu um sistema de controle de cadeira de rodas de baixo custo, onde foram implementados dois modelos de interface de controle destinados a pessoas com tetraplegia. O primeiro utilizou sensores de pressão para acionar os motores por meio de sopro e sucção. O segundo modelo utilizou um joystick adaptado ao queixo para a execução dos comandos.

Rabhi *et al.* (2018) propôs um sistema de controle para cadeira de rodas inteligente de baixo custo, que usa uma câmera fixada na cadeira de rodas, capturando os movimentos das mãos do usuário em tempo real. Através de um algoritmo de VC e técnicas de inteligência artificial os movimentos são reconhecidos. Uma das principais características deste trabalho é a velocidade variável semelhante à de um joystick padrão.

As propostas acima descritas apresentam soluções que possuem problemas com relação ao conforto e a usabilidade, além de exigir do usuário a movimentação da mão. A proposta aqui apresentada objetiva superar tais problemas.

## 3. Metodologia

A arquitetura do sistema aqui proposto segue as seguintes etapas apresentadas na Figura 1. A metodologia inicia-se na aquisição, seguida do processamento do vídeo para a detecção da face do usuário em tempo real e segmentação da mesma, que corresponde à ROI (*Region of Interest*).

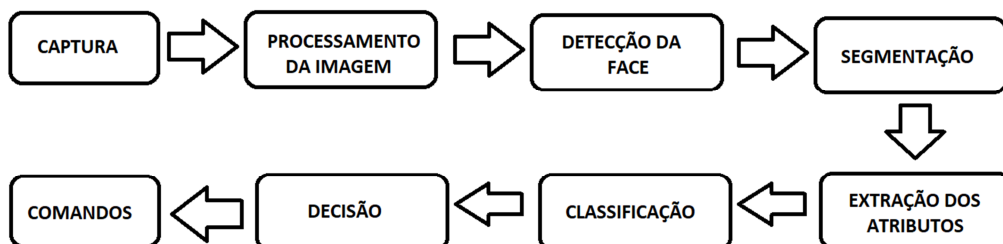


Figura 1. Arquitetura do sistema de VC.

Com a região de interesse segmentada, são encontrados pontos que servirão de atributos para as etapas de classificação, de modo a possibilitar a tomada de decisão do sistema. Nas subseções seguintes serão apresentados os métodos e ferramentas utilizadas para cada etapa.

### 3.1. Biblioteca OpenCV

Utilizou-se a biblioteca OpenCV, referência nas áreas de VC e PDI, desenvolvida inicialmente pela Intel e atualmente de código aberto sob licença BSD [Kaehler and Bradski 2016].

O OpenCV é primordial para a aquisição e processamento da imagem, além de prover técnicas para obtenção da posição e orientação da cabeça do usuário, respectivamente descritas nas Seções 3.3 e 3.5.

### 3.2. Biblioteca Dlib

A Dlib contém algoritmos de aprendizado de máquinas e ferramentas de criação de softwares complexos para solucionar problemas do mundo real. É usada na indústria, academia, além de variados domínios incluindo a robótica, dispositivos embarcados, celulares e computadores de alta performance [King 2009].

Essa ferramenta é fundamental na predição da forma e posterior detecção da face, descritas na Seção 3.4

#### 3.2.1. Descritores HOG – *Histogram of Oriented Gradients*

Os descritores HOG extraem características da imagem baseando-se na ideia de que objetos podem ser caracterizados de maneira satisfatória por meio da distribuição dos gradientes de intensidades dos *pixels* ou pelas direções das bordas, mesmo sem conhecer precisamente as posições de inclinação ou bordas correspondentes [Dalal and Triggs 2005].

O processo para gerar os descritores pode ser dividido em quatro etapas: cálculo do gradiente em cada *pixel*, agrupamento dos *pixels* em células, agrupamento das células em blocos e obtenção do descritor [Cruz et al. 2012].

#### 3.2.2. Máquina de Vetores de Suporte – *Support Vector Machine*

A Máquina de Vetores de Suporte (MVS) é uma técnica de aprendizado de máquina (AM), que se baseia nos princípios da minimização do risco estrutural, proveniente da teoria do aprendizado estatístico. A MVS tem como objetivo construir hiperplanos ótimos que apresentem a maior margem de separação entre classes. O hiperplano gerado é determinado por um subconjunto dos pontos das classes, chamado vetores de suporte [Soares 2008].

### 3.3. Aquisição e Processamento de Imagem

Para a tarefa de captura da imagem utilizou-se o OpenCV. Os frames capturados por uma webcam foram redimensionados para 360p, e em seguida convertidos para a escala de cinza, de modo a reduzir o esforço computacional para as demais etapas de processamento.

### 3.4. Detecção de Face

Com base nos frames de entrada, ou seja, nas imagens em cinza obtidas na etapa anterior, utilizou-se um preditor de formas, cujo papel é localizar marcos faciais, que são pontos-chave de interesse ao longo da face, ilustrados na Figura 2.

A primeira etapa do processo foi a delimitação da face, isto é, a obtenção das coordenadas (x, y) do rosto na imagem. Utilizou-se para isso o algoritmo de detecção facial padrão da biblioteca Dlib. Esse algoritmo utiliza um detector de objetos, se valendo dos descritores HOG combinados ao classificador MVS Linear, descrito por Osuna et al. (1997), pré-treinado para rostos.

Com a face delimitada, detectou-se estruturas importantes na região. O objetivo do preditor é localizar zonas específicas da face, que são: boca, sobrancelha esquerda, sobrancelha direita, olhos, nariz e mandíbula. O preditor faz esse processo através de 68 pontos, marcados em coordenadas específicas, de regiões que envolvem cada estrutura facial. Com base nisso é feita a análise da distância provável entre os frames de entrada e o preditor, e assim um conjunto de árvores de regressão é responsável por estimar as posições de referência da face.

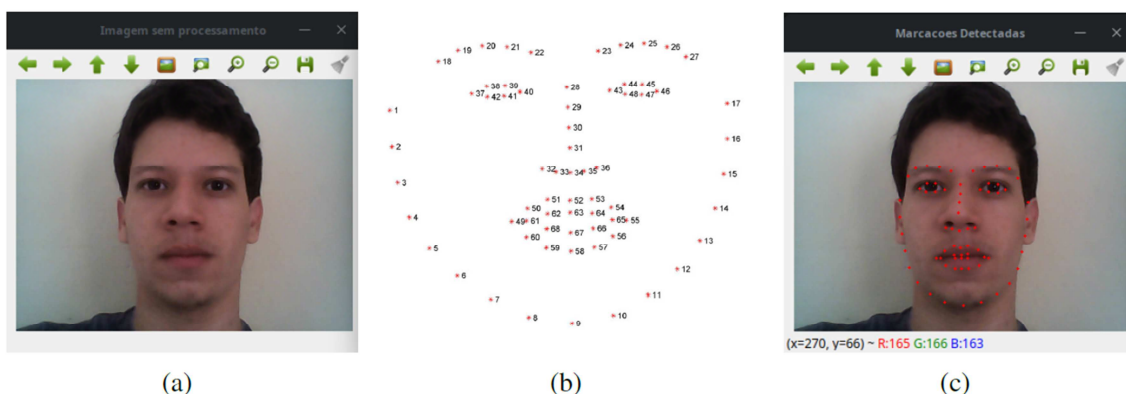


Figura 2. a) Imagem sem processamento. b) Preditor. c) Marcações detectadas.

### 3.5. Posição estimada da cabeça

Para obter a posição e orientação da cabeça do usuário, utilizou-se a técnica de estimação de posição de objetos POSIT, proposta por DeMenthon e Davis (1995). O algoritmo relaciona os pontos 2D (Seção 3.4) e um modelo geométrico 3D aproximado da face humana. Após este relacionamento, projeta-se um vetor que representa a orientação da face, onde o mesmo é utilizado para o processo de classificação e decisão de comandos.

### 3.6. Hardware do Sistema de Controle

O sistema de controle desenvolvido fez uso de uma Raspberry Pi (RPi), microcomputador que no modelo B, em sua terceira geração, conta com um processador Broadcom 64-bits ARM Cortex-A53 Quad-Core, com um clock de 1.2Ghz. Possui conectividade *wireless*, *bluetooth* e *ethernet*, além de contar com portas HDMI, GPIO e USB para conexões de diversos tipos de periféricos. O sistema operacional da mesma é uma versão específica do Debian (distro Linux) para processadores ARM, e encontra-se

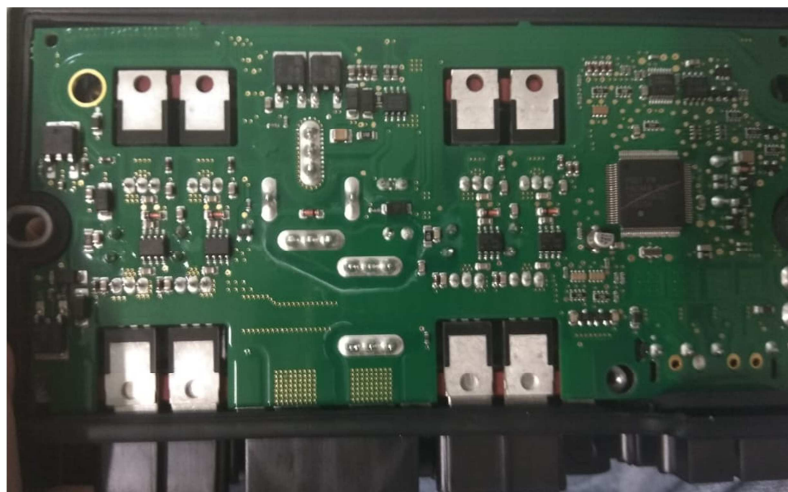
instalado em um cartão microSD. A RPi, portanto, se mostrou adequada à aquisição e ao processamento de imagens aqui abordados.

Para a captura de imagens foi acoplada à RPi uma Pi NoIR Camera V2, que possui um sensor Sony IMX219 com 8 *megapixels*. Seu diferencial se dá por conta da ausência do filtro infravermelho (NoIR), que a torna capaz de capturar imagens mesmo em ambientes com pouca ou nenhuma luminosidade.

Por meio da porta USB da RPi foi conectado um Arduino UNO – plataforma de prototipagem amplamente utilizada – que conta com um microcontrolador ATmega328P, além de possuir vários pinos de entrada e saída digitais programáveis. Para este microcontrolador foi atribuída a função de classificar e interpretar em comandos os dados obtidos pela RPi.

Deixar as operações algébricas, tais como somas entre pontos e multiplicações de matrizes para o Arduino, se mostrou uma tarefa necessária para reduzir o esforço da RPi, uma vez que o processamento em tempo real das imagens adquiridas exige um grande custo computacional. Além disso, no Arduino as operações são realizadas diretamente em *hardware*, o que por si só já torna o processo muito mais rápido e eficiente.

A cadeira de rodas motorizada Ottobock B400, por sua vez, possui nativamente o sistema de controle VR2, visto na Figura 3, que consiste em um módulo de potência, onde está incorporado todo o controle para acionamento dos dois motores DC de 24V, alimentados por duas baterias de 12V ligadas em série. A cadeira conta ainda com um *joystick* responsável pela interação com o usuário para o controle de direção e velocidade. Por meio da adaptação proposta foi emulado no Arduino o comportamento do joystick original do sistema.



**Figura 3. Módulo de potência VR2.**



### 3.7. Classificação e Decisão de Comandos

#### 3.7.1. Primeira Abordagem

Inicialmente, projetou-se o vetor de orientação obtido na seção 3.5, de modo que fosse possível classificá-lo através apenas de suas componentes bidimensionais.

Com este novo vetor 2D foi possível obter um retângulo cujo centro coincidissem com a origem do vetor de orientação 2D. Horizontalmente, o retângulo é equivalente ao tamanho da face detectada, e verticalmente correspondente à cerca de 40% da região delimitada, de modo a deixá-lo invariante à escala, ou seja, sem importar a distância entre o rosto e a câmera. Essa redução vertical mostrou-se necessária, uma vez que se demanda maior liberdade de movimento, por conta das limitações do usuário.

De acordo com a posição deste retângulo menor, alinhadas lateralmente a ele, traçou-se duas retas verticais de modo a segmentar cinco regiões, denominadas comandos principais, correspondentes às seguintes direções: parado, esquerda, direita, trás e frente, conforme ilustrado na Figura 4.

Além disso, foram construídas quatro retas horizontais, também ilustradas, de modo a subdividir as regiões esquerda e direita em quatro outras sub-regiões, referentes aos comandos combinados, descritos como: esquerda/trás, direita/trás, esquerda/frente e direita/frente.

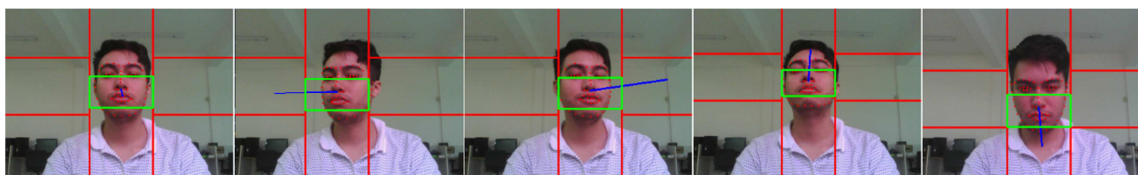


Figura 4. Regiões de Comandos.

Com base na localização do vetor nas regiões anteriormente descritas, um caractere correspondente à cada direção foi transmitido através de uma porta serial ao dispositivo embarcado utilizado na etapa de testes.

A utilização dessa abordagem, no entanto, não leva em consideração a variação de velocidade. Por fazer uso apenas de comandos relacionados a direção optou-se, em um primeiro momento, por manter a velocidade constante. Notou-se nesse ponto a necessidade de incluir ao sistema o suporte à tal variação, uma vez que é interessante e fundamental ao usuário alterar a velocidade da cadeira de acordo com o cenário que se encontra.

#### 3.7.2. Segunda Abordagem

Em uma segunda abordagem, optou-se por utilizar as coordenadas 2D, resultantes da Seção 3.5, aplicadas a uma matriz de posição, de dimensão proporcional ao tamanho do frame capturado (Seção 3.3). O processamento do sistema, como citado anteriormente, foi redistribuído de maneira que os recursos da *Raspberry Pi* fossem inteiramente destinados ao *software* de visão computacional. Além disso, a RPi transmite a cada *frame* – via porta serial – ao Arduino os pontos que compõem o vetor de posição, que por sua vez, realiza os cálculos algébricos para obter as direções e velocidades.

O *software* embarcado no Arduino tem como entrada as coordenadas dos pontos  $P_1$  – fixo na região compreendida entre o nariz e a boca – e  $P_2$  – situado no outro

extremo da reta, e variável conforme a posição da cabeça do usuário – que podem ser visualizados na Figura 4.

Para a classificação dos comandos foi tomada uma matriz nula quadrada  $A$  de ordem 9. O primeiro ponto referencia sempre o elemento central da matriz, denominado  $P_c$ . Chamou-se de  $\vec{v}$  o vetor que tem como ponto inicial  $P_1$  e ponto final  $P_2$ .  $P$  é o ponto (com coordenadas inteiras) resultante da soma de  $P_c$  com  $\vec{v}$ , multiplicado por um valor experimental  $0 < \alpha < 1$ .

$$P \approx P_c + \alpha \vec{v} \quad (1)$$

O elemento situado na linha e coluna da Matriz de Controle  $A$  – referente às coordenadas do ponto  $P$  – é preenchido com o valor 1, de modo a ocupar a posição da qual se extraiu os comandos. Foram também adotadas duas matrizes de velocidades de ordem 9, chamadas de  $Y$  e  $X$ , com a primeira do tipo coluna e a segunda do tipo linha.

As regiões centrais das matrizes  $Y$  e  $X$  possuem cada uma, entre os seus nove elementos, três correspondentes às velocidades nulas. A multiplicação das matrizes de velocidades pela matriz de controle  $A$  é feita como se segue:

$$V_y = A \cdot Y \quad (2)$$

$$V_x = X \cdot A \quad (3)$$

Da operação descrita acima, obtiveram-se dois vetores,  $V_y$  e  $V_x$ . Cada um desses vetores possui entre seus elementos apenas um valor diferente de zero, relativos à velocidade nos dois eixos de orientação. Um exemplo de tal operação para  $V_y$  está ilustrado na Figura 5. O procedimento para  $V_x$  é análogo, entretanto, o resultado será uma matriz do tipo coluna.

$$V_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 190 \\ 170 \\ 150 \\ 128 \\ 128 \\ 128 \\ 140 \\ 155 \\ 170 \end{bmatrix} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 140 \ 0]$$

Figura 5. Exemplo de obtenção da velocidade em  $V_y$

Estes valores contidos nas matrizes de velocidades foram previamente definidos através de testes e são semelhantes aos valores de ciclos PWM (*Pulse Width Modulation*) necessários para gerar as tensões equivalentes às que seriam produzidas pelo joystick original da cadeira.

### 3.8. Eletrônica de Arduino/Emulação

O circuito eletrônico para controle da cadeira de rodas motorizada, ilustrado na Figura 6, consiste em um microcontrolador *Arduino UNO*, conectado aos pinos do *joystick* da placa de controle da cadeira.

Para o acionamento dos atuadores, aproveitou-se o controlador original VR2 da cadeira de rodas motorizada. Os dados digitais do Arduino foram convertidos com o auxílio de um circuito conversor digital-analógico (DAC, do inglês *Digital-to-Analog Converter*). Para o processo de conversão, utilizou-se 4 filtros passa-baixa do tipo RC, de modo a suprimir a frequência característica do PWM do microcontrolador. Os sinais de tensão analógicos foram enviados diretamente ao barramento do joystick, onde o controlador natural da cadeira interpreta e aciona os motores.

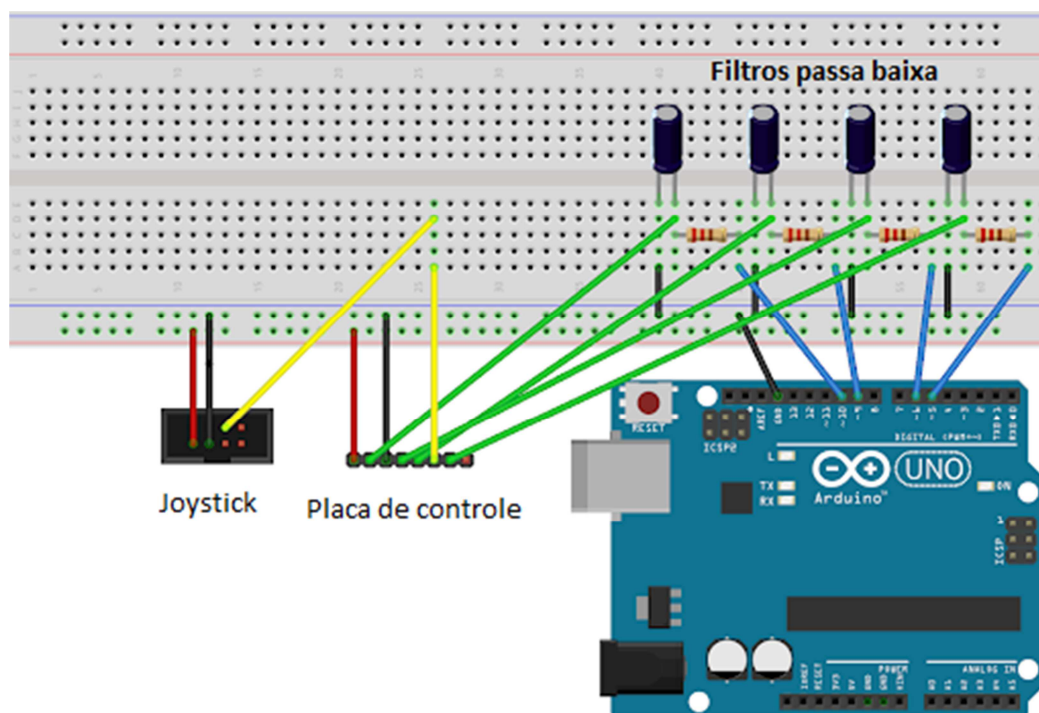


Figura 6. Descrição do circuito eletrônico.

O *joystick*, por sua vez, teve de permanecer conectado ao sistema da cadeira através de dois pinos de alimentação e um de verificação; este último responsável por detectar alterações e bloquear a cadeira de rodas motorizada quando não é verificada a verdadeira tensão analógica na placa de controle.



## 4. Etapa de Teste

De acordo com o que foi descrito nas seções anteriores, se fez necessária a realização de um conjunto de testes para cada uma das metodologias utilizadas.

### 4.1. Primeira Abordagem

Na primeira abordagem, para que esta etapa fosse a mais próxima da aplicação real objetivada, que se trata de uma cadeira de rodas, utilizou-se um protótipo do sistema, composto por um Arduino e dois Servo motores.

Foram realizados testes de interpretação dos comandos do sistema desenvolvido, de modo a testar se a captura e a resposta nos motores do protótipo condizem com a real intenção do usuário. Cada instrução foi testada 40 vezes, de modo que a transição entre os estados (direções) fosse o mais aleatório possível.

### 4.2. Segunda Abordagem

Na segunda abordagem utilizou-se o controle de velocidades, e as verificações foram realizados diretamente na cadeira de rodas motorizada. Assim, de posse do circuito necessário para a classificação, decisão e controle na aplicação real, foi avaliada a interpretação dos comandos de modo semelhante ao que foi realizado na primeira abordagem, com 40 amostras para cada modalidade de teste.

Desta vez, no entanto, cada direção verificada possuiu associada a si quatro subníveis de velocidade – nulo, baixo, médio e alto – levados em consideração de acordo com o ângulo de inclinação da cabeça do usuário.

## 5. Resultados

### 5.1. Primeira Abordagem

Na Figura 7 é apresentada a matriz de confusão resultante dos testes realizados na primeira abordagem. Para extrair os resultados com base na matriz, utilizou-se as métricas descritas em Andrade *et al.* (1997). Como resultado da etapa de testes obteve-se a Acurácia igual à 87,78 %, a Precisão de 90,00%, a Sensibilidade em 87,70% e uma Medida-F de 88,90%. O índice estatístico Kappa indicou que 86,25% dos comandos dados, foram interpretados corretamente.

a	b	c	d	e	f	g	h	i	
38	1	0	0	0	1	0	0	0	a = Esquerda
0	37	0	0	0	2	0	0	1	b = Direita
0	0	22	0	0	1	0	0	17	c = Frente
2	0	0	35	0	0	0	0	3	d = Trás
5	0	1	0	32	2	0	0	0	e = Esquerda/Frente
1	0	1	0	0	36	0	0	2	f = Direita/Frente
0	0	0	2	0	0	37	1	0	g = Esquerda/Trás
0	0	0	1	0	0	0	39	0	h = Direita/Trás
0	0	0	0	0	0	0	0	40	i = Parado

Figura 7. Matriz resultante dos testes da primeira abordagem.

Analisando a matriz de confusão (Figura 7) gerada, observa-se que os comandos de “Frente” foram os que tiveram menos resultados satisfatórios. No entanto, dos 18 casos de erro, 17 resultaram em movimento de parada e 1 de “Direita/Frente”. Ainda foi possível perceber a diminuição das incursões incorretas em algumas orientações próximas ao comando de parada.

## 5.2. Segunda Abordagem

Na Figura 8 estão descritas as matrizes de confusão resultantes dos testes da segunda abordagem. Para a extração dos resultados, utilizou-se as mesmas métricas descritas anteriormente. Os resultados gerais (de todas as direções individuais e seus subníveis de velocidade) mostraram uma acurácia média de 94,69%, precisão média de 91,04% e uma sensibilidade de 89,38%. A média do índice Kappa revelou a corretude de 85,83% dos comandos interpretados.

	A	B	C	D
A	40	4	1	16
B	0	36	2	0
C	0	0	37	0
D	0	0	0	24

(a) Frente

	A	B	C	D
A	40	0	0	0
B	0	40	0	0
C	0	0	40	3
D	0	0	0	37

(b) Trás

	A	B	C	D
A	40	2	0	0
B	0	38	3	0
C	0	0	37	7
D	0	0	0	33

(c) Esquerda

	A	B	C	D
A	39	1	0	2
B	1	24	7	0
C	0	15	33	4
D	0	0	0	34

(d) Direita

**Figura 8. Matrizes resultantes dos testes da segunda abordagem.**

Observou-se através dos índices individuais de cada matriz, que os comandos direita e frente foram os que apresentaram maior dificuldade de interpretação, no entanto, novamente os erros foram minimizados por resultarem em comandos semelhantes, de modo a alterar somente a velocidade na direção pretendida.

## 6. Conclusões e Trabalhos Futuros

Os resultados obtidos em ambas as etapas de testes demonstraram que o sistema possui um bom reconhecimento dos comandos, visto que as maiores falhas na interpretação resultam em uma orientação muito próxima à objetivada; evidenciando, porém, a necessidade de melhoria da implementação do interpretador de comandos.

Para trabalhos futuros objetiva-se a realização de testes em uma cadeira de rodas de baixo custo adaptada, além de testes com usuários reais, com a finalidade de validar a usabilidade do sistema. Propõe-se também a utilização do filtro de Kalman, descrito por Gonzalez e Woods (2007).

## Referências

- Andrade, A., Zicker, F., and Andrade, A. (1997). Avaliação de testes diagnósticos. *Métodos de investigação epidemiológica em doenças transmissíveis*, 1:9-30.
- Cerezetti, C. R. N., Nunes, G. R., Cordeiro, D., and Tedesco, S. (2012). Lesão medular traumática e estratégias de enfrentamento: revisão crítica. *O Mundo da Saúde*, 36(2):318-326.
- Cruz, J. E., Guimarães, L. N., and Shiguemori, E. H. (2012). Um estudo da detecção automática de campos de futebol de imagens aéreas e orbitais utilizando svm e descritores hog. In *XII Workshop de Computação Aplicada. São José dos Campos:[sn]*.

- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886-893. IEEE.
- Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International journal of computer vision*, 15(1-2):123-141.
- Gonzalez R. C. and Woods, R. E. (2007). *Digital Image Processing. Ed III*. Pearson Education Asia, New Delhi.
- Kaehler, A. and Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. "O'Reilly Media, Inc."
- King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755-1758.
- Kleiner, A. F. R. (2009). Mobilidade funcional em indivíduos com paralisia cerebral espástica.
- Medola, F. O., Castello, G. L., Freitas, L. N., and Busto, R. M. (1999). Avaliação do alcance funcional de indivíduos com lesão medular espinhal usuários de cadeira de rodas. *Revista movimenta*, 2(1).
- Osuna, E., Freund, R., and Girosit, F. (1997). Training support vector machines: an application to face detection. In *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pages 130-136. IEEE.
- Pinto, F. F. (2017). Interfaces de controle de cadeira de rodas motorizadas para pessoas com tetraplegia. Trabalho de conclusão de curso (graduação), Universidade de Brasília.
- Rahbi, Y., Mrabet, M. and Fnaiech, F. (2018). Intelligent control wheelchair using a new visual joystick. *Journal of Healthcare Engineering*, 2018.
- Soares, H. B. (2008). Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte.