

Jogo da soma - Computação Visual

Aloisio Kleyner da S. Lima Junior¹, Leandro P. Sampaio²

¹Engenharia de Computação - Universidade Estadual de Feira de Santana (UEFS)
44036-900 – Feira de Santana – BA – Brazil

kleyner2@hotmail.com, leandrosampaio827@gmail.com

1. Introdução

A computação visual é um ramo da computação responsável por fazer a captura e tratamento de imagens conforme as necessidades para a mesma. Dentro do componente que trata desde assunto, como trabalho final e objeto de estudo do presente relatório, foi desenvolvido um software capaz de fazer o reconhecimento das quantidades de dedos visíveis em duas mãos e mostrar a contagem dos mesmos.

Para que o projeto fosse desenvolvido, foi necessário o estudo de diversos assuntos (segmentação de imagens, reconhecimento de formas) dentro da computação visual para que o mesmo fosse colocado em prática, que serão abordados e discutidos neste relatório, bem como o processo de desenvolvimento do programa que foi obtido como resultado final.

2. Fundamentação Teórica

Nas seguintes subseções serão abordados os conceitos utilizados para a resolução do problema proposto.

2.1. Computação Visual

A computação visual abrange dois conceitos:

- 1) um ambiente no qual humanos programam, monitoram e controlam processos conduzidos por computadores, por meio da manipulação de imagens visuais;
- 2) captura, reconhecimento, modificação e renderização de arquivos de mídia visual extremamente grandes.

A computação visual faz uso de imagens para permitir que as pessoas enxerguem, controlem e interajam com dados visualmente. Ela oferece novas perspectivas e capacidades poderosas para os setores de manufatura, pesquisa, educação e treinamento, incluindo interação entre humanos e computadores, controle de qualidade, imagens médicas, medições, robótica, multimídia, modelagem e animação faciais computadorizadas, arquivamento de patrimônio, videogames, realidade virtual, efeitos especiais em filmes, entre muitos outros.

2.2. OpenCV

No projeto do jogo da soma foi utilizado o OpenCV que é a maior biblioteca de código aberto para a visão computacional, processamento de imagem e aprendizagem de máquina. O OpenCV (Open Source Computer Vision Library) foi lançado sob uma licença BSD e, portanto, é gratuito para uso tanto acadêmico quanto comercial. Possui interfaces C++, C, Python e Java e suporta Windows, Linux, Mac OS, iOS e Android. O OpenCV

foi projetado para eficiência computacional e com foco forte em aplicações em tempo real. Escrito em C / C ++ otimizado, a biblioteca pode aproveitar o processamento multi-core. Ativado com o OpenCL, ele pode aproveitar a aceleração de hardware da plataforma de computação heterogênea subjacente.

2.3. Hsv

HSV é a abreviatura para o sistema de cores formadas pelas componentes hue (matiz), saturation (saturação) e value (valor). Esse sistema de cores define o espaço de cor conforme descrito abaixo, utilizando seus três parâmetros:

Tonalidade ou Matiz:

- Medida do comprimento de onda médio da luz que ele reflete ou emite define a cor do objeto.
- Atinge valores de 0° a 360° , mas para algumas aplicações, esse valor é normalizado de 0 a 100

Saturação:

- Também chamado de "pureza". Quanto menor esse valor, mais com tom de cinza aparecerá a imagem. Quanto maior o valor, mais "pura" é a imagem.
- Expressa o intervalo de comprimentos de onda ao redor do comprimento de onda médio no qual a energia é refletida ou transmitida.
- Atinge valores de 0 a 100

Valor ou Brilho:

- Define o brilho da cor. Atinge valores de 0 a 100

2.4. Convex Hull

O algoritmo do Convex Hull (Envoltório Convexo) consiste em encontrar o menor número de pontos que gerem um polígono convexo no qual abranja todos os outros pontos. A figura abaixo é um exemplo do convex hull. Dos vários pontos, somente os pontos mais externos e que formam o menor polígono que “fechem” todos os outros pontos.

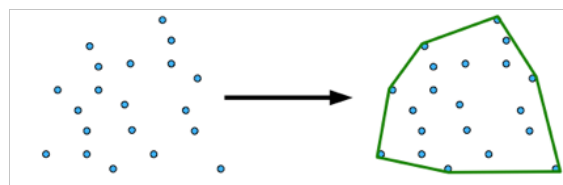


Figura 1. Convex Hull

O convex hull pode ser expandido para 3 ou mais dimensões. Portanto, não se limita ao plano. Além disso, ele pode ser aplicado em diversos problemas: ray tracing, videogames, busca de caminhos em robótica, sistemas de informação geográfica, entre vários outros.

3. Metodologia

No processo de captura da imagem até a exibição da informação desejada em tela, diversos processos são executados afim de tornar válida a informação que é transmitida para o usuário. Neste programa em estudo, o primeiro processo, como tudo que envolve a computação visual, é a captura de imagem, sendo esta obtida através de uma câmera digital. O programa então recebe os dados do sensor da câmera e converte em dados para serem tratados.

Declarado todas as variáveis imagens que serão gerados para análise e tratamento dos dados, uma delas é responsável por receber as imagens da câmera, e a partir desta, realizar os devidos processos. Duas das imagens serão responsáveis por selecionar duas áreas dentro da imagem principal, cada uma delas sendo de uma das mãos que serão analisadas pelos algoritmos. Isto faz com que exista uma menor quantidade de informação a ser analisada, assim, otimizando o tempo de processamento.

Um segundo par de variáveis de imagens serão responsáveis por fazer a seleção do que será utilizado para o processamento baseado em determinados intervalos de cores, sendo estas correspondentes ao mais próximos tons de cores da pele humana. Para isto, foi utilizado um algoritmo que seleciona cores no intervalo determinado e "binariza" a imagem, tornando em branco o que se encontra na faixa e em preto o que está fora da faixa de cor.

Para evitar que ruídos criassem furos da imagem binária gerada, a mesma foi submetida a transformação morfológica de fechamento, que soluciona este problema. Após o fechamento, para que a imagem gerada seja melhor analisada pelas funções posteriores, ela foi submetida a um duplo processo de dilatação da imagem, para que as linhas se tornem mais espessas e com isso, os dados sejam melhores analisados e com menos erros na execução dos processos.

Estas duas imagens são submetidas ao Gaussian Blur, que é responsável por reduzir os ruídos na imagem, logo, removendo as imperfeições nas formas das imagens capturadas, que necessitam serem o mais consistentes possíveis para uma melhor análise posterior. O threshold é utilizado para fazer a limiarização da imagem em tom de escala de cinza.

Obtida as imagens segmentadas é utilizado o Convex Hull para a seleção dos pontos da imagem correspondente aos dedos da mão. Através da seleção dos pontos é feito a contagem dos mesmos, assim, calculando a quantidade de dedos em cada mão e ao final, dando a soma da quantidade de dedos nas duas mãos pro usuário. No caso do usuário não colocar nenhuma mão, o sistema não identificará nenhum elemento a ser analisado, pois ele foi segmentado a analisar apenas o aspecto de cor da pele, logo aparece uma mensagem pedindo para que o usuário insira as mãos nas posições corretas, conforme mostrado na Figura 2.

A Figura 3 mostra o sistema funcionando, onde ao mostrar três dedos numa mão e dois dedos em outra, o número cinco é mostrado na tela, que corresponde a soma destes dois números.

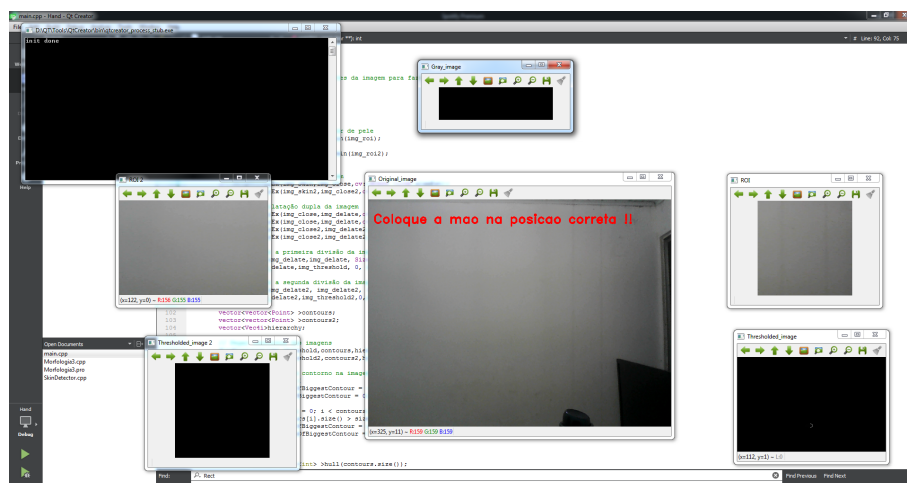


Figura 2. Mensagem informando a falta de mãos na tela

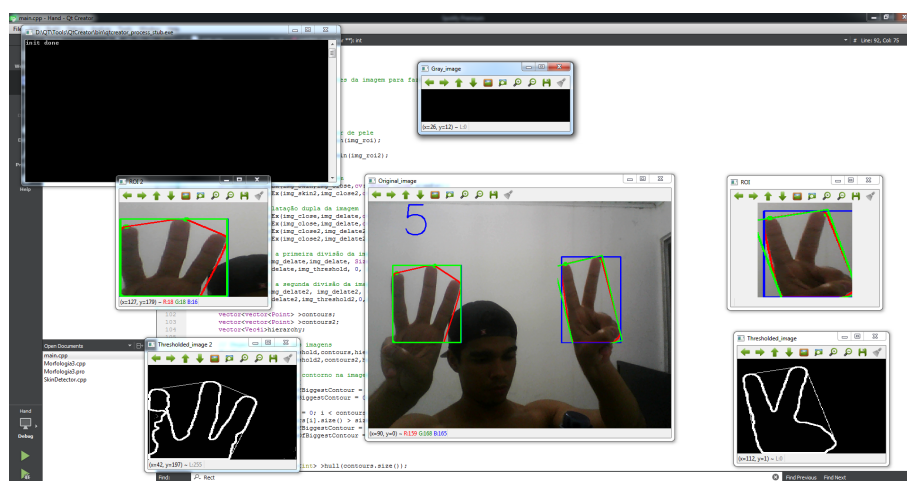


Figura 3. Programa em funcionamento

4. Conclusão

O jogo da soma desenvolvido é capaz de levar entretenimento e aprendizagem para as pessoas, demonstrando um jogo simples de ser utilizado e funcional. Algumas melhorias poderiam ser utilizadas como o aperfeiçoamento da segmentação visando uma melhor taxa de acerto na quantidade de dedos do usuário.

Referências

<https://www.hpe.com/br/pt/what-is/visual-computing.html>
<https://handmap.github.io/convex-hull-and-contours/>
<https://picoledelimaio.github.io/blog/2015/11/15/fingertip-detection-on-opencv/>