

Detecção de Gestos Com a Mão

Cleyton José Rodrigues da Silva

Departamento de Computação

Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Recife, PE – Brasil

cleytonjrodrigues98@gmail.com

Abstract. *This report-paper describes the steps on the production of an image processing model for hand-gesture detection. Details on the implementation of these steps will be explained, which includes the image processing operations utilized for reaching the proposed goal. This will range details starting from the color treatment steps, to hand-gesture classification.*

Resumo. *Este relatório objetiva detalhar as etapas da produção de um projeto de processamento de imagem cuja proposta é a detecção de gestos de mão. Serão explicados detalhes a implementação dessas etapas, bem como as operações de processamento de imagens utilizadas para alcançar o objetivo, indo do tratamento de cores para identificação de tons de pele, até a classificação dos gestos de mão.*

1. Introdução

Numa sociedade cujo estilo de vida se torna cada vez mais integrado a tecnologia, a questão da interação homem-máquina é inevitável. Diversos meios de interação com a tecnologia vêm se desenvolvendo ao longo dos anos, e a interação por gestos capturadas em vídeo para realização de operações pode ser uma alternativa aos comandos normalmente realizados por periféricos como mouses ou telas sensíveis ao toque.

Este projeto tem o propósito de produzir um modelo de processamento de imagem que realize a detecção de mãos e classifique seus gestos de acordo com algumas das técnicas apresentadas originalmente no artigo “*Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware*”, o qual este projeto se inspirou. A base de dados utilizada para conseguir amostras foi a *Database for hand gesture recognition* do projeto *Hand Detection and pose Estimation* que envolve o Dr. e pesquisador Michal Kawulok da Silesian University of Technology [4]. Boa parte das operações usadas na implementação se dão pelo uso da API OpenCV através da linguagem Python e também é feito uso da biblioteca Numpy.

2. Pré-processamento e Segmentação

De modo a proporcionar a detecção da mão numa dada imagem, será necessário realizar algumas etapas que resultarão na porção a que a classificação dos gestos poderia feita, isto é, a Região de Interesse na imagem. Como parte do pré-processamento inicialmente necessita-se remover a informação de fundo que representa

o plano de trás em relação a que uma pessoa e/ou sua mão estará na imagem. Neste projeto o pré-processamento foi colocado como unido à segmentação aplicada aqui, devido à natureza da técnica utilizada.

Visto que se lida aqui com informações de pele, foi utilizada uma técnica de segmentação de pele encontrada em artigos como [1], [2] e [3]. É explicado em [1] e [2], diante de experimentos, que com o espaço de cor YCrCb tem-se isolado um componente de luminância (Y) que não afeta a indicação de cor feita pelos outros dois componentes (Cr e Cb), e isso simplifica a identificação do alcance de cores específicas que representem tons de pele. Tais valores de alcance são encontrados levemente variados em torno de artigos como [2] e [3] que lidam com a segmentação de pele por meio desse espaço de cor. Portanto, aqui o tratamento de cores é feito de modo a destacar na imagem valores de interesse que indiquem a presença de pele na imagem, onde provavelmente se encontra a região da mão.

Neste ponto inicial, a intenção é de extrair/remover o fundo da imagem, então como primeiro passo para isto, é utilizada a técnica de thresholding sob o algoritmo Otsu em cada um dos três canais Y, Cr e Cb da imagem de modo a remover o máximo de detalhes que não são de interesse para a detecção feita aqui. Como pré-processamento para tal operação de thresholding, aplicou-se a operação de suavização com filtro Gaussiano, com o parâmetro de kernel (17,17) para definir a intensidade, seguido da operação morfológica de Dilatação. Tais operações têm a intenção de reduzir a quantidade de informações dispersas, a fim de fornecer uma imagem que é mais uniforme em cor e detalhes possível, em preparação à limiarização realizada a seguir.

Como resultado disso (fig 2) temos uma imagem cujos detalhes ignoráveis de fundo não estão em tanto destaque quanto valores onde provavelmente se encontra pele.



Fig. 1. Imagem de amostra

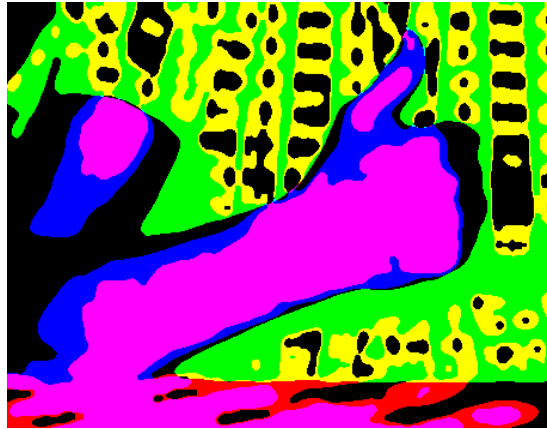


Fig. 2. Amostra pós operação de threshold por canal YCrCb

Em adição ao processamento explicado anteriormente, foi notada uma condição interessante quando temos tons de cinza ou mais próximos do branco em roupas: ao passar pela limiarização, tais valores de pixel ficavam muito próximos aos valores esperados para as regiões de pele na imagem (Fig. 3). Contatou-se que nesses casos específicos, a limiarização inicial iria ativar tais pixels de roupa em todos os canais da imagem pós-limiarização ainda em YCrCb, o que não permitia uma execução favorável do próximo passo da remoção de fundo a ser feita aqui. Então para estes casos foi aplicada uma busca por estes pixels que estariam representados pelos valores [255, 255, 255], para então torná-los preto [0, 0, 0].

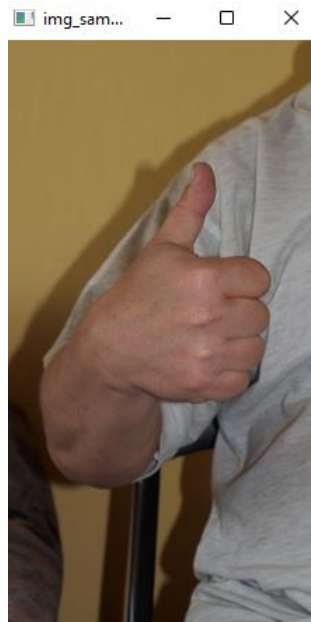


Fig. 3. Amostra original com roupa cinza-claro

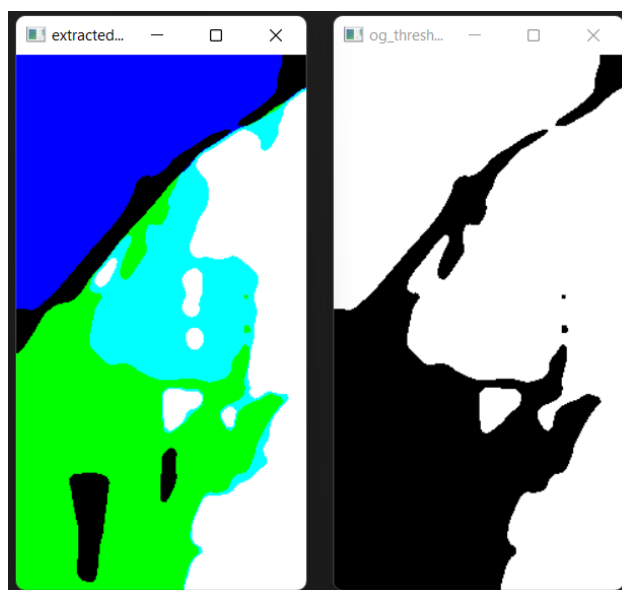


Fig. 4. Amostra da Fig. 3. pós-limiarização YCrCb

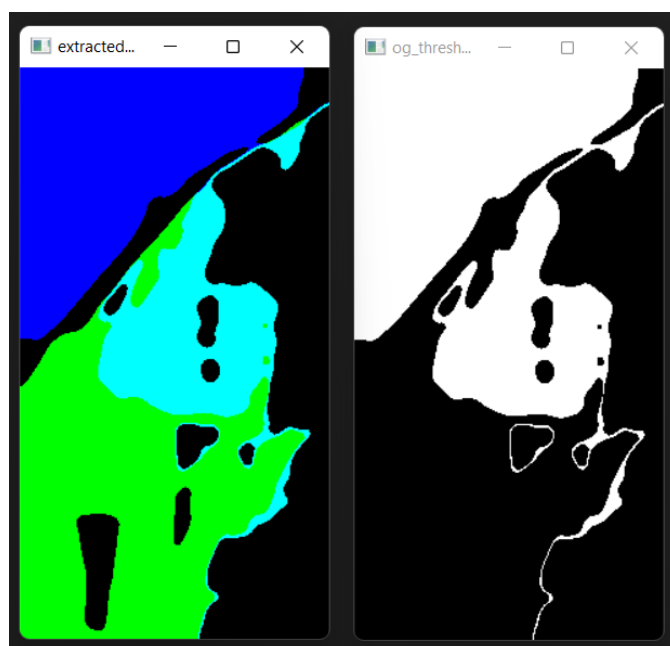


Fig. 5. Amostra da Fig. 3. pós-limiarização YCrCb e desativação dos pixels de valor [255, 255, 255].

No segundo passo que resulta na remoção do fundo, utilizou-se de fato a segmentação de imagem citada anteriormente e encontrada em [1], [2] e [3]. Na prática, o processamento feito até aqui já pode ter fornecido uma imagem cujo fundo está removido, principalmente numa amostra cujo fundo é muito simples e de poucas cores, nota-se na Fig. 6. Entretanto, como forma a produzir uma remoção mais robusta de fundo, foi feito a operação de limiarização na imagem original em YCrCb não pre-processada, desta vez usando os valores de alcance indicados [3], respectivamente para os canais Y, Cr e Cb:

- (54, 131, 110) para os valores mínimos do alcance;

- (163, 157, 135) para os valores máximos.

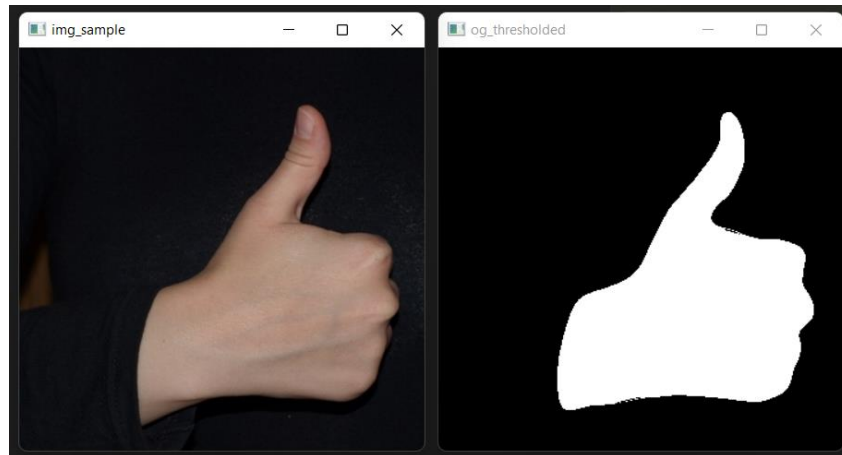


Fig. 6. Amostra de fundo simples pós-limiarização YCrCb e limiarização da escala de cinza.

Extraídas todas as porções da imagem que estiverem dentro deste alcance, ainda foram feitas operação de suavização e a operação morfológica *Open*, que basicamente realiza uma operação de erosão seguida por uma dilatação. Com essa imagem preparada, aplicou-se na imagem pré-processada do primeiro passo, um filtro de cinza, a fim de destacar por meio de limiarização as regiões que estarão em mais destaque e que provavelmente contém pele. Será feita então uma operação lógica AND entre a imagem pré-processada e a imagem que aplicou a segmentação de pele por alcance de valores YCrCb, para resultar na imagem em que será feita a detecção da mão.



Fig. 7. Resultados da imagem pós-limiarização YCrCb e limiarização da escala de cinza (esquerda) e a imagem da segmentação de pele por limiarização dentre alcance de valores



Fig. 8. Resultado da operação BITWISE_AND das imagens mostradas na Fig. 7.

3. Detecção de Bordas e da Mão

Definida o que pode se chamar de “máscara de pele” da imagem original, é preciso agora definir o objeto que a mão representa, bem como sua forma, e isso é feito inicialmente com a detecção de bordas.

3.1. Definição de Bordas Canny

O algoritmo usado aqui para detecção de bordas foi o popular Canny, desenvolvido por John F. Canny. Neste algoritmo, é usado de filtros de lineares com kernel Gaussiano para suavizar as bordas, e computar a força de cada borda, bem como a direção de cada pixel contido na imagem. Pixels mais fortes, que sobrevivem um processo de “afinamento” da imagem por um processo conhecido por “*Non-maximum Suppression*”, vão ser identificados como bordas. Este algoritmo existe para utilização na biblioteca OpenCV utilizada em todo este projeto.



Fig. 9. Resultado de uma detecção de bordas Canny

3.2. Contornos e Aproximação de polígonos

Com o algoritmo Canny fornecendo as bordas do que será a mão, partiu-se para finalmente a detecção da mão como objeto. Os Contornos são as curvas que unem toda um contínuo de pontos nas bordas, e ajudam justamente a detectar o objeto e sua forma. Isso é feito aqui, e para identificar a região de interesse que contém a mão, buscamos levar em conta para processamento apenas unidades de contornos cujo tamanho é maior que um limiar definido a ajuste fino neste caso. Assim tem-se a intenção de evitar pequenos ruídos possivelmente detectados, ou regiões que não são do tamanho ou parte de uma mão.

Seguinte, é feita a Aproximação de polígonos de contornos, aqui usando a implementação pela API Open CV do algoritmo de Douglas-Peucker, onde se define polígonos ao aproximar as formas de contidas num conjunto de contornos sob uma quantidade mínima de vértices, a depender da precisão dada. Para este projeto, a Aproximação de polígonos basicamente vai definir um conjunto de polígonos em cima dos contornos da mão detectada.

3.3. *Maximum Inscribed Circle* e *Minimum Enclosing Circle*

O Maximum Inscribed Circle, ou Maior Circulo Inscrito, se trata do maior circulo possível a ser contido dentro da figura ou contornos que nesse caso compõem a mão. Isto é feito descobrindo primeiro qual ponto dentro da figura é seu centro, de acordo com o critério de que este é o ponto mais distante dos outros pontos, do que qual distância entre outros pontos comparados entre si. Aqui utilizaríamos este ponto central para detectar como ele se localiza em relação a maneira que os dedos estão dispostos.

O Minimum Enclosing Circle é o menor círculo possível que envolve a figura da mão detectada aqui, uma implementação também da API OpenCV usa como parâmetro os contornos da mão detectada, para retornar o ponto central da figura e o raio desse círculo.

3.4. *Convex Hull* e Defeitos de convexidade

A envoltória convexa, ou *ConvexHull* é dada como um embrulho de toda a figura detectada, e indica as concavidades entre as curvas na figura, o que para uma mão, significa que a distância dos dedos pode estar causando esta curva. Os defeitos de convexidade se dão justamente como a unidade que representa estas concavidades embrulhadas pela envoltória convexa.

Com o uso da OpenCV, cada unidade de defeito é dada por uma estrutura de vetor de quatro índices, que de 0 a 4, respectivamente, representam o início, o fim, o ponto mais distante, e a profundidade do defeito, este último valor significando a distância entre o ponto mais distante (que neste caso se localiza entre os dedos).

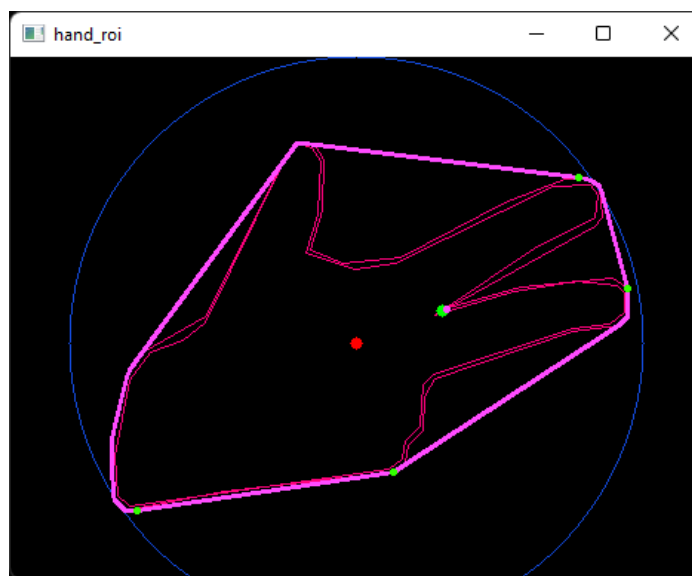


Fig. 10. O desenho da aproximação de polígonos dos contornos; os pontos centrais do max inscribed circle e o próprio círculo; a envoltória convexa e os defeitos de convexidade.

4. Melhorias no algoritmo de detecção

A fim de buscar melhorias nos resultados a serem obtidos por este modelo de detecção de mão, foram necessários ajustes simples, mas instrumentais para execução da tarefa em questão. Como se constatou, a porção do projeto que isola o objeto de mão se dá com o processo de remoção de fundo com a segmentação de pele na imagem, portanto melhorias com ajustes finos, e filtragem otimizada foram aplicadas nessas etapas do modelo.

4.1. Melhorias na segmentação de pele

Apesar de inicialmente servirem de maneira decente para o objetivo de segmentação de pele, o uso dos valores de limiarização dos canais YCrCb para esse propósito precisaram ser ajustados a fim de se obter um resultado de segmentação melhorado. Antes tínhamos os parâmetros de limiarização:

- $54 < Y < 163$
- $131 < Cr < 157$
- $110 < Cb < 135$

Seguindo os estudos [5] e [6], foram usados a partir de então os valores parâmetros de limiarização:

- $54 < Y < 163$ (permanecendo igual a forma inicial)
- $133 < Cr < 177$
- $77 < Cb < 121$

Utilizando destes parâmetros, foi possível obter resultados de segmentação mais satisfatórios, com a segmentação apresentada isolando melhor as regiões de uma imagem amostra que de fato contém pele. Ao analisar a figura 11, nota-se como o uso destes novos valores de parâmetros produzem uma segmentação que confunde menos elementos da imagem possivelmente de tons parecidos com valores de pele, evitando vazamentos de elementos para esta região de interesse inicial, como ocorre mais intensamente a esquerda.

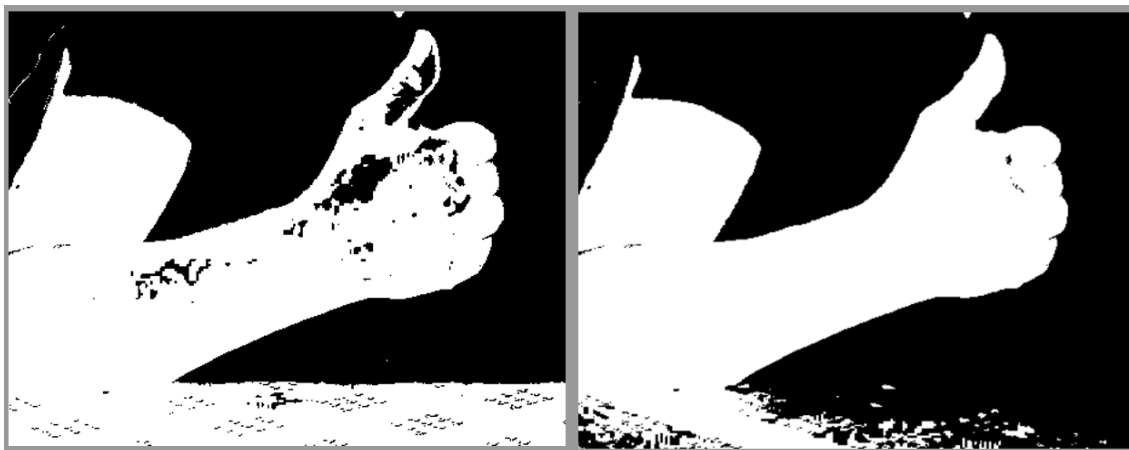


Fig. 11. Segmentação por limiarização com parâmetros antigos (esquerda) e com parâmetros ajustados (direita).

4.2. Melhorias com aplicação dos filtros morfológicos de Erosão e Abertura

Os filtros de Erosão e Abertura precisaram ser aplicados de maneira consciente nessas que são as etapas de anteriores à definição de bordas, para que surtissem efeito eficiente contra os problemas que existiam ambos na subtração de fundo, e durante a segmentação de pele.

A operação de Erosão busca erodir as bordas que circundam a definição de um dado objeto. Na prática, tendo o objeto de mão, por exemplo, este filtro causa um afinamento neste objeto; dedos ficam mais finos, bem como a mão reduzida. No propósito deste projeto, o uso do filtro de erosão foi muito efetivo na remoção de ruídos que eram resquícios da segmentação de pele. Isto é de suma importância para se produzir um objeto que representa o máximo da mão ou região de pele isolada.

De forma semelhante, o filtro de Abertura também vai ser utilizado aqui para remoção de ruídos que circundam o objeto de interesse detectado. Consistindo em, numa única aplicação, erodir a imagem e em seguida fazer a dilatação, temos que o objeto maior acaba se mantendo, praticamente, do mesmo tamanho, enquanto ruídos

finos são eliminados da imagem resultante. Este filtro foi aplicado tanto durante a subtração de fundo, quanto segmentação de pele.

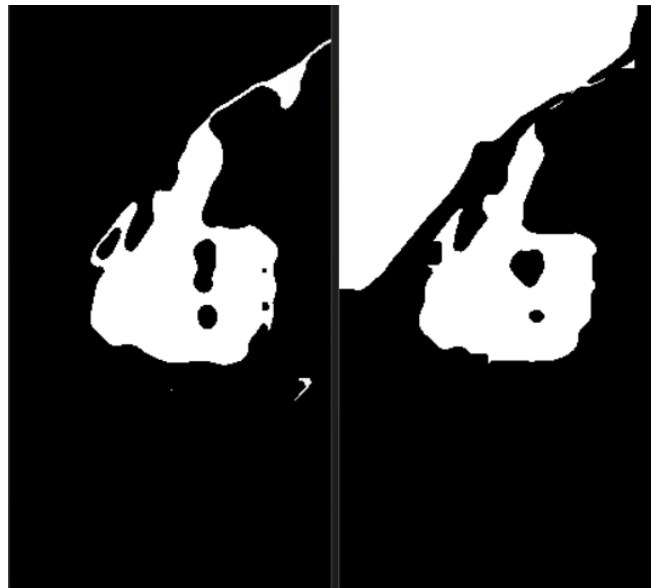


Fig. 12. Objeto de mão detectado antes da aplicação do filtro de Erosão (esquerda) e o mesmo objeto após a aplicação do filtro (direita).

Tais operações de aplicação simples permitiram produzir um objeto de mão detectado muito mais adequado para o passo de detecção de bordas e aproximação de polígonos, pois foi possível tornar tal objeto mais consistente com o que se espera da região de mão a ser detectada na imagem. Onde antes, tinha-se uma definição de contornos extremamente insatisfatória justamente devido a uma detecção fraca, principalmente em imagens mais complexas, agora um objeto bem definido por sua vez permite aceitável detecção de bordas. Nota-se que também foi aumentado o limiar de tamanho de contornos detectados, a fim de se ignorar possíveis contornos cujo tamanho são menores que objeto de mão.



Fig. 12. Definição de Bordas, Aproximação de Polígonos e Envolutória Convexa em cima da detecção como feita inicialmente.

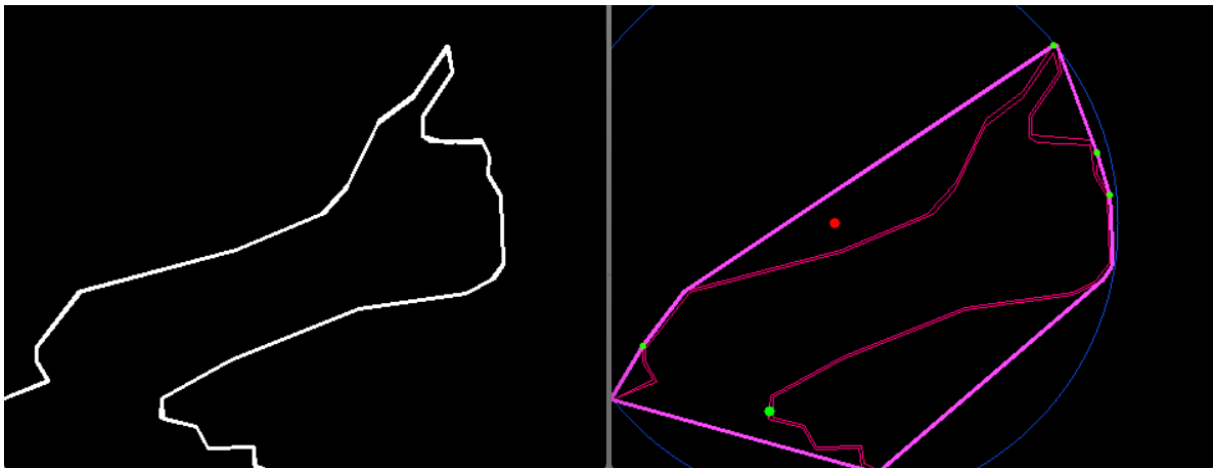


Fig. 13. Definição de Bordas, Aproximação de Polígonos e Envolutória Convexa em cima da detecção pós aplicação das melhorias.



Fig. 13. Resultado da detecção de mão a partir da amostra da figura 3, feita da forma inicial (esquerda) e após as melhorias (direita)

5. Conclusão

O projeto de detecção de mão se provou desafiador por exigir um número de tarefas não exatamente básicas, e cujo modelo final não se tornou tão robusto, em conseguir sempre resultados fortes acerca de toda e qualquer amostra processada. Da forma que se fez o projeto, a fase de pré-processamento é simplesmente primordial para o sucesso da detecção da mão, pois é visto que, a partir do momento que se define a máscara de pele, tudo da definição dos contornos em diante, depende da boa definição

do objeto/mão (evitando que ele seja “vazado em seus contornos”), para sucesso das etapas de detecção que se implementou em sequência.

A segmentação de pele por limiarização sob alcance de valores de cor YCrCb, se mostrou limitado, para substituir uma técnica mais robusta de segmentação de fundo, se não usada sob refinamentos e pré-processamento de detalhes; como foi visto com as melhorias, a aplicação de filtros morfológicos que ajudem na definição do objeto mão de interesse foram importantíssimos para conseguir resultados mais robustos onde outrora não eram possíveis.

4. Referências

- [1] - Sajaa G. Mohammed a, Abdulrahman H. Majeed a, Ali Aldujaili b, Enas Kh. Hassan c, Safa S. Abdul-Jabbar d. (2020) “Image Segmentation For Skin Detection”.
- [2] - Varsha Powar and Aditi Jahagirdar and Sumedha Sirsikar. (2012) Skin Detection in YCbCr Color Space.
- [3] - H. Yeo, B.G. Lee, and H. Lim. (2013) “Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware”.
- [4] - Michal Kawulok, D.Sc.. Database for hand gesture recognition. Silesian University of Technology, <https://sun.aei.polsl.pl/~mkawulok/gestures/>.
- [5] Vinícios Cavalcanti Nogueira de Sá. (2018) “Detecção de Mãos Através da Combinação de Técnicas de Detecção de Tom de Pele e Movimento para Background Complexo”. Monografia, DEINFO, UFRPE.
- [6] BASILIO, J. A. M. et al. (2011) “Explicit image detection using ycbcr space color model as skindetection”. Applications of Mathematics and Computer Engineering, p. 123–128.