

**CENTRO UNIVERSITÁRIO SENAC  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**GABRIEL VIEIRA FIGUEIREDO TOMAZ  
TALES CARLOS DE PÁDUA  
VINICIUS DE CARVALHO**

Mini Games com Visão Computacional

**SÃO PAULO  
MAIO DE 2014**

**CENTRO UNIVERSITÁRIO SENAC  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Gabriel Vieira Figueiredo Tomaz, Tales Carlos de Pádua,  
Vinicius de Carvalho.**

**viera\_frifri@hotmail.com, talescpadua@gmail.com,  
carvalho.v@outlook.com.**

**Mini Games com Visão Computacional**

"Pequenos jogos eletrônicos utilizando conhecimentos de Visão Computacional apresentados para a conclusão da disciplina Projeto Interativo III, do bacharelado em Ciência da Computação, Centro Universitário Senac."

Sob orientação do Prof.º: Marcelo Hashimoto

**SÃO PAULO  
MAIO DE 2014**

## 1 Resumo

Conforme proposto na disciplina de Projeto Interativo III, a partir do estudo de algoritmos relacionados à visão computacional foram desenvolvidos pequenos jogos eletrônicos (mini games) em linguagem C usando a interface gráfica provida pela biblioteca Allegro 5 e uma interface de acesso à câmeras de vídeo provida pela biblioteca OpenCV, de modo que a visão computacional oferecesse não apenas uma opção de controle para o jogador, mas sim um diferencial na experiência e imersão do usuário ao vivenciar os mini games.

Palavras-chave: jogos eletrônicos, visão computacional, Allegro 5.

## 2 Abstract

As proposed by the Interactive Project III discipline, from the study of algorithms related to computer vision were originated little electronic games (mini games) in C language utilizing a graphical interface provided by Allegro 5 library and a web cam access interface provided by OpenCV library, in order to make computer vision offer not only another controller option for the player, but a different experience and imersion for the user while playing the mini games.

Keywords: electronic games, computer vision, Allegro 5.

## 3 Introdução

A visão computacional é uma ciência e tecnologia voltada a lidar com a forma como as máquinas enxergam o mundo ao seu redor. As informações captadas por meio de sensores (como scanners, câmeras de vídeo, etc.) podem ser modeladas de diversas formas a fim de suprir necessidades que permeiam desde ramos diretamente ligados à tecnologia de informação (como robótica e áreas de automação tecnológica) até os que se utilizam da tecnologia para dadas outras necessidades, como ciências ambientais, medicina e outros.

Com o objetivo de dar um passo inicial para dentro da visão computacional, este trabalho visa utilizar técnicas e algoritmos da mesma aliada à captação de imagens por câmera de vídeo para a produção de jogos simples, mas que mantenham a jogabilidade focada no poder da visão computacional, de modo que a experiência do jogador, ao invés de ser restringida pela interface proposta, se torne um diferencial por conta deste quesito.

## 4 Revisão da Literatura

%% TODO referenciar trabalhos utilizados como base para nossos jogos, etc.

## 5 Desenvolvimento

### Ponto de Partida

Visto que este trabalho foi o primeiro contato formal com a visão computacional por parte do grupo, a estratégia adotada desenvolver os mini games foi uma via de mão-dupla passando por um *brainstorm* de jogos existentes até quais algoritmos poderiam modelar uma interface de controle aceitável para os mesmos e fazendo o caminho de volta, onde eram estudados algoritmos existentes e se imaginava o que era possível, em termos de jogos, produzir a partir deles.

A partir desta metodologia aliada à orientação e pesquisa, surgiram as técnicas e algoritmos a seguir e a consequente combinação dos mesmos para elaboração dos jogos.

No início do trabalho foi introduzido uma biblioteca, baseada em OpenCV, que faz a interface de acesso à câmera de maneira bem restrita, possibilitando apenas que houvesse contato com os quadros capturados da câmera fornecidos por uma matriz tridimensional onde a primeira dimensão é a altura (em pixels) da imagem, a segunda representa a largura (também em pixels) e a terceira são os componentes vermelho (red), verde (green) e azul (blue), respectivamente, do espaço de cores RGB. Todos os valores da matriz representam um número entre 0 e 255 do padrão RGB.

### Jogo Genius

Esse jogo consiste em um remake do classico jogo genius com a jogabilidade alterada para ser possível jogar com imagens capturadas pela web cam.

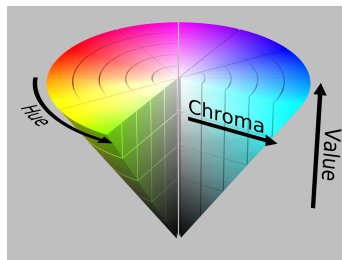
A principal ideia do jogo se baseou no reconhecimento básico da cor vermelha, onde se era verificado se o componente R do pixel era maior que a soma dos outros dois G e B, conseguindo detectar com sucesso a cor desejada. Apesar de funcionar relativamente bem para detecção de vermelho, outras cores apresentavam uma resistência maior ao método devido a pequenas instabilidades e mudanças de luz. Sendo necessário buscar algum outro método que nos pudesse fornecer uma melhor precisão nesse reconhecimento . Dentre alguns pesquisados , a solução adotada foi a conversão do espaço de cor RGB para o HSV pois com ele, definir um range e uma intensidade para a detecção de uma cor especifica fica muito mais preciso e intuitivo , devido suas características unicas.

Color	Hue	Saturation	Value
Black	$0^\circ < H < 360^\circ$	$0 < S < 1$	$V < 0.1$
White	$0^\circ < H < 360^\circ$	$S < 0.15$	$V > 0.65$
Gray	$0^\circ < H < 360^\circ$	$S < 0.15$	$0.1 < V < 0.65$
Red	$H < 11^\circ, H > 351^\circ$	$S > 0.7$	$V > 0.1$
Pink	$H < 11^\circ, H > 351^\circ$	$S < 0.7$	$V > 0.1$
	$310^\circ < H < 351^\circ$	$S > 0.15$	$V > 0.1$
Orange	$11^\circ < H < 45^\circ$	$S > 0.15$	$V > 0.75$
Brown	$11^\circ < H < 45^\circ$	$S > 0.15$	$0.1 < V < 0.75$
Yellow	$45^\circ < H < 64^\circ$	$S > 0.15$	$V > 0.1$
Green	$64^\circ < H < 150^\circ$	$S > 0.15$	$V > 0.1$
Blue-green	$150^\circ < H < 180^\circ$	$S > 0.15$	$V > 0.1$
Blue	$180^\circ < H < 255^\circ$	$S > 0.15$	$V > 0.1$
Purple	$255^\circ < H < 310^\circ$	$S > 0.5$	$V > 0.1$
Light Purple	$255^\circ < H < 310^\circ$	$0.15 < S < 0.5$	$V > 0.1$

Range das cores em HSV

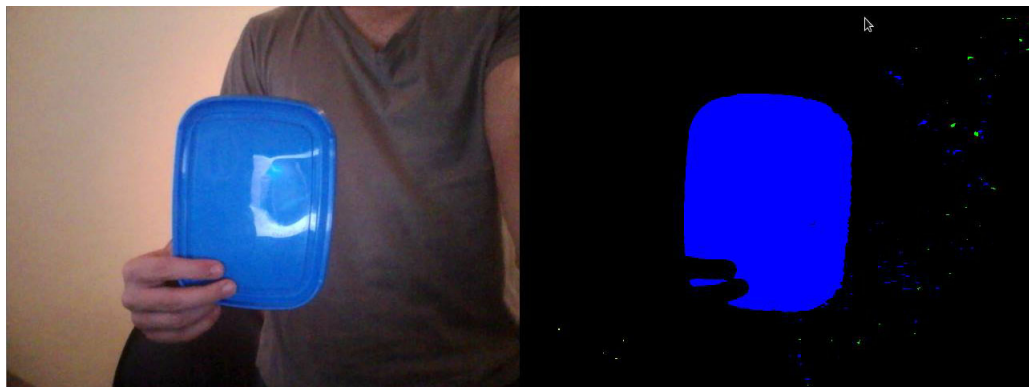
## Espaço de cor HSV

O sistema de cores HSV formadas pelas componentes Hue (tonalidade), Saturation (Saturação) e Value (Valor). Esse sistema também é conhecido como HSB (Hue, Saturation e Brightness - Tonalidade, Saturação e Brilho, respectivamente). A primeira componente H define a cor propriamente dita, podendo variar de 0 a 360 graus, a segunda componente S define a pureza ou intensidade da cor contida na componente H e por último a componente V define o brilho da componente H.



**Espaço de cor HSV**

Como resultado da conversão para esse espaço de cor podemos ver um exemplo da detecção de azul :



## Jogo: Sorvete Hoje?

A ideia deste jogo surgiu a partir do estudo de certos algoritmos que aliados permitem uma detecção de borda, que foi a base para a detecção de face. A detecção desta borda é feita por meio de um Filtro de Sobel, que será explicado mais adiante. Aliado ao filtro, para que houvessem bordas mais consistentes, foi utilizado um algoritmo de binarização de imagem. A técnica utilizada é conhecida como algoritmo de Otsu.

Apenas o filtro, porém, não era suficiente para atingir os objetivos, visto que a manipulação de quadros da câmera pode ser um processo bastante custoso, o que gera uma experiência ruim para o usuário. Visando um modo de se otimizar este processo, antes de se aplicar o algoritmo do filtro foi utilizada uma conversão da imagem para escala de cinza.

Por fim, para evitar impecilhos com a imagem de fundo da câmera, também foi agregado um algoritmo de remoção de fundo simples utilizando a fórmula matemática da Distância Euclidiana nos pixels dos quadros.

### **Distância Euclidiana**

%% TODO: distância euclidiana

### **Detecção de Cor**

%% TODO: detecção de cor

### **Escala de Cinza**

%% TODO: grey scale

### **Binarização**

%% TODO: binarização

### **Filtro Sobel**

%% TODO: sobel operator para detecção de bordas

### **Filtro Gaussiano**

%% TODO: distância euclidiana

## **6 Resultados**

%% TODO o que conseguimos produzir com os algoritmos citados no desenvolvimento, e descrever por que funcionou fazer os jogos desta forma.

## **7 Considerações Finais**

%% TODO conclusões sobre o nosso trabalho.

Ex: Flappy Bino: estudamos X algoritmo e Y técnica mas não deu certo por motivo A.

Jogo do Mexe-Mexe: com o X algoritmo do Flappy Bino conseguimos fazer o jogo com sucesso por motivo B.

## **8 Referências Bibliográficas**

%% TODO formatar referências em ABNT

- Livros do nosso Google Drive sobre Visão Computacional
- Livro de Álgebra Linear da aula da Dani com a distância euclidiana
- Tutoriais do OpenCV