Segmentação de Faces em Vídeos com uso de *Supervoxels* e dados *Kinect V2*

Filipe Teixeira, 14/0139486
Lucas Santos, 14/0151010

***Resumo*—Este documento apresenta a implementação de um algoritmo que realiza a detecção de faces em uma imagem segmentada por *depth-adaptive superpixels* a partir de informações de cor e profundidade advindas do *Kinect V2*. A realização deste algoritmo se dá por meio da detecção de cores de pele, criada a partir de uma paleta de cores de pele, e a realização de operações morfológicas na imagem binária criada a partir da detecção de cores de pele na imagem segmentada por *depth-adaptive superpixels* para detecção facial. Aplicando o algoritmo implementado em um vídeo frame a frame, se torna possível a detecção de faces na imagem de cor provida pelo *Kinect V2*, utilizando informações de profundidade advindas do mesmo dispositivo.**

I. Introdução

Não raramente a primeira e mais importante etapa para o funcionamento de aplicações de vigilância, biometria facial (voltada tanto para segurança de informação quanto para segurança social), interação humano computador, manuseio de bancos de imagens ou simplesmente o principal processo para determinar-se a presença ou ausência de rostos em uma imagem, entre outros usos, a detecção de faces mostra-se uma área bastante ativa com relação à pesquisas na comunidade cientifica da visão computacional.

Como colocado por um dos principais especialistas biométricos do mundo, Dr. Robert Frischholz, em sua página web [3]: "O reconhecimento de rostos humanos não é tanto sobre o reconhecimento facial afinal - é muito mais sobre a detecção facial! Está provado que o primeiro passo no reconhecimento facial automático - a detecção precisa de rostos humanos em cenas arbitrárias, é o processo mais importante envolvido. Quando os rostos podem ser localizados exatamente em qualquer cena, o passo de reconhecimento seguinte não é tão mais complicado."

Dito isso, trata-se de um problema com possibilidades de resolução por meio das mais diversas abordagens: em ambientes com fundo controlado, por meio do uso de cores de pele típicas, em função de movimentos (como um piscar de olhos [7]), além de combinações destas e entre outras. Dessa forma, o artigo em questão desenvolve-se ao redor de uma nova abordagem com relação ao problema de detecção de face em vídeo a qual envolve o uso de dados fornecidos pelo *Microsoft Kinect Sensor* (capaz de capturar informações de cor e profundidade num ambiente), Supervoxels (grupos de voxels - equivalentes do pixel em três dimensões - com características semlehantes: valores de cor próximos, por exemplo) e uma paleta de cores de pele.

II. Conceitos Teóricos

1) Componentes Conectados: Para o entendimento do projeto, alguns relacionamentos básicos entre *pixels* devem ser explicitados, o que será feito no próximo parágrafo, sendo estes o conceito de vizinhança, conectividade, região e contorno.

Um *pixel* p com coordenadas (x, y) possui quatro vizinhos horizontais e verticais, cujas coordenadas são $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$. Este conjunto de *pixels* são denominados *vizinhança-de-4* de p, sendo representado pela notação $N_4(p)$. Cada *pixel* está a uma unidade de distância de p. Os pares ordenados dos quatro vizinhos diagonais de p são $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$ e são denotados por $N_D(p)$. A junção dos pontos das vizinhanças $N_4(p)$ e $N_D(p)$ é chamada de *vizinhança-de-8* de p. Nem todos os vizinhos permanecem dentro da imagem quando p se encontra na borda da mesma.

A conectividade entre *pixels* é um conceito relevante, utilizado no estabelecimento dos componentes conectados e também da borda de um objeto. Para determinar se dois *pixels* estão conectados, deve-se determinar alguma forma de adjacência entre eles, como estes serem vizinhos-de-4 e seus níveis de cinza iguais.

Seja V, o conjunto de valores de níveis de cinza utilizados para definir a conectividade, consideramos três tipos de conectividade:

a) *conectividade-de-4*: Dois *pixels* p e q, assumindo níveis de cinza em V, são conectados-de-4 se q está na *vizinhança-de-4* de p;
b) *conectividade-de-8* Dois *pixels* p e q, assumindo níveis de cinza em V, são conectados-de-8 se q está na *vizinhança-de-8* de p;
c) *conectividade-de-m*: Dois *pixels* p e q, assumindo níveis de cinza em V, são conectados-de-m se:
   (i) q está na *vizinhança-de-4* de p, ou
   (ii) q está em $N_4(p)$ e o conjunto $N_4(p)$ $N_4(q)$ for vazio.

Se p e q pertencerem a um subconjunto S de uma imagem, p estará conectado a q se existir um caminho entre eles constituído apenas de *pixels* pertencentes a S. Para qualquer *pixel* em S, o conjunto de *pixels* em S que estão conectados a este *pixel* é denominado *componente conectado* de S. Sendo assim, quaisquer par de *pixels* de um componente conectado estão conectados entre si, sendo que os componentes conectados distintos são disjuntos.

Uma região R é um *componente conectado*, e finalmente, um contorno C de uma região R é composto por todos os

pixels que possuem vizinhos não pertencentes a esta região R.

2) Morfologia Matemática e Segmentação: Com a finalidade de facilitar o entendimento das soluções desenvolvidas, um resumo sobre os conceitos de operação morfológica e das principais operações morfológicas (*Dilatação* e *erosão*) e de elemento estruturante, será apresentado nos parágrafos a seguir.

Uma operação morfológica consiste essencialmente da comparação da imagem com outra menor, cuja geometria é conhecida, denominada *elemento estruturante*.

Um elemento estruturante planar é um conjunto de coordenadas de pixel. Uma transformação morfológica requer uma operação não-linear entre a imagem e o elemento estruturante, o qual desliza sobre a imagem de forma similar à convolução discreta.

Um elemento estruturante não-planar é um par (E, V) que consiste de um conjunto de coordenadas de pixel E e um conjunto de valores V associados a cada coordenada, assim como uma imagem. Este tipo de elemento é usado apenas em operações com imagens em tons de cinza (O que não é abordado neste trabalho). Neste caso, o elemento estruturante pode ser visto como uma máscara de convolução, muito embora a operação seja outra. No caso particular, onde todos valores em V são zero, o elemento estruturante se torna planar.

Depois da explicitação do conceito de operação morfológica e elemento estruturante, as principais operações morfológicas serão apresentadas logo após estas definições básicas, sejam A e B conjuntos de Z^2, com componentes $a = (a_1, a_2)$ e $b = (b_1, b_2)$, respectivamente:

a) A *translação* de A por $z = (z_1, z_2)$, representada por $(A)_z$, é definida por:

$$(A)_z = \{c | c = a + z\}, \forall a \in A.$$

b) A *reflexão* de B, representada por $\hat{B}$, é definida por:

$$\hat{B} = \{z | z = -b\}, \forall b \in B.$$

c) O *complemento* do conjunto A é definido por:

$$A^c = \{z | z \notin A\}.$$

d) E finalmente, a *diferença* entre dois conjuntos A e B, denotada por $A - B$, é definida por:

$$A - B = \{z | z \in A, z \notin B\} = A \cap B^c.$$

A *dilatação* é definida por, tomando A e B como conjuntos de Z^2 e $\emptyset$ como o conjunto vazio:

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

Sendo assim, a dilatação começa na obtenção da reflexão de B em relação à sua origem, seguido da translação dessa reflexão por z, portanto, a dilatação de A por B é o conjunto de deslocamentos z tais que B e A se sobreponham em pelo menos uma elemento não-nulo.

A *erosão* é definida por, tomando A e B como conjuntos de Z^2:

$$A \ominus B = \{z | B_z \subseteq A\}$$

Sendo assim, a erosão de A por B é o conjunto de todos os pontos z tais que B, quando transladado por z, fique contido em A.

Praticamente todas as outras operações morfológicas utilizam a dilatação e a erosão em seus métodos, como por exemplo as operações de *abertura* e *fechamento*.

III. Modelo (Descrição detalhada do problema (matemática ou algorítmica))

De um ponto de vista algorítmico, o Modelo do problema em questão envolve desde a obtenção das filmagens até o cálculo da(s) posição(ões) da(s) face(s):

- Obtenção, por meio do Kinect V2, das filmagens contendo informações de cor e profundidade e separação em frames individuais, cada qual com seu par cor-profundidade;
- Pré-processamento dos pares nos frames individuas caso sejam necessárias adaptações (ajuste da resolução do par cor e/ou profundidade, como visto na próxima secção);
- Segmentação de cada frame individual a partir das informações de cor e profundidade pré-processadas;
- Detecção de pele em cada frame individual baseada em uma paleta de cores de pele definida previamente;
- Detecção de rosto(s) a partir das infomações relativas à segmentação e detecção de pele em um certo frame;

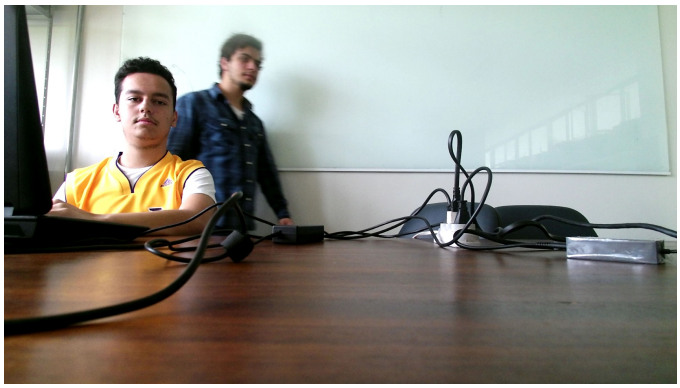

Figura 1. Imagem de Cor obtida a partir do *Kinect V2*.

IV. Solução e Análise

A Solução foi desenvolvida na plataforma *MATLAB R2017a* no sistema operacional *Windows 10* e possui 6 arquivos, sendo:

- 1 arquivo para redimensionamento das imagens de cor e profundidade para maior precisão do algoritmo *RedimensionaCorProfundidade.m*;
- 1 arquivo que realiza o modelo descrito na sessão anterior *Principal.m*, que possui 4 funções:
  - *Paleta.m*;
  - *DeteccaoPele.m*;
  - *DeteccaoRosto.m*;
  - *Resultado.m*.

Vários métodos foram utilizados para ser possível chegar a um resultado final. Os seguintes métodos, serão listados a seguir.

Figura 2. Imagem de Profundidade obtida a partir do *Kinect V2*.

Figura 3. Densidade de pixels DASP.

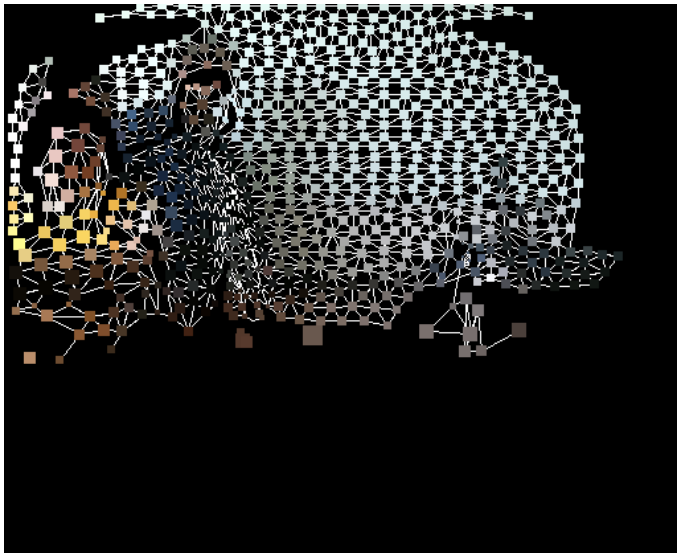

Figura 4. Grafo DASP.

A. Extração Frames de Cor e Profundidade a partir do Kinect V2

A obtenção dos frames de cor e profundidade de um vídeo foi feita a partir de um *Kinect V2*, utilizando a aplicação *KinectStudio*. O *KinectStudio* bloqueia a extensão do vídeo obtido, permitindo a execução do vídeo apenas nele mesmo, portanto, foi utilizada a ferramenta (https://github.com/LuciaXu/Xef2Mat-Jpg) para extrair os dados de cor e profundidade frame a frame do vídeo gravado. Sendo assim o processamento do vídeo foi implementado frame a frame, devido ao bloqueio de execução por parte da aplicação *KinectStudio*.

B. Redimensionamento Imagens de Cor e Profundidade

O *Kinect V2* captura as informações de cor e profundidade de maneira diferente, pois para a captura de cor uma câmera *Full HD* é utilizada, enquanto a captura de profundidade é realizada por mais de uma câmera. Devido à esta diferença na captura, as imagens de cor e profundidade apresentam diferentes dimensões e escalas, fazendo com que a segmentação *DASP* cause algumas falhas perceptíveis, porém que não atrapalharam o resultado do projeto no contexto testado. O redimensionamento de tais imagens é implementado pelo código *RedimensionaCorProfundidade.m*. Para realizar o redimensionamento das imagens, deve-se alterar as imagens de entrada dentro do código *RedimensionaCorProfundidade.m*.

C. Segmentação DASP

A segmentação *Depth-Adaptive SuperPixels* foi realizada com a utilização do código do autor (https://github.com/Danvil/asp), no sistema operacional *Ubuntu 14.04 LTS* para a obtenção das imagens segmentadas pelo algoritmo *DASP*. As saídas obtidas por meio da execução do algoritmo são ilustradas nas Figuras (3 até 7), sendo que a única saída que foi utilizada para a realização do projeto foi a apresentada na Figura 7, pois os autores deste projeto julgaram a mesma como a imagem com maior informação para processamento.

D. Definição de uma Paleta de Cores de Pele

A definição do tom de pele a ser detectado pelo algoritmo é feita a partir de uma paleta de cores selecionada a partir de 10 imagens de pessoas, construindo uma única imagem a partir de recortes de regiões de cor de pele. Esta paleta de cores é passada da escala de cores *RGB* para a escala *YCbCr*, e após isso calcula-se a média e o desvio padrão das camadas *Cb* e *Cr* para definir o intervalo a ser considerado como cor de pele nas imagens a serem analisadas.

E. Detecção de Pele

O processo de detecção de pele nas imagens consiste em mudar a escala de cores da imagem do *RGB* para a escala *YCbCr*, e verificar se os *pixels* das camadas *Cb* e *Cr* estão no intervalo estabelecido como cor de pele, se estes estiverem, o *pixel* correspondente na imagem binarizada recebe o valor 1,

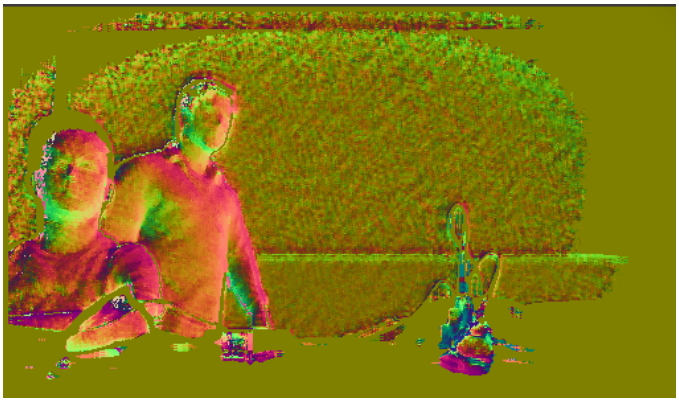

Figura 5. Normais de pixels DASP.

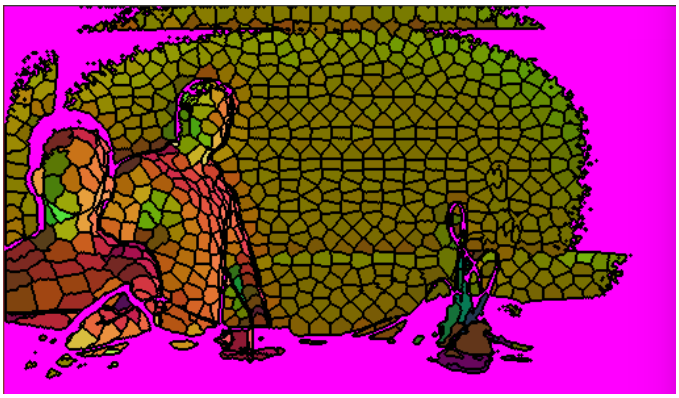

Figura 6. Normais de pixels segmentadas em superpixels DASP.

caso contrário, este *pixel* recebe o valor 0. Portanto a cor de pele na imagem binarizada é representada pelo 1, e o que não é caracterizado como cor de pele recebe 0.

F. Detecção de Rosto

A detecção de rosto é realizada a partir de um conjunto de métodos que serão explanados a seguir:

1) Operações Morfológicas: Primeiro, são realizadas operações morfológicas para a facilitação da detecção de formas que se assemelham a faces na imagem binarizada, para isso, a primeira operação morfológica realizada é a erosão, tomando como elemento estruturante um quadrado de tamanho 2 (Obtido de maneira empírica), o resultado desta erosão em um frame de exemplo é ilustrado na Figura 11.

A partir do resultado obtido com a realização da erosão, é realizada a operação morfológica de dilatação, tomando como elemento estruturante também um quadrado, porém de tamanho 10 vezes maior que o elemento estruturante citado anteriormente. O resultado desta operação sobre o frame de exemplo pode ser visualizado na Figura 12.

Pode-se observar que o resultado obtido com a realização destas operações morfológicas é satisfatório, pois os componentes conectados restantes se assemelham à forma elíptica de um rosto humano.

2) Deteccção de Bordas: A partir da imagem resultante representada na Figura 12, para se obter um efeito melhor na imagem final, a aplicação de um detector de bordas é realizada.

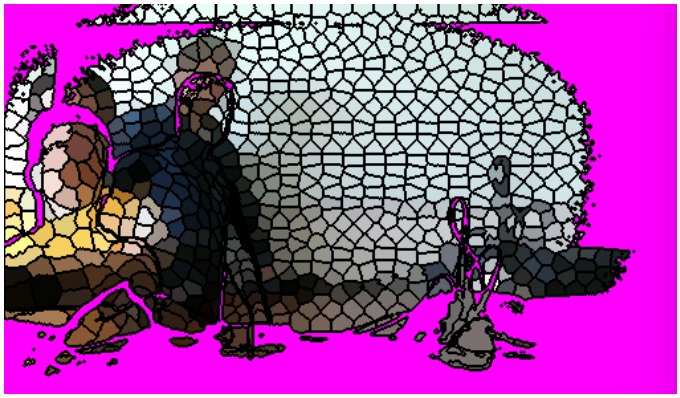

Figura 7. Segmentação DASP.

Figura 8. Paleta de cores construída.

O detector de bordas utilizado é o *Canny*. Desenvolvido por John F. Canny em 1986, o detector de bordas de Canny utiliza um algoritmo multi-estágios para detectar uma ampla margem de bordas na imagem. John Canny propôs que o detector de bordas ótimo deveria respeitar os seguintes parâmetros:

- **Boa Detecção** - O algoritmo deve ser capaz de identificar todas as bordas possíveis na imagem.
- **Boa Localização** - As bordas encontradas devem estar o mais próximo possível das bordas da imagem original.
- **Resposta Mínima** - Cada borda da imagem deve ser marcada apenas uma vez. O ruído da imagem não deve criar falsas bordas.

Para satisfazer tais condições, Canny utilizou um cálculo de variações, visando encontrar uma função que otimizasse o funcional desejado. A função ideal para o detector de Canny é descrito pela soma de quatro termos de exponenciais, que pode ser aproximada pela primeira derivada de uma gaussiana [8].

3) Deteccção de Faces: A detecção de face foi realizada a partir da imagem com as bordas detectadas, representada na Figura 13. Para realizar a filtragem de quais bordas dos componentes conectados possuíam maior relevância e semelhança com o formato de uma cabeça humana, foram utilizadas duas funções:

- *bwareaopen*;
- *bwpropfilt*.

A função *bwareaopen* retira os componentes conectados de área menor que a especificada, o tamanho de 20 foi utilizado, para a retirada de ruídos da imagem de acordo com o alcance máximo do dispositivo *Kinect V2*.

A função *bwpropfilt* obtém os componentes conectados de acordo com uma característica específica. A característica selecionada foi a *excentricidade* que é um parâmetro associado a qualquer cônica, que mede o seu desvio em relação a uma circunferência, ou seja, quanto maior a excentricidade, mais próxima de uma reta a forma se torna, e quanto menor, mais próximo de um círculo. Como a distorção da segmentação

Figura 9. Imagem binária advinda da detecção de pele realizada na imagem segmentada pelo algoritmo *DASP*.

Figura 10. Representação da detecção de pele realizada na imagem segmentada pelo algoritmo *DASP* a partir da binarização na imagem segmentada original.

DASP deixa o rosto humano mais distante de um círculo perfeito, os componentes conectados com maior excentricidade foram selecionados para a implementação.

V. Resultados

A Figuras 15 até 20 apresentam os resultados da aplicação do algoritmo implementado em alguns frames do vídeo obtido por meio do *Kinect V2*.

Figura 11. Resultado da erosão.

Figura 12. Resultado da dilatação.

VI. Conclusões

O modelo proposto para detecção de faces com o uso de Supervoxels e dados Kinect V2 apresentou resultados satisfatórios (principalmente para faces estáticas e direcionadas à câmera) para seu propósito, levando em consideração seu nível de simplicidade e que sua principal fonte de falhas reside nas limitações do equipamento de captura de imagens utilizado

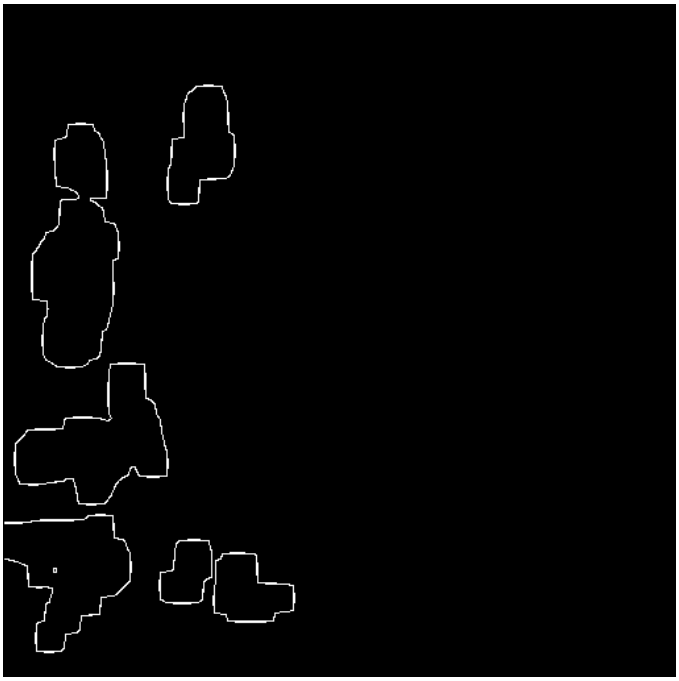

Figura 13. Bordas obtidas a partir do resultado da erosão.

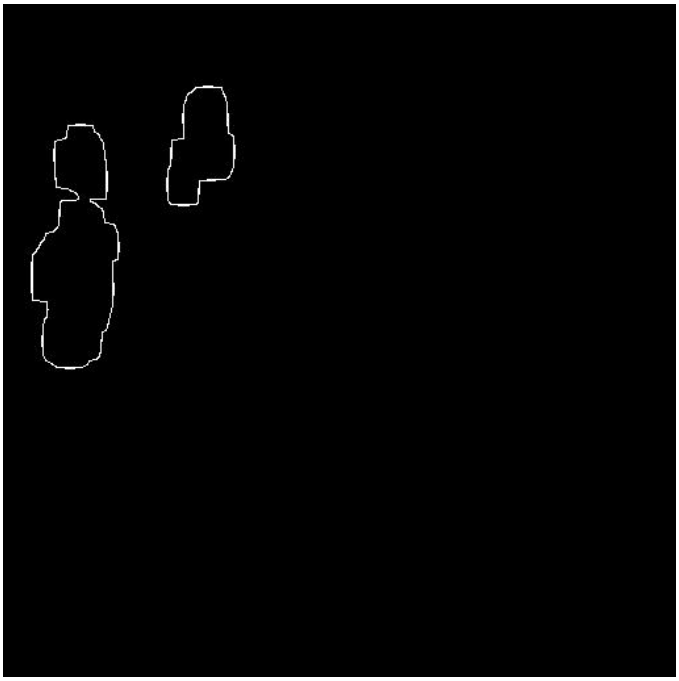

Figura 14. Resultado final obtido pelo algoritmo desenvolvido.

(resoluções e proporções das imagens de cor e profundidade diferentes). Em trabalhos futuros, cogitaria-se explorar outras formas para obtenção de imagens com informação de profundidade, realizar todo o procedimento em um único sistema operacional, aprimoramento de performance e a realização de teste mais variados.

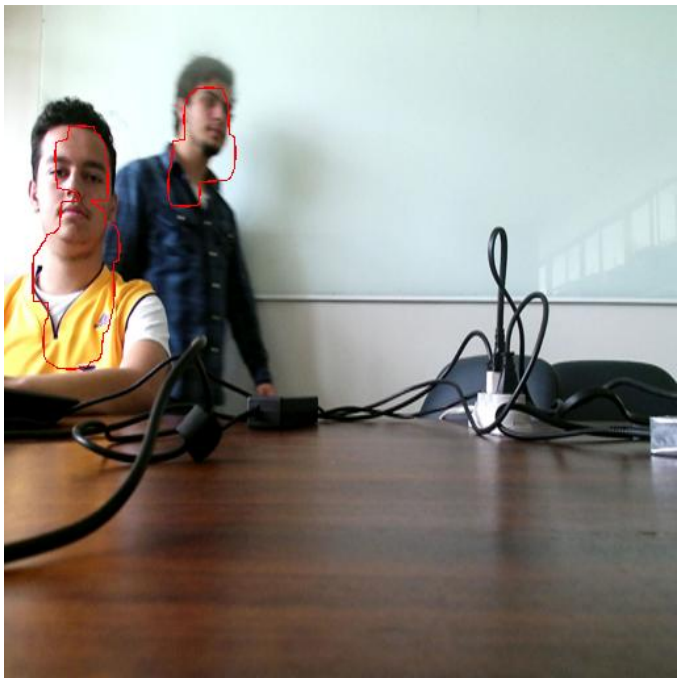

Figura 15. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

Figura 16. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

Figura 17. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

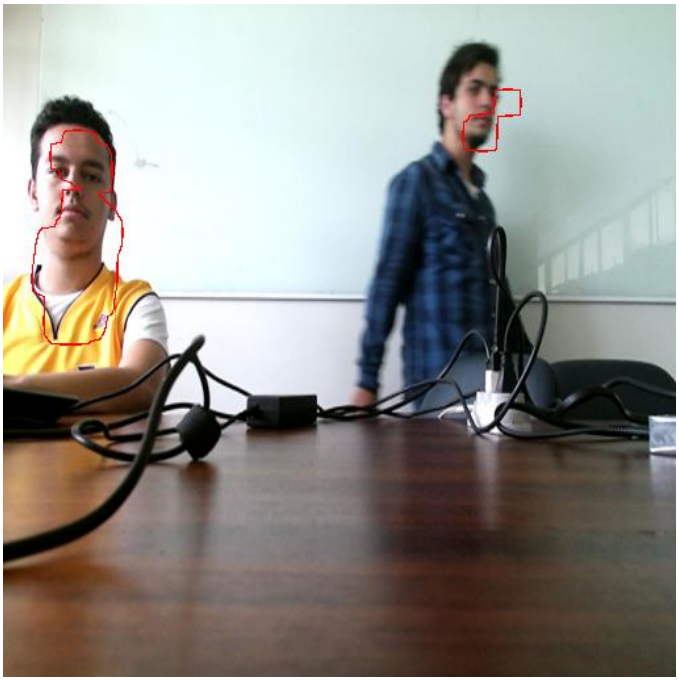

Figura 18. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

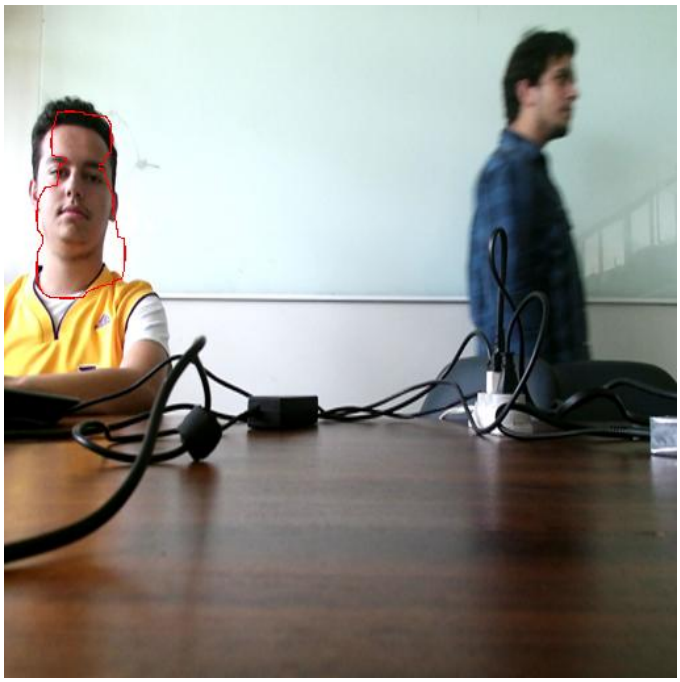

Figura 19. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

Figura 20. Detecção de face no dado frame do vídeo obtido a partir do *Kinect V2*.

REFERÊNCIAS

[1] David Weikersdorfer, Alexander Schick e Daniel Cremers, **DEPTH-ADAPTIVE SUPERVOXELS FOR RGB-D VIDEO SEGMENTATION**.

[2] Gonzalez, Rafael C. e Woods, Richard E.,**Processamento de imagens digitais**, 1a ed. São Paulo, Brasil: Editora Edgard Blücher Ltda., 2000, ISBN 85-212-0264-4.

[3] Frischholz, R. **The Face Detection Homepage**.

[4] Moraes, R. **Perceptor**. Brasil, 2011. Disponível: http://www.blogpercepto.com/2011/02/sistemas-computacionais-de.html

[5] Satone, M.P. Kharate,G.K. **Face detection and recognition in color images**. India, 2011. Disponível: https://www.ijcsi.org/papers/IJCSI-8-2-467-471.pdf

[6] BioID. **Our management team**. 2017. Disponível: https://www.bioid.com/About/Team

[7] Reignier, P. **Finding a face by blink detection**. 1995. Disponível: http://www-prima.imag.fr/ECVNet/IRS95/node13.html

[8] Reignier, P. **Detector de bordas de Canny**. 1995. Disponível: https://pt.wikipedia.org/wiki/Detector_de_bordas_de_Canny