PM

Interações Gestuais e Faciais Detetadas com Câmara de Profundidade

PROJETO DE MESTRADO

David Gomes de Aguiar
MESTRADO EM ENGENHARIA INFORMÁTICA

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

setembro | 2018

Interações Gestuais e Faciais Detetadas com Câmara de Profundidade

PROJETO DE MESTRADO

David Gomes de Aguiar
MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR
Diogo Nuno Crespo Ribeiro Cabral

Agradecimentos

Eu gostaria de começar agradecendo os meus pais, João Belchior e a minha mãe Maria Salete Aguiar e meu irmão Bruno Aguiar, pelo todo o apoio e o incentivo. Eu estou eternamente grato pela oportunidade que me deram para perseguir e continuar a minha educação até hoje.

Eu também agradeço ao professor Diogo Cabral, o meu supervisor. Sem a sua orientação e apoio, este projeto nunca estaria concluído.

Agradeço também a todos os meus colegas, que me acompanharam todo este percurso, dando o seu apoio.

Este trabalho foi parcialmente financiado pelo Madeira-ITI, FCT/MCTES LARSyS (UID/EEA/50009/2013 (2015-2017)).

Obrigado!

Resumo

Vários métodos de interação foram criados ao longo do tempo, na tentativa de substituir os métodos tradicionais (teclado e rato) por outros mais naturais. Porém estas alternativas acabam por ser na maioria das vezes dispendiosas e não se adaptarem a diferentes cenários de aplicação. As câmaras de profundidade demonstram ser uma alternativa de interação com menos limitações, mas cujos cenários de aplicação em que esta tecnologia seja uma mais valia real ainda se encontram por definir. Neste projeto foi desenvolvido um protótipo que permite a manipulação de imagens através de interações faciais e gestuais e uma frame háptica de apoio ao utilizador e que explora a profundidade dos gestos como meio de seleção. Estas interações têm como base a captação e seguimento de múltiplos pontos da face e pontos das mãos.

Palavras-chaves:

Câmara de Profundidade, Interações Gestuais, Interações Faciais, Mesas Multi Toque.

Abstract

Several interaction methods have been used, aiming to replace traditional methods (keyboard and mouse) with more natural ones. However, these alternatives turn out to be most of the time costly and do not adapt to different application scenarios. Depth cameras can be used as interaction alternative with less limitations but whose application scenarios in which this technology is a real added value are still to be defined. In this project a prototype was developed, allowing the manipulation of images through facial and gestural interactions, as well as a haptic frame that supports the user and exploits the depth of the gestures as a way of selection. These interactions are based on capturing and tracking multiple points on the face and hand points.

Keywords:
Depth camera, Gestural Interactions, Facial Interactions, Multitouch Table

Índice

Índice de Figuras

1

Introdução

1. Introdução

Ao longo dos anos, foram criadas alternativas ao rato e ao teclado, para interagir com sistemas digitais. Os métodos tradicionais envolvem a necessidade do utilizador aprender a utilizar tecnologias e dispositivos em atividades que não são habitualmente aplicados, por exemplo utilizar o rato para desenhar, tornando a tarefa mais difícil comparada se utilizasse uma caneta (Tscheligi et al., 1995). Porém estes métodos alternativos, normalmente envolvem múltiplas câmaras ou sensores e ecrãs de elevadas dimensões (Ruotsalo et al., 2016) (Andolina et al., 2015), capturando assim o máximo de informações fornecidas pelos utilizadores. Isto pode gerar um sistema com vários níveis de complexidade, o que o torna demasiado limitado e dispendioso, e acabando por não se adaptar totalmente a diferentes cenários de aplicação. Por alternativa a estas abordagens surgem as câmaras de profundidade.

Nas câmaras de profundidade, sensor de profundidade é composto por dois elementos: emissor de infravermelho (IR) e por câmara de infravermelho. O emissor infravermelho emite um padrão de luz infravermelho que chega sobre os objetos em redor, como um conjunto de pontos não visíveis ao olho humano, mas reconhecidos pela câmara. A câmara de infravermelhos é essencialmente o mesmo que uma câmara de cor, exceto que as imagens capturadas são na gama das ondas infravermelhas. A câmara de infravermelho envia o vídeo com padrão de pontos distorcidos para o processador do sensor de profundidade, e este calcula a profundidade a partir do deslocamento dos pontos (McWilliams, 2013). Estas câmaras surgem como solução de facilitar a captação e os movimentos do utilizador e da possibilidade aos utilizadores a opção de modificar as operações fornecidas pela câmara. Com tecnologia das câmaras profundidade, pretende-se criar um cenário de trabalho, que seja fácil de montagem e o com as mesmas funcionalidades dos meios convencionais.

1.1 Objetivos

O principal foco deste projeto, reside na criação de um cenário que utiliza uma câmara de profundidade Creative Blasterx Senz3D/Intel Real Sense D300, para

realizar as mesmas ações do que os meios convencionais (rato e teclado) e possibilitando ao utilizador ações mais naturais, dado o uso de gestos, expressões e movimentos da face. A outra parte do cenário, consiste em dar ao utilizador de alguma forma de feedback háptico, melhorando assim a sua experiência, reduzindo os erros e cansaço devido às várias horas de utilização (Ott et al., 2005) (Varcholik et al., 2012).

1.2 Contribuições

As contribuições neste projeto foram:

1. Estudo da framework da Intel Realsense Gold R3.
2. Captação de múltiplos pontos da face e pontos das mãos.
3. Movimento do rato através do movimento da mão da direita.
4. Reconhecimento de emoções e gestos.
5. Interceção entre objetos e os movimentos do utilizador.
6. Cenário de Utilização e frame háptica para interação com gestos e faces.

1.3 Organização do Documento

O capítulo 2 apresenta as principais diferenças entre as três câmaras de profundidade da Kinect, Leapmotion e Creative Blaster Senz3D. Neste capítulo também se apresenta o software, a arquitetura e algoritmos presentes na Creative Blaster Senz3D, como também se faz referência a trabalhos relacionados.

O capítulo 3 corresponde a implementação do protótipo. Este capítulo apresenta as várias etapas e as decisões para a construção do protótipo. O capítulo 4 foca-se nos testes realizados, dos módulos e limitações da câmara.

Por fim, e de modo a concluir o trabalho, o capítulo 5 são apresentadas as contribuições e são definidos os objetivos para trabalho futuro.

2

Contexto e Trabalho Relacionado

2. Contexto e Trabalho Relacionado

Este capítulo descreve as principais câmaras de profundidade que estão no mercado, nomeadamente Kinect, Leap Motion e a Creative Blasterx Senz3D/Intel Realsense. Descrevemos as principais funcionalidades da Creative e também nas principais caraterísticas, funcionalidades da framework utilizada pela câmara Creative Blasterx Senz3D, que são usadas para este projeto. Em seguida, é feito uma análise aos mecanismos da deteção da trajetória das mãos, e em simultâneo é apresentado uma investigação relativa ao trabalho relacionado. Depois é feito uma análise das diferenças dos modos de captação das mãos e o algoritmo de suavização. Por fim são descritas diferentes implementações de mesas multitoque que usam câmaras de profundidade de modo a enriquecer a interação com o utilizador criando assim cenários de aplicação para este tipo de câmaras.

2.1 Câmaras de Profundidade

O Leap Motion (“Leap Motion,” 2013.), Kinect (“Kinect” , 2011.) e Creative Blasterx Senz3D (“BlasterX Senz3”, 2016.) são câmaras de profundade que fornecem novas maneiras de interagir com o computador. Estas permitem aos utilizadores interagir, por meios sem fios, com o computador movendo o seu corpo no espaço físico embora existem diferenças significativas entre eles e no seu funcionamento interno.

A Kinect v1 (figura 1) foi desenvolvido pela Microsoft para o uso da consola Xbox 360 e foi lançada no início de novembro de 2010. Foi desenhada para ser colocada em frente da televisão, apontando para o utilizador. A API da Microsoft permite os criadores aceder informação sobre a posição e velocidade das diferentes partes do corpo. A Kinect foi o primeiro 3D dispositivo de captação de movimento (motion-tracking device) com o preço relativamente mais baixo, e devido a isso a comunidade cresceu em volta. Estas pessoas desenvolveram bibliotecas alternativas para comunicar com a Kinect, bem como programas que usam o Kinect de formas inesperadas (Zhang, 2012). A Kinect já foi usada para mapear cavernas (Mann, 2011), digitalizar objetos em 3D (Talldrinks, 2011.) e mapear mapas para o uso de robôs(Roth and Vona, 2012).

FIGURA 1 - PRIMEIRA VERSÃO DA KINECT[1]

A Kinect funciona por projetar uma grelha de pontos por infravermelhos e determina a distância de cada ponto, medindo o tempo que leva a luz bater numa superfície e retornar ao sensor.

A segunda versão a Kinect (figura 2), é uma evolução a nível de hardware bem como o software. Nesta versão da Kinect, a deteção do esqueleto (skeletal tracker) supera a versão anterior. Este reconhece mais pessoas, mais articulações (26 articulações), em menos tempo e com mais precisão. Outra diferença entre as duas versões é o cálculo da profundidade, a Kinect v1 utiliza o cálculo da profundidade usando projeção do infravermelho, enquanto a Kinect v2 utiliza o tempo de voo (Time-of-flight). Isto significa que a Kinect v2 emite luz aos objetos e calcula o tempo de que demora a chegar a câmara. É um método mais estável comparado ao método da Kinect v1(Wasenmüller and Stricker, 2017).

FIGURA 2 - SEGUNDA VERSÃO DA KINECT[2]

[1] https://pt.wikipedia.org/wiki/Ficheiro:Xbox-360-Kinect-Standalone.png, 21/01/2019
[2] https://pt.wikipedia.org/wiki/Ficheiro:Xbox-One-Kinect.jpg, 21/01/2019

O Leap Motion (figura 3) é outro dispositivo 3D, embora seja diferente da Kinect. O Leap Motion foi anunciado em 2010, embora foi só lançado ao mercado em 2013.

FIGURA 3 - LEAP MOTION[3]

O controlador do Leap Motion consiste em duas câmaras duas câmaras de infravermelho e três projetores de infravermelhos. As faixas de luz infravermelhos dos projetores imitem comprimentos de ondas de 850 nanómetros, que se encontram fora do espetro da luz visível. Com as lentes de forma angular, o Leap Motion tem um espaço de interação, aproximado 8 cm^3, assim tomando a forma de uma pirâmide invertida. Com o uso do software Orion, a distância de visualização chega aos 80 cm. O alcance é limitado devido à propagação da luz do díodo emissor de luz (LED) pelo espaço, o que torna difícil inferir a posição da mão num espaço em 3D. A intensidade do LED é limitada pela corrente extraída pela convecção USB(Han and Gold, 2014).

A Creative Blasterx Senz3D (figura 4), tal como Leap Motion, foi também desenhada, para o uso próximo do utilizador, contudo este equipamento também capta a face, gestos e emoções reconhecido através da framework da Intel, denominada por Intel Realsense (Modelo D300), devidamente incorporada dentro deste modelo de câmara da Creative. Esta framework que dá possibilidade aos utilizadores de manipular os dados adquiridos pela câmara, da face e das mãos, e utilização de algoritmos que melhoram esses dados, tal com suavização dos dados

[3] https://commons.wikimedia.org/wiki/File:Leap_Motion_Orion_Controller_Plugged.jpg, 21/1/2019

(Smother) e outras funcionalidades que vai ser falado mais para frente (Pham et al., 2015).

A Creative Blasterx Senz3D disponibiliza as seguintes funcionalidades:

- Controlo de Gestos;
- Deteção do rosto 3D;
- Controlo de voz;
- Scan 3D;
- Remoção de imagem de fundo;
- Resolução de 720p60/1080p30;

As principais diferenças técnicas entre os sistemas da Kinect, o Leap Motion e Creative Blasterx Senz3D está resumido na Tabela 1.

	Kinect	Leap Motion	Creative Blasterx Senz3D/Realsense
Fabricante	Microsoft	Leap Motion Inc	Creative/Intel
Alcance	1.2 - 4.5 m	0.002 – 0.61 m	0.2 – 1.5 m
Reconhecimento	Corpo, Face e Voz	Mãos e Dedos	Face, Mãos, Dedos, Gestos, Emoções e Voz
Sensores	1 transmissor infravermelhos, 1 câmara de infravermelhos de 0.3 megapixéis, 1 câmara de RGB e 4 microfones direcionais	3 transmissores infravermelhos, 2 câmaras de infravermelhos de 1.3 megapixéis	1 câmara de infravermelhos, 1 câmara de alta definição em cores, 1 transmissores infravermelhos e 2 Microfones em array
Configurações	SDK para Windows para Microsoft	AirSpace Home	Intel RealSense

Tabela 1- Diferenças técnicas entre Kinect, Leap Motion e Creative Blasterx Senz3D

Se por um lado a Kinect reconhece diferentes posições de corporais, mas não consegue distinguir gestos de maior detalhe como os das mãos e dedos, a Leap Motion limita-se a detetar gestos dos dedos. Assim a Creative Blasterx Senz3D/Intel Real D300 apresenta-se como uma alternativa que permite o reconhecimento de elementos maiores como mãos e faces como também dos seus elementos (por exemplo, dedos, olhos, nariz, boca, entre e outros). Como tal, esta foi a câmara utilizada neste projeto.

2.2 Creative Blasterx Senz3D: Hardware e Software

Na tabela 2 encontram-se a especificações da câmara Creative Blasterx Senz3D, com estes dados é possível verificar quais são as limitações desta. Com esta informação é possível planear a melhor forma da construção do projeto.

Tabela 2 - Importantes especificações da Câmara Creative Blasterx Senz3D

Resolução RGB	Full HD 1080p (1920x1080)
Resolução do sensor infravermelho	VGA (640x480)
RGB Taxa de quadro (Frame rate)	60 fps @ 720p, 30 fps@1080p
Infravermelho Taxa de quadro	60 fps @ 680x480
FOV (campo de visão)	77º (RBG), 85º (IR Depth)
Alcance	0,2m ~ 1,5m
Resposta da Frequência	20Hz – 20kHz

A Creative Blasterx Senz3D usa a framework da Intel Realsense. A Intel Realsense é uma biblioteca de deteção de padrões e implementação de algoritmos de reconhecimento expostas através de interfaces padronizadas. A Intel Realsense contém um sensor de RGB (Vermelho, verde e Azul), um emissor de luz infravermelho, uma câmara de infravermelho e múltiplos microfones. O emissor infravermelho e a câmara de infravermelho funcionam em conjunto, projetando inicialmente uma grelha na imagem e gravando-a e calculando a informação de profundidade. Os microfones permitem localizar o ponto de origem do som no espaço e o cancelamento do barulho de fundo. Atualmente existem três modelos, com distintas especificações, pela Intel(“Intel® RealSense Technology | Intel®

Software," 2015.). Neste projeto é usado o modelo SR300 da Intel RealSense, que é uma câmara frontal dedicada a analisar o rosto e a mão humana. Juntamente com a câmara, é necessário o ambiente de desenvolvimento, Visual Studio, da Microsoft e o Kit de Desenvolvimento de software (SDK) da Intel Realsense, para obter os dados necessários das mãos e da face.

A Intel RealSense em conjunto com a Intel Realsense SDK é capaz de fornecer 78 pontos faciais e detetar 16 expressões faciais("Face Landmark Data," 2016.), bem como, 22 pontos em cada mão e 14 gestos ("Hand Joints," 2016.). A figura 4 mostra esses pontos importantes. Os pontos faciais e das mãos podem trabalhar em duas planos separados, o plano 2D e no plano 3D, sendo assim possível de obter mais dados dependendo no plano usado.

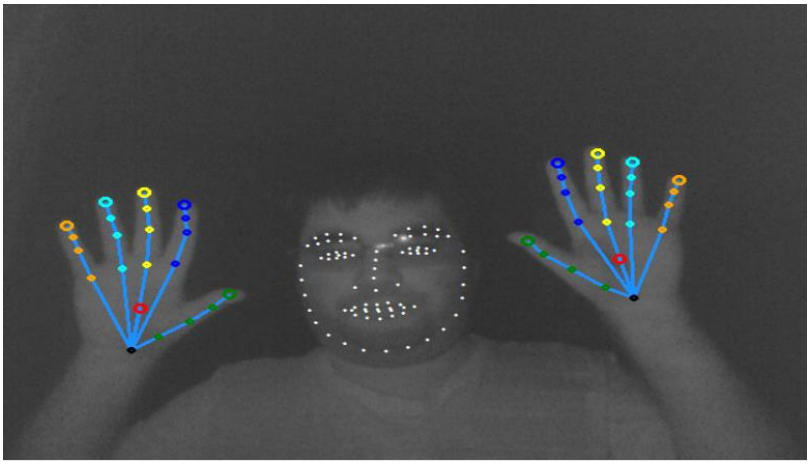

FIGURA 4 - OS 78 PONTOS FACIAIS E 22 PONTOS DAS MÃOS

Porem não são usados todos os pontos e as Ponto, na lista de seguinte estão presentes os pontos mais usados para este projeto.

- Ponto da articulação do indicador.
- Ponto da articulação do polegar.
- Gesto punho fechado.
- Ponto central do olho direito.
- Expressão boca aberta.

2.2.1 Arquitetura

A versão da framework da Intel Realsense utilizada é denominada Intel Realsense SDK 2016 R3, contudo existe uma versão mais atualizada que é chamada por Intel Realsense SDK 2.0. Na subsecção das limitações (5.1) é mais detalhado o motivo da escolha da versão da framework.

A arquitetura da biblioteca do SDK, ilustrada na figura 5, consiste em várias camadas de componentes. A essência do funcionamento do SDK está nos módulos de entrada/saída (I/O) e nos módulos dos algoritmos. Os módulos de entrada e saída recebem os dados de um dispositivo de entrada ou envia os dados para um dispositivo de saída. O módulo dos algoritmos inclui vários algoritmos de deteção e reconhecimento de padrões que são importantes para experiência inovadora humana-computador, como reconhecimento de face, reconhecimento de gestos, reconhecimento da voz e texto para fala.

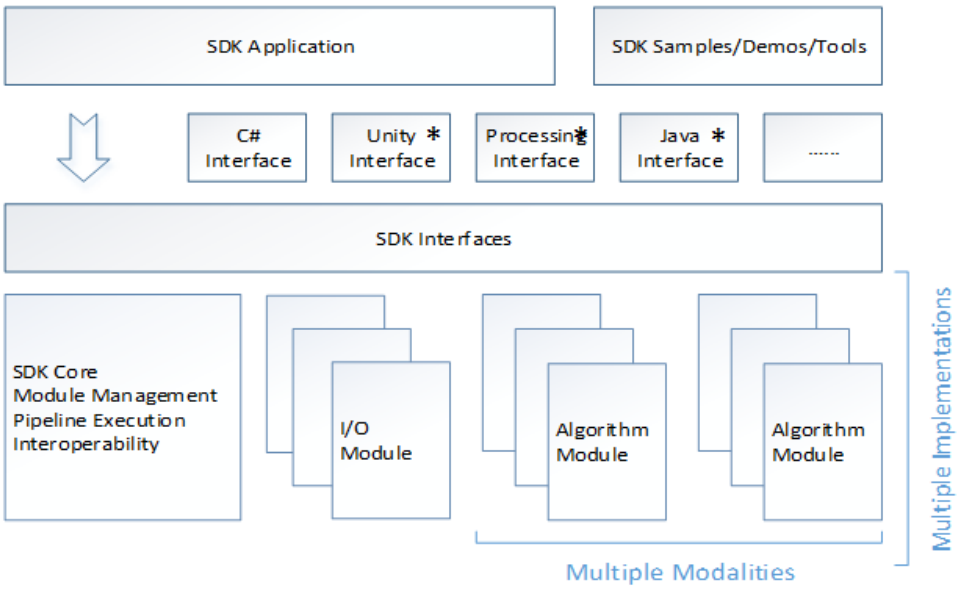

FIGURA 5 - ARQUITETURA DA INTEL REALSENSE SDK[4]

[4] https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_essential_programming_guide.html, 20/11/2018

O SDK padroniza as interfaces dos módulos de entrada e saída e nos módulos dos algoritmos para as aplicações possam aceder as funcionalidades sem se preocupar com as implementações subjacentes. Múltiplas implementações das interfaces do SDK podem coexistir. O SDK fornece o mecanismo para procurar uma implementação em especifica a partir dos módulos dos algoritmos disponíveis, bem como outros recursos, como a criação de uma instância de um algoritmo de implementação.

2.2.2 Interfaces dos Módulos da Intel Realsense SDK

A Intel Realsense SDK consiste em múltiplos módulos. A figura 6 ilustra a hierarquia da interface.

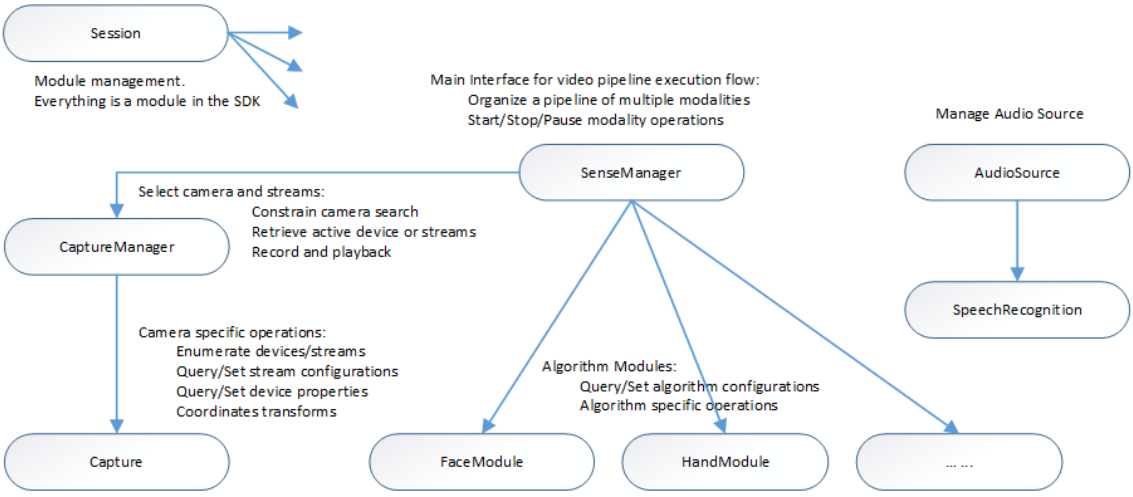

FIGURA 6 - HIERARQUIA DOS MÓDULOS DA INTEL REALSENSE SDK[5]

A interface da sessão (Session) administra os seguintes módulos: módulo da entrada/saída, módulo dos algoritmos e qualquer outra implementação de interfaces do SDK. Primeiro é necessário criar uma instância da interface da sessão na aplicação, em seguida criar outra instância do módulo a partir da instância da sessão.

[5] https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_essential_programming_guide.html, 20/11/2018

Para usos predefinidos, como reconhecimento manual e reconhecimento da face, pode ser usado a interface do SensorManager. Esta interface organiza a pipeline multimodal e controla a execução do pipeline, como iniciar, parar, pausar e recomeçar a pipeline.

Internamente, o módulo do SenseManager usa o módulo da CaptureManger para escolher o equipamento de entrada/saída e os fluxos de cor/profundidade/som. Existe a possibilidade da reutilização da interface do Capture para as operações físicas da câmara, como enumeração de equipamentos/fluxos, listar as configurações da ligação e as propriedades do equipamento.

Durante a execução do pipeline, quando algumas amostras de vídeo estiverem prontas a sair da câmara, é possível aceder a essas amostras a partir da interface de Imagem, que abstraem os buffers de imagem. Quando um módulo de um algoritmo no pipeline esta pronto com alguns resultados de processamento, é necessário obter as especificas interfaces dos algoritmos, como HandModule para o reconhecimento da mão e FaceModule para o reconhecimento da face. Estas Interfaces fornecem funções especificas dos algoritmos para definir configurações dos algoritmos e dados do algoritmo.

A ligação do áudio é um pouco diferente, onde a aplicação gere a fonte do som por meio da interface AudioSource e recursos de voz específicos diretamente na interface do modulo, por exemplo, SpeechRecognition.

2.3 Deteção de movimento com Câmaras de Profundidade

Com o avançar dos anos, o uso da deteção de movimento é cada vez mais usado. O motivo para tal acontecimento é a sua facilidade de utilização e os movimentos realizados são mais naturais/instinto do que utilizar intermediário para realizar as mesmas ações. Contudo a deteção de movimento das mãos são considerados um desafio (Sharp et al., 2015). As mãos podem formar vários conjuntos de poses com vários níveis de liberdade, têm movimentos rápidos e que podem ser de várias formas e tamanhos. Os dedos podem ser difíceis de decompor e podem estar ocultos dependo do angulo que a mão se encontra. Assim a maioria

dos projetos, limitam a distancia como é apresentado nos trabalhos de Paul Dezentje (Dezentje et al., 2015) e por Rupam Das(Das and Shivakumar, 2016), ou mesmo por uso de uma só mão como o caso dos trabalhos apresentados nos trabalho de Rui Vilaça (Vilaça et al., 2017) e por Sridhar et al(Sridhar et al., 2013).

A deteção de movimento das mãos (hand tracking), consiste em detetar a trajetória da mão numa sequência de imagens, que deteta quais são os pixéis que pertencem à mão em cada frame. A este processo dá-se o nome de segmentação das mãos (hand segmentation). Para fazer segmentação das mãos, consiste em tirar partido da informação da câmara de profundidade, assumindo que a mão esta mais próxima a câmara, logo, tudo o resto é ignorado.

A deteção e movimento da face (face tracking), trabalha na mesma nos mesmos princípios da deteção de movimento das mãos, ou seja, numa sequência de frames, o modulo só vai detetar os vários pontos da face, assim consegue retirar a informações necessários, para detetar as emoções, expressões e o ângulo da face.

Existem 3 métodos para fazer a deteção de movimento: métodos discriminatórios que trabalham diretamente com os dados da imagem (e usados na Intel Realsense); métodos generativos que utilizam um modelo 3D de uma mão ou face para recuperar a pose ou emoção; e, por fim, métodos híbridos que são uma combinação dos anteriores.

2.3.1 Métodos discriminativos

Os métodos discriminativos usam conjuntos de treino (training sets) e técnicas de aprendizagem automática para fazerem o mapeamento direto a poses através das características extraídas das imagens. Existe uma base de dados com várias imagens de poses de mãos e tentam depois fazer a correspondência com base no que é capturado pela câmara.

2.3.2 Métodos generativos

O trabalho apresentado por Oikonomidis (Oikonomidis et al., 2011), apresentam o problema da deteção de movimento das mãos, baseado em métodos generativos

e que é formulado como um problema de otimização que minimiza a discrepância entre um modelo virtual de uma mão 3D, usada como base e o que observada pela câmara de profundidade da Kinect. Por outras palavras, existe um modelo com 27 parâmetros que correspondem às articulações dos dedos de uma mão 3D e a captura da câmara de profundidade, o objetivo é estimar os valores dos parâmetros do modelo de modo a ficarem próximos da mão correspondente à realidade.

É pretendido minimizar a função que calcula a discrepância entre o observador pela câmara de profundidade e os mapas de profundidade calculados com base do modelo 3D da mão. A otimização é feita através de um algoritmo estocástico de otimização, o Particle Swarm Optimization(PSO). O princípio base do algoritmo é criar um exame de partículas que se move num espaço multidimensional predefinido à procura do seu objetivo, ou seja, a posição no espaço que melhor satisfaz as suas necessidades.

O algoritmo tem como base dois conceitos principais. O primeiro é que uma partícula pode determinar a qualidade da sua posição atual. O segundo consiste num fator estocástico que faz as partículas moverem-se. Portanto, temos um conjunto de partículas a moverem-se num espaço em que a posição de cada uma é avaliada de acordo com uma função fitness. A função de fitness depende do problema a ser otimizado. No trabalho de Oikonomidis o algoritmo PSO opera sobre um espaço de 27 dimensões que correspondem aos parâmetros das articulações da pose 3D da mão, ou seja, para cada frame, temos 27 parâmetros desenvolvidos pelo algoritmo. Contudo, este trabalho não consegue recuperar da perda do movimento.

2.3.3 Métodos híbridos

Os métodos híbridos combinam os métodos discriminativos e generativos baseados em análise por síntese em que muitas deles utilizam o algoritmo PSO. Inicialmente a componente discriminativa faz uma predição direta através da imagem de entrada sobre os parâmetros da pose da mão. Depois, a componente generativa tenta aproximar um modelo 3D da mão com o que é observado através do sensor, em que o principal objetivo é tentar minimizar uma função de energia.

Uma função de energia ideal é o erro de reconstrução, ou seja, a distância entre o observado e a hipótese do modelo.

O trabalho descrito por Sharp (Sharp et al., 2015) e desenvolvido pela Microsoft, apresenta um sistema com uma abordagem hibrida, capaz de resolver algumas restrições descritas anteriormente, como o funcionamento apenas em close-range e sobre tudo a perda de movimento. Usam também o algoritmo PSO e uma função de energia, definida pelos autores de “golden energy”. No final permite-lhes obter um modelo 3D da mão o mais próximo possível do que é observada pelo sensor (figura 7).

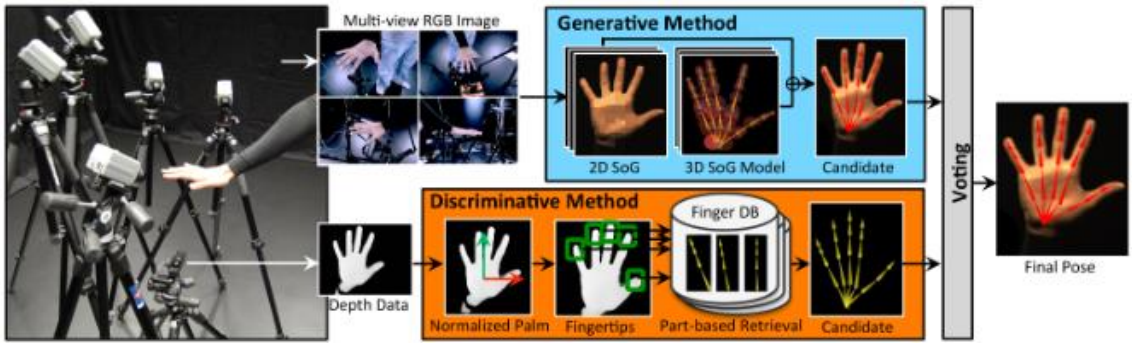

FIGURA 7 – PIPELINE DA SRINATH SRIDHAR (TAGLIASACCHI ET AL., 2015)

O trabalho descrito em Tagliasacchi (Tagliasacchi et al., 2015), ao contrario do anterior, usa um sensor diferente, o Senz3d é focado sobretudo no ICP(Iterative Closet Point) (figura 8). O ICP é um algoritmo de alinhamento, responsável por minimizar a diferença entre conjuntos de pontos de controlo. O ICP é geralmente utilizado para reconstruir superfícies 2D e 3D. OS dados de entrada do ICP são conjuntos de pontos de controlo. Estes conjuntos de pontos podem ser conseguidos através de algoritmos de deteção de contornos.

A combinação destas técnicas resulta nas poses ilustradas na figura 9.

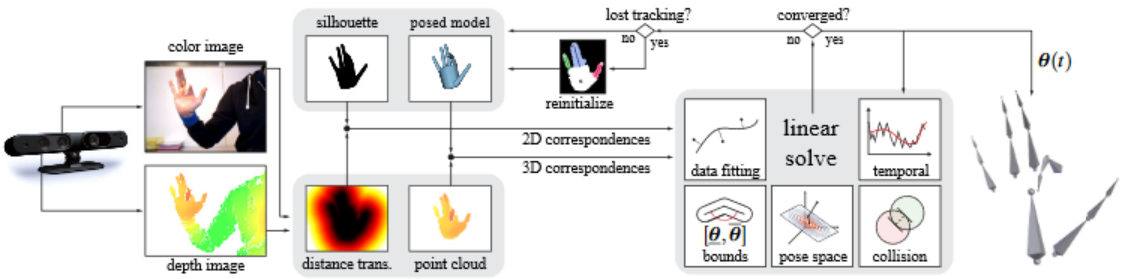

Figura 8 - Pipeline da Tagliasacchi (Tagliasacchi et al., 2015)

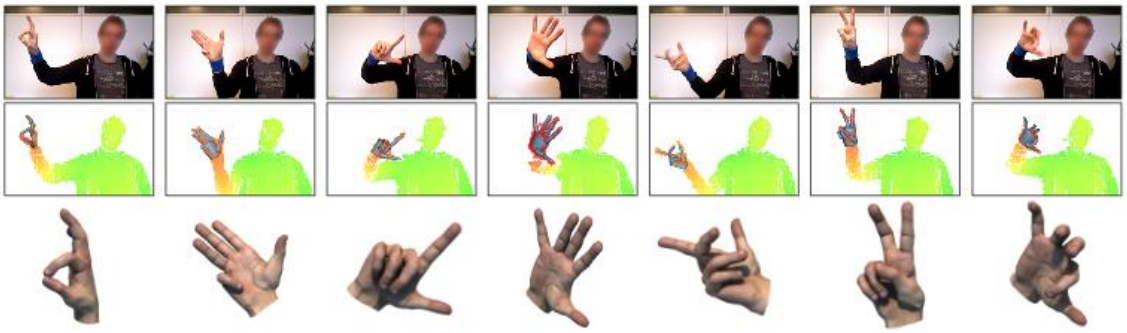

FIGURA 9 - RESULTADO DA TAGLIASACCHI (TAGLIASACCHI ET AL., 2015)

2.3.4 Modos de deteção de movimento da Creative Blasterx Senz3D/Intel RealSense

O Módulo da mão, da Intel RealSense, consiste em dois modos separados da deteção e movimento. Um dos modos consiste na deteção de movimento da mão completa (Full-hand) e/ou as extremidades (Extremities). O outro módulo é o do cursor que é um simples modo da deteção de movimento. Estes modos (tabela 3) diferem na informação que providenciam e nos recursos computacionais que exigem:

- Modo simples da deteção de movimento (Módulo do Cursor) – retorna um único ponto da mão, permitindo uma deteção de movimento com alta precisão, responsivo e limitado leque de gestos. O Módulo do Cursor foi projetado para resolver de casos de uso em Interfaces baseado com a mão.

- Modo Extremities (Módulo da mão) – retorna uma localização geral da silhueta e das extremidades da mão. Este modo foi projetado para fornecer um método simples da deteção e movimento da mão do utilizador.
- Modo Full-Hand (Módulo da mão) – retorna um esqueleto da mão em 3D, incluindo as vinte e dois pontos, informação dos dedos, gestos e mais. Este modo foi concebido para providenciar todas as características de deteção do movimento das mãos.

TABELA 3 - SUMARIZAÇÃO DOS MODOS

Modo do Deteção de movimento	**Unicamente mão?**	**Saída(Output)**	**Recursos Computacionais**	**Limitações**	**Câmara & alcance**
Simples (Cursor)	Sim	Ponto do cursor em 2D e 3D, alertas e gestos	Poucos, única linha de execução	2 mãos, velocidade rápida	110 cm
Full-Hand	Sim	Imagens segmentadas, Pontos de extremidade, alertas, info dos pontos, info dos dedos, abertura da mão e gestos	Elevado, encadeamento de execução	2 mãos, velocidade lenta	85 cm
Extermities	Sim	Imagens segmentas, Pontos de extremidades e alertas	Medio, única linha de execução	2 mãos, velocidade media	120 cm
Blob*	Não	Imagens segmentadas, Pontos de extremidade e linha de contorno	Poucos, única linha de execução	4 objetos, velocidade rápida	150 cm

*Blob: uma forma de uma imagem que representa um objeto.

Contudo a framework também é capaz de fazer a deteção de movimento de objetos. A interface do modo Blob e as suas interfaces relacionadas permitem que câmara detete objetos à sua frente e faz o extrato de imagens segmentadas, o contorno das linhas e dos pontos de interesses desse blobs.

Suavização dos movimentos das mãos

A suavização (Smother), é um algoritmo da Intel RealSense, pode ser usada para suavizar series de dados contendo pontos de 1,2 ou/e 3 dimensões numa variedade de algoritmo. O objetivo do algoritmo de suavização é para reduzir o ruído dos dados em série, causado por uma amostra atípica verdadeira ou por um erro na amostra. Suavizar uma série de pontos ao longo do tempo (por exemplo suavizar a mão ou posição de uma articulação ao longo do tempo) resulta numa posição da deteção e movimento mais estável e menos agitado de um objeto. A figura 10 ilustra suavização de uma série de dados.

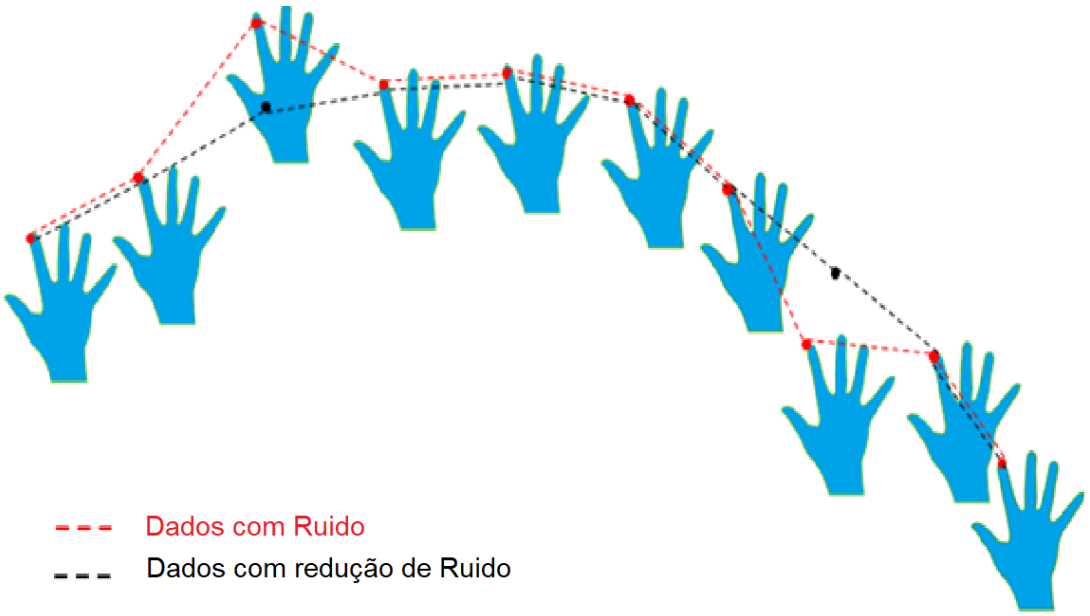

FIGURA 10 - SUAVIZAÇÃO DOS DADOS[6]

[6] https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_utils_the_smoother_utility.html, 20/11/2018

Podemos usar o algoritmo de suavização para suavizar qualquer série de dados online (não é necessários representar uma mão ou um objeto físico). O sentido de "Online" significa que os valores em série são recebidos um por um e suavizado imediatamente, e por oposto dos algoritmos "Offline", que recebe todos os dados em série primeiro e só depois é efetuada a suavização.

Algoritmos de suavização

Existem quatro algoritmos que podem ser aplicados aos dados em serie contendo os pontos de 1, 2 ou 3 dimensões. Assim sendo, é possível criar doze tipos de suavização (dependendo do algoritmo e do ponto da dimensão).

Os algoritmos de suavização disponíveis na framework da IntelRealSense são:

- Estabilizador – para estabilizar um ponto no espaço que representa um objeto estacionário. Ignora pequenas mudanças sobre um determinado limite e representa os pontos próximos com o mesmo ponto. O Movimento suavização também é aplicado quando um novo ponto excede o raio de estabilização.
- Média – substitui o ponto atual por uma média ponderada dos últimos N pontos, de acordo com pontos escolhidos.
- Quadrático – (baseado em tempo*) usa a equação quadrática para interpolar entre o ponto antigo com o ponto atual. Produz suavização do movimento e estabilização, dependendo da distância entre o novo ponto do ponto atual.
- "Spring" – (baseado em tempo*) usa a equação linear para interpolar entre o ponto antigo com o ponto atual (resultando num efeito de estabilização).[7]

[7] Em "baseado em tempo", a interpolação dos valores depende da passagem do tempo entre a aquisição das duas amostras.

2.4 Mesas multi toque com Câmaras de Profundidade

As mesas multitoque tradicionalmente só utilizam duas dimensões (X, Y), um exemplo é a Microsoft PixelSense.

Esta mesa foi a primeira versão Microsoft Surface, tendo sido lançada para o mercado no ano de 2008 foi descontinuada em 2011. (“Retro review,” 2017). A Microsoft Pixel Sense mesa foca em três principais componentes: interação direta, múltiplos toques e reconhecimento de objetos.

A interação direta refere-se capacidade de o utilizador simplesmente alcançar e tocar diretamente na interface de uma aplicação para interagir com ele. O multitoque refere-se capacidade de ter vários pontos de contacto numa interface. No reconhecimento de objetos é capacidade de o equipamento reconhecer a presença e orientação dos objetos colocados em cima dele (“Windows USER,” 2018). Na figura 11, visualiza-se a primeira versão da Microsoft Surface.

FIGURA 11 - MESA MULTI TOQUE DA MICROSOFT COM RECONHECIMENTO DE OBJETOS

No entanto, a utilização de câmaras de profundidade permite implementações de mesas multitoque que explorem as 3 dimensões (X, Y, Z).

Um exemplo destas mesas multitoque é a MT-50 (figura 12) desenvolvida pela companhia Ideum. A mesa, atualmente esta fora do mercado, tinha um ecrã de 127 cm, com uma resolução de 1280 cm por 780 cm que usa a Leap Motion para registar múltiplos utilizadores, múltiplos toques (“Ideum update 50-inch multitouch table [Video],” 2009) e usa software Snowflake para processamento ótico. Suporta mais de 50 toques simultâneos (“Museums to get high-res multi-touch table from Ideum,” 2009).

FIGURA 12 - MT-50 MULTITOUCH TABLE

Outro exemplo de uma mesa multitoque que usa a Kinect (figura 13) foi criado por “Bastian”, esta mesa usa o conceito de “exibição holográfica (holographic display)”, usando um projeto 2D e a Kinect. Esta mesa funciona no princípio de redesenhar o espaço 3D em relação a cabeça do utilizador. A Kinect capta a localização da cabeça em volta da mesa e move em três dimensões a grelha de caixas em direção oposta (“Multitouch table uses a Kinect for a 3D display | Hackaday,” 2012.). Em adição a Kinect a mesa usa o princípio da mesa da Microsoft Surface (“Augmented reality table - Simple 3D Brick-game - YouTube,” 2012.).

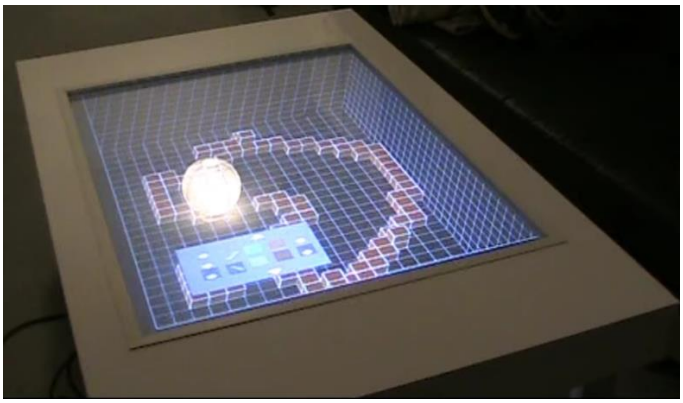

FIGURA 13 - MESA MULTITOUCH DO CRIADO PELO BASTIAN

O próximo exemplo de uma mesa multitoque é proposto por Eduardo S. Silva. Objetivo deste projeto é converter uma superfície transparente, num piano (embora de objetivo principal é tentativa de qualquer instrumento de toque). Esta mesa utiliza o Leap Motion, localizado por baixo da mesa, para a capar a profundidade/localização dos dedos do utilizador. Num entanto a projeção não se dá na mesa, mas sim numa parede ou um monitor a frente de utilizador. Nas figuras 14 e 15 é possível visualizar configuração da mesa com a projeção num monitor e outra configuração com a projeção na parede respetivamente (Silva et al., 2013).

FIGURA 14 - CONFIGURAÇÃO DA MESA COM MONITOR POR EDUARDO SILVA

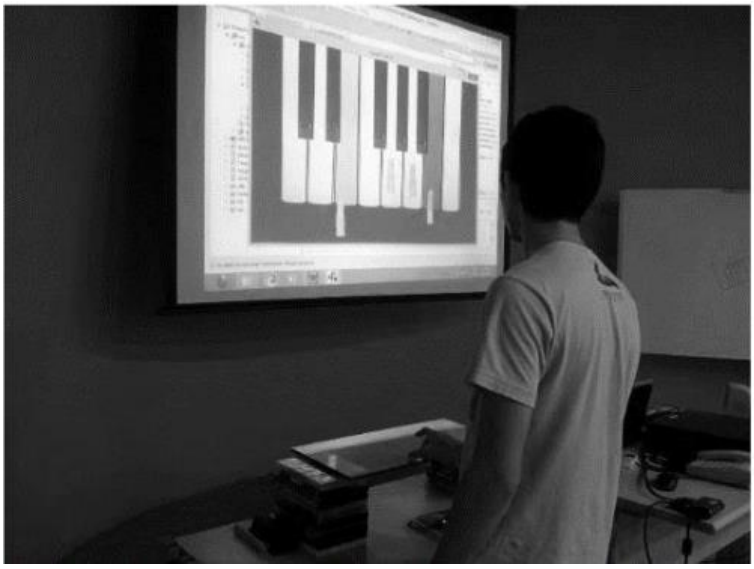

FIGURA 15 - CONFIGURAÇÃO DA MESA COM PROJEÇÃO DA PAREDE POR EDUARDO SILVA

Outro exemplo de uma mesa multitoque é proposto por Touchless Touch. Esta proposta utiliza Kinect ou com outras câmaras de profundidade para detetar os movimentos dos utilizadores. Ao contrário de outras mesas multitoque, a Kinect fica afastada da superfície tátil, normalmente diretamente por cima na superfície. Esta mesa funciona para maioria das aplicações disponível pelo Windows. Nas figuras 16 e 17 pode observar a melhor posição da Kinect face a superfície tátil e a utilização de uma aplicação (“How it Works - Touchless Touch,” 2015.).

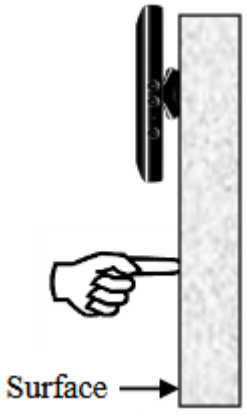

FIGURA 16 - LOCALIZAÇÃO DA KNIECT FACE ÁREA DE TOQUE, POR TOUCHLESS TOUCH

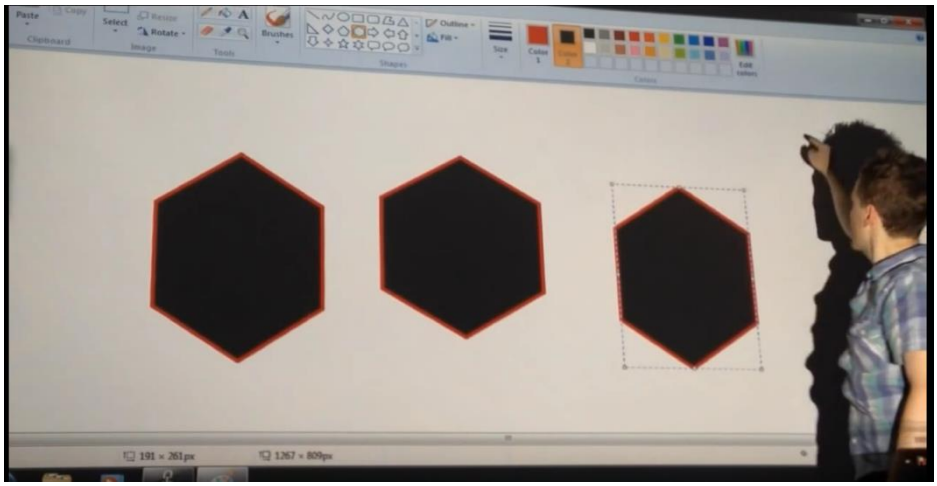

FIGURA 17 - FUNCIONAMENTO DA MESA COM APLICAÇÃO DA MICROSOFT, POR TOUCHLESS TOUCH

O último exemplo é a mesa multitoque Depthtouch. Esta mesa apresenta um conceito diferente das outras mesas mencionadas anteriormente. O conceito desta mesa é explorar elasticidade natural do tecido (figura 18) como modo de interação, adicionando a dimensão da profundidade. Com a utilização da Kinect, localizado por baixo do pano, que examina a compressão e elevação do pano, feita pelas mãos ou objetos, tais como esferas. Igualmente por baixo do pano esta um projetor que utiliza um espelho para aumentar a distância entre projetor e o pano. E utilizado um portátil ou computador para fazer o processamento da Kinect e do projetor.

As esferas virtuais, projetadas no pano, tem o mesmo comportamento do que as esferas reais, com a manipulação do pano com as propriedades hápticas, as esferas virtuais podem dispersas ou colecionadas (Peschke et al., 2012).

FIGURA 18 - PANO E INTERAÇÃO COM AS ESFERAS, DE PESCHKE

3

Implementação de Interações com Câmara de Profundidade

3. Implementação de Interações com a Câmara de Profundidade

Neste capítulo são apresentadas várias etapas da construção do protótipo, o desenvolvimento, o tratamento dos dados da captação das mãos ou da face e a construção do um frame para dar o utilizador o feedback háptico. É importante referir, que construção deste protótipo tem como objetivo de criar um cenário de trabalho que utiliza uma câmara de profundidade Creative Blasterx Senz3D, sendo a melhor opção para este protótipo comparado com as outras câmaras de profundidades disponíveis no mercado, como já foi referido anteriormente. Com este cenário os utilizadores terão possibilidade usar ações naturais, como gestos, expressões e movimentos da face, para controlar as funções da aplicação.

A aplicação foi desenvolvida utilizando o software Microsoft Visual Studio e a linguagem C#.

Um dos primeiros testes realizados foi capacidade da deteção, movimento das mãos e da face. Para tal, foi usado um exemplo proporcionado pela Intel. Na figura 19, podemos verificar os 22 pontos em cada mão e na figura 20 podemos visualizar os 78 pontos da face. Estas amostras também são uma excelente forma para testar as limitações da deteção de movimento, mais apropriadamente a distância, a velocidade e ângulos.

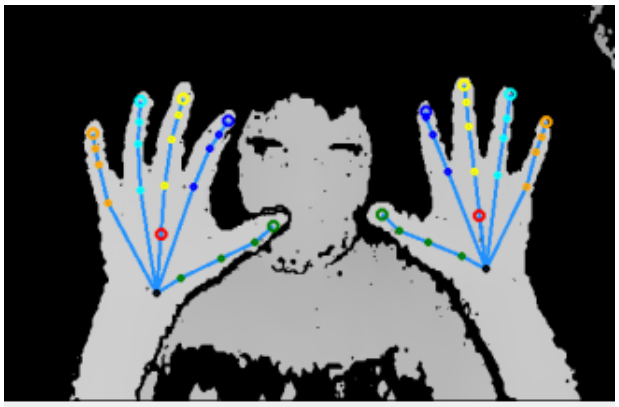

FIGURA 19 - AS 22 PONTOS DAS MÃOS

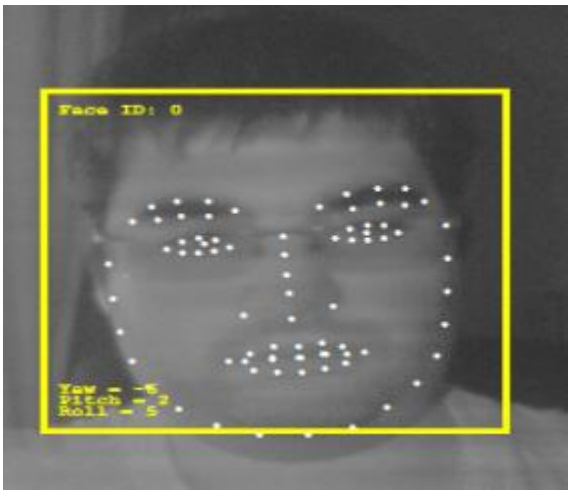

FIGURA 20 - OS 78 PONTOS DA FACE

A estrutura do exemplo é flexível o suficiente permitindo testar de várias alterações, num curto espaço de tempo, acelerando o processo da construção do protótipo. Na figura 21, podemos observar, numa forma simplificada como a estrutura do exemplo é implementada.

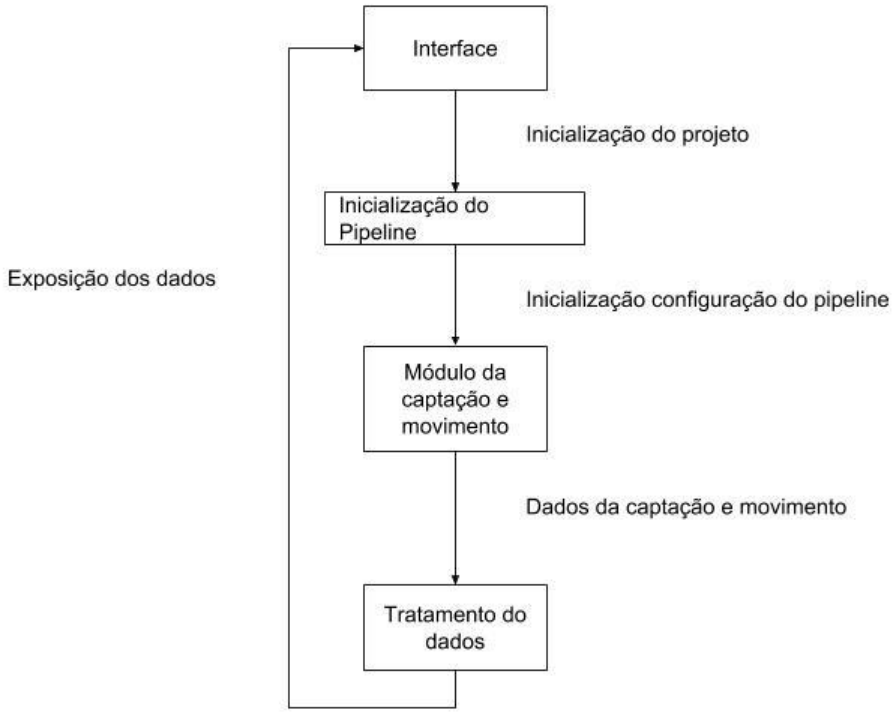

FIGURA 21 - ESTRUTURA SIMPLIFICADO DA EXEMPLO DA DETEÇÃO DE MOVIMENTO DAS MÃOS

3.1 Framework para Deteção de Dedos e Face

Como já foi referido, a arquitetura da Intel RealSense funciona em módulos e a comunicação dos módulos e a aplicação criado deste projeto é através de pipelines. Para inicializar a deteção de movimento das mãos e da face é necessário criar uma thread (linha execução) única para cada pipeline, que por sua vez só vai trabalhar unicamente com cada um dos módulos de deteção de movimento. A interface também esta incluída na mesma linha de execução do modulo das mãos, devido ao não causar impacto a nível do desempenho da aplicação.

É possível colocar múltiplos módulos na mesma linha execução, contudo acaba por trazer problemas a nível de performance, nomeadamente falhas sistemáticas (crashes) na aplicação construída. Com isto é necessário ter duas threads para os módulos utilizados neste projeto. Na figura 22 está representado um esquema simplificado como o projeto esta estruturado a nível pipelines e das threads.

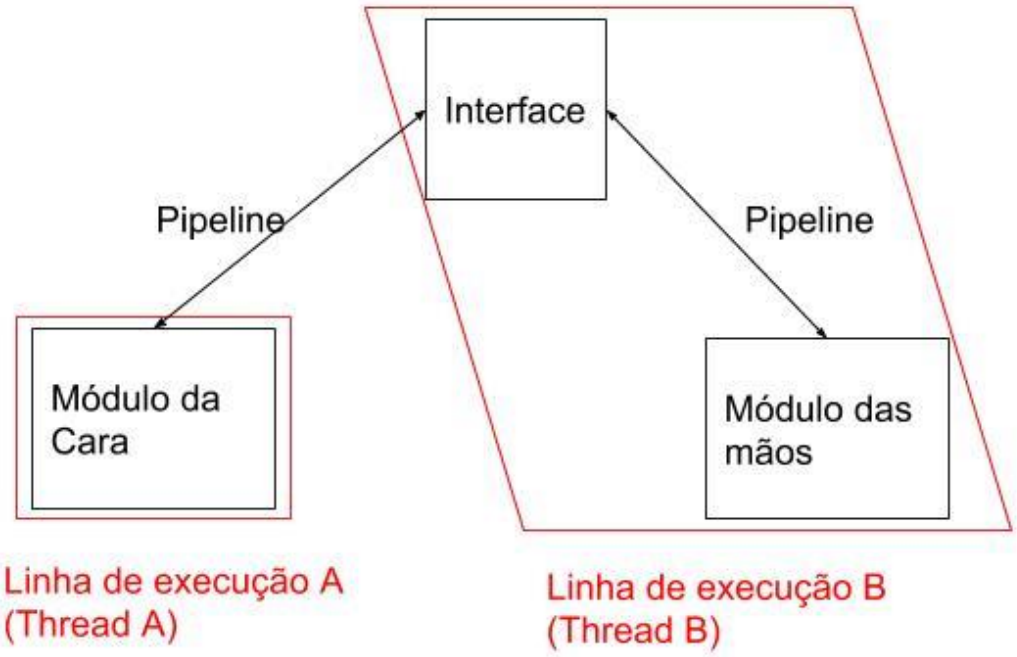

FIGURA 22 - ESQUEMA SIMPLIFICADO DAS PIPELINES E DAS LINHAS DE EXECUÇÃO (THREADS).

Com este conhecimento, o primeiro passo é a implementação de uma interface. Nesta interface é onde são executados os módulos da face e das mãos, a receção dos dados dos módulos e onde são criadas as funcionalidades que os utilizadores podem utilizar. Após da criação das threads, são criadas duas classes, uma para cada módulo. Nestas classes é onde ficam as configurações e tratamentos dos dados dos módulos.

Para dar o início da configuração da pipeline é necessário criar o SenseManager, da Classe PXCMSenseManager. Este é responsável conectar diretamente com a câmara e processa ações como a análise da face e das mãos. Na figura 23 podemos visualizar uma representação básica do trabalho do SensorManager.

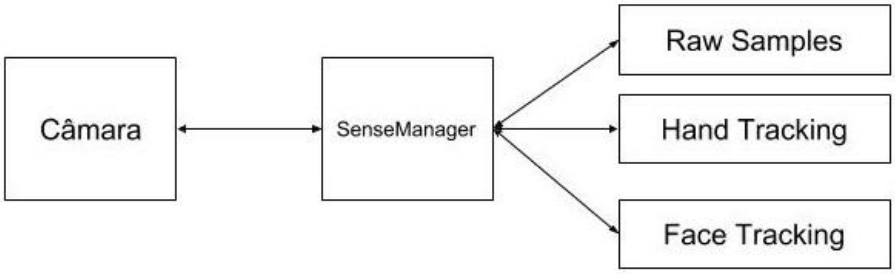

FIGURA 23 - REPRESENTAÇÃO SIMPLIFICADO DO SENSEMANAGER

Com o SenseManager criado, é necessário definir qual o módulo que será utilizado. Para definir qual o módulo ser utilizado, é necessário chamar o SenseManager que tem as propriedades para ativar módulos pretendidos.

O próximo passo são as configurações da pipeline relativamente ao módulo. Na deteção de movimento da mão existem dois modos: utilizar a extremidades da mão ou a mão completa. Para este projeto é utilizado o modo da mão completa, pois os outros modos não têm a capacidade de fornecer a informação dos vinte e duas pontos da mão.

O passo seguinte é a configuração dos gestos, que não difere muito do passo anterior, mas onde é necessário definir os gestos a serem reconhecidos. É possível ativar todos os gestos fornecido pela câmara, mas caso não seja

necessário do uso de todos os gestos, é melhor escolher quais são os gestos pretendidos. Ao limitar do número de gestos detetados, representa menor processamento da câmara o que cabe a deteção constante dos gestos.

Outra configuração necessária é do algoritmo de suavização, com a opção estabilizador. Este passo é importante pois sem ele os movimentos introduzidos ao rato não seriam regulares, ou seja, caso o utilizador faça um simples movimento da esquerda para a direita, o movimento do rato não seria completamente igual, introduzindo movimentos extras ao movimento original do utilizador.

Após aplicar as configurações é necessário criar um método de saída dos dados. Isto é conseguido através da chamada do módulo da mão. Em seguida é criada uma função em ciclo while, assim adquirindo e lendo os todos os frames e as informações e tendo a condição de paragem de algum erro nos frames recebidos.

O método principal para receber os frames é a função AcquireFrame, que tem o parâmetro “ifall” que por sua vez tem dois valores:

- “True” – aguarda até que todos os pedidos (input e output) sejam concluidas no frame atual e o todos os módulos de processamento concluem o processamento do frame atual.
- “False” – aguarda até que qualquer pedido (input e output) ou as operações dos módulos estejam concluídas.

O AcquireFrame trabalha diretamente com outro método chamado ReleaseFrame, pois cada frame deve ser processado e libertado para continuar a leitura dos próximos frames (ver figura 24).

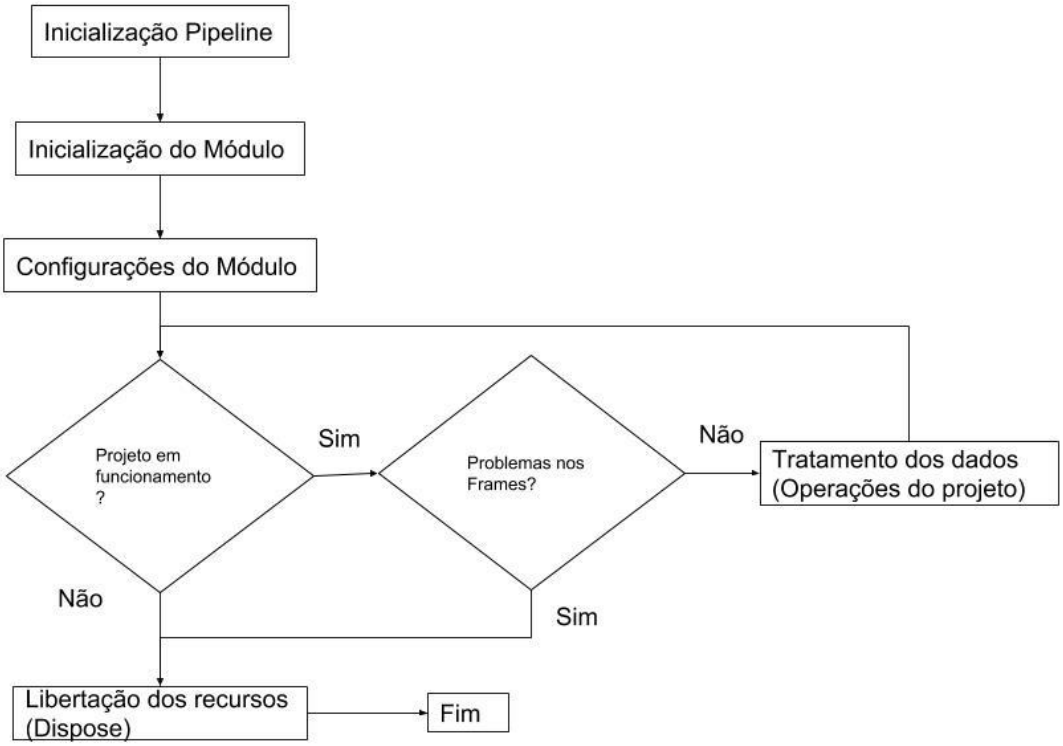

FIGURA 24 - FLUXOGRAMA DA ESTRUTURA DO PIPELINE

Com o ciclo while feito agora é possível fazer o tratamento dos dados. É criada uma função que vai servir como intermediário dos dados recebidos pelo modulo e da interface, e também na estruturação das bases das funcionalidades, relacionadas com as mãos e os gestos (que projeto irá conter). Uma das primeiras funcionalidades introduzidas é a capacidade de mover o cursor, através dos dados obtidos da posição da articulação pretendia da mão. Para este movimento é criada uma função separada, que trata de todas as funcionalidades do cursor para tal é utilizado o dll chamado por User32.dll.

O User32.dll é um componente da Windows que providencia funcionalidade para as interfaces do utilizador, tais como, gestão do Windows, passagem de mensagens e processamento de entrada de dados (“Windows USER,” 2018).

Concluindo esta etapa, juntamente com o algoritmo de suavização, é possível mover o rato com movimento semelhante do utilizador. O ponto de origem do rato na aplicação é no canto superior direito, algo inesperado, normalmente o ponto de origem costuma ser no canto inferior esquerdo em outras aplicações. Para

finalizar esta etapa são enviados os dados para a interface, onde é tratado o movimento dos objetos.

Para fazer uma seleção, o utilizador tem de fazer o gesto de apontar, colocando o dedo indicador a uma certa distância da câmara. O utilizador para fazer mover o objeto, tem de colocar o cursor em cima do objeto, mantendo o mesmo gesto, e esperar uns 2 segundos.

É importante referir que no gesto de apontar, o dedo indicador tem que estar na direção da câmara, independente do lado mão esta virada, e a mão não pode estar coberta, o que acabaria por destorcer a deteção da mão.

A implementação da deteção dos gestos é semelhante, o que difere é o desencadeio (trigger). A câmara está continuamente a recolher os dados ao longo do período de ação, em parte da deteção dos gestos é ligeiramente diferente. A câmara fica constantemente a verificar se o utilizador faz o gesto, e caso este seja observado os seus dados são enviados para a aplicação.

Os gestos que já estão definidos pela framework da Intel Realsense são mais simples em detetar pela câmara e não tem tantos limites, tais como, a profundidade e a localização do gesto que é realizada.

A implementação deteção e movimento da face é semelhante à implementação da deteção e movimento das mãos, mas é necessário fazer mais configurações no módulo da face relativamente ao módulo das mãos. Um dos passos importantes é decidir o modo de deteção e movimento, a Intel RealSense providencia cinco modos de deteção e movimento:

- Cor – Requer dados de cores na entrada do módulo.
- Profundidade – Requer dados de profundidade na entrada do módulo.
- Cor e Profundidade – Requer dados de cores e profundidade na entrada do módulo.
- “Color Still” – Requer os dados de cores e executa os algoritmos da deteção e movimento da face. Usando para processar imagens estáticas que normalmente não fazem parte de um filme/sequencia.
- Infravermelhos – Requere dados de Infravermelhos na entrada do módulo.

O modo escolhido é o infravermelho, motivos para esta escolha tem a ver com o posicionamento dos pontos da face, ou seja, os quatro primeiros tipos da deteção e movimento da face entre si utilizam a mesma lente, enquanto a deteção e movimento das mãos em infravermelhos utiliza outra lente. Em outros casos não influenciaria, mas, no entanto, ao existir uma de distância entre as duas lentes levam uma discrepância entre os pontos da face e das mãos (parallax), o que influencia a experiência do utilizador.

O próximo passo é limitar o número de faces que são detetadas pela câmara, de modo a poupar os recursos utilizados pela câmara, mas também, para remover a hipótese de outra pessoa se aproximar da câmara e ficar no controlo da aplicação. Para tal é necessario mudar configuração da pipeline levando a escolho da estratégia utilizada pela câmara para a escolha da face, apesar de não fazer alguma diferença, já que só pode ser uma pessoa a utilizar o projeto, contudo é uma configuração exigida pelo SDK, mas abre a possibilidade no futuro do desenvolvimento de projeto de colocar mais uma face a trabalhar no mesmo cenário.

Existem cinco estratégias na escolha da Face:

- Reconhecimento da mais recente.
- Da mais próxima a para a mais distante.
- Da mais distante para a mais próxima.
- Da esquerda para a direita.
- Da direita para a esquerda.

A estratégia escolhida para efeitos experimentais foi a da direita para esquerda, contudo já foi referido nenhum das escolhas influencia para o estado atual do projeto. No tratamento dos dados, é escolhido um ponto da face, para obter os dados da sua posição das coordenadas x e y. Com estes dados o utilizador é capaz de fazer de aumentar zoom, ao mover a cabeça para direita, e diminuir o zoom ao mover a cabeça para a esquerda.

Tal como no funcionamento dos gestos, as expressões funcionam na mesma maneira, a diferença entre eles é uma propriedade denominada de intensidade. A contrário dos gestos em que o utilizador necessita de realizar o gesto para provocar

uma reação, nas expressões necessitam de uma intensidade para evocar as funcionalidades do projeto, por exemplo, a expressão de abrir a boca, a medida que o utilizador abrir a boca, a intensidade da expressão vai aumentando.

3.2 Interface

A interface contém algumas opções (Figura 25), e é o primeiro contacto com o utilizador e tem como o objetivo de servir como zona de trabalho para o utilizador e como base das funcionalidades.

É na interface que são construídas as funcionalidades típicas das mesas multitoque, tais como o movimento dos objetos e o aumento/diminuição da imagem.

Um facto de referir é a dimensão dos botões, são maiores comparativamente a botões de outras aplicações, pois apesar dos cuidados na replicação dos movimentos das mãos do utilizador, é difícil de reproduzir estes com exatidão. Ao aumentar o tamanho dos botões torna-se mais fácil para o utilizador interagir com eles.

FIGURA 25 - INTERFACE

A interface esta dividida em três secções principais. A secção 1 contém três botões: Start, Stop e Insert Image.

- Start – Corresponde o início das funcionalidades da câmara.
- Stop – Corresponde o parar das funcionalidades da câmara.
- Insert Image – Inserção da imagem na interface.
-

A secção 2 apresenta três opções que o utilizador pode escolher que influenciam o ambiente de trabalho e a imagem introduzida:

- Increase Face – Aumento ou diminuição da imagem.
- Gesture Lock – Ativação dos gestos na interface.
- Hide – Esconde as opções.

E na última secção, secção 3, é a área onde o utilizador pode interagir com a imagem ou fotos da interface, onde pode mover e aumentar/diminuir a imagem.

A figura 26, demonstra como o utilizador pode interagir com a imagem, como anteriormente mencionado, ao colocar o rato em cima da imagem e espera uns segundos, a imagem seguirá os movimentos do rato.

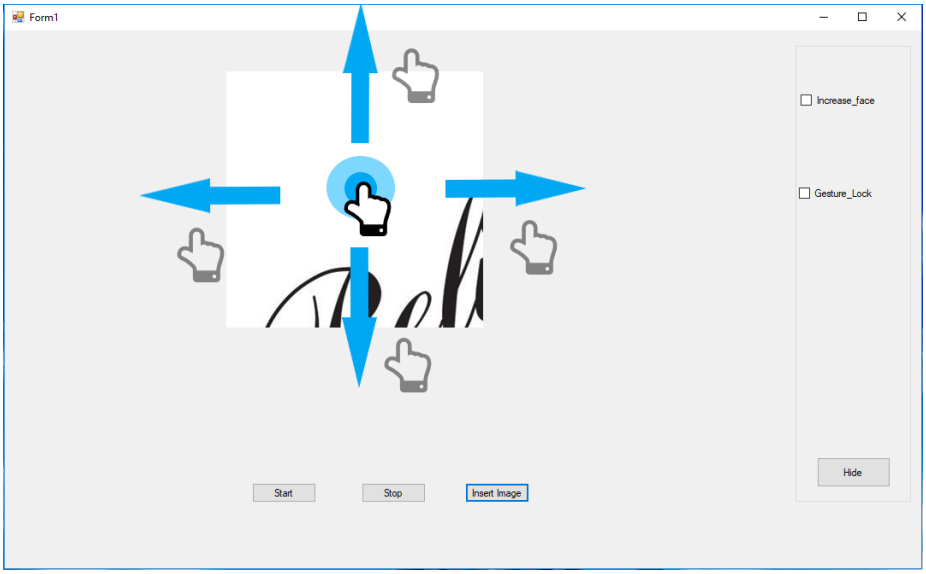

FIGURA 26 - MOVIMENTO DA IMAGEM COM O RATO

Na figura 27 demostra a imagem do tamanho normal introduzido na aplicação. As imagens 28 e 29, demonstram como o utilizador pode aumentar e diminuir a imagem com os movimentos da cabeça do utilizador. Ao mover a cabeça para direita aumenta a imagem e ao mover a cabeça para esquerda diminui a imagem.

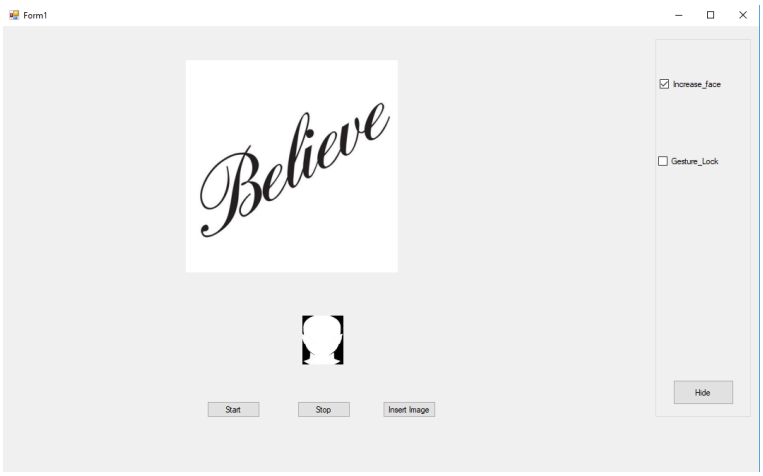

FIGURA 27 - IMAGEM COM TAMANHO ORIGINAL

FIGURA 28 – AUMENTO DA IMAGEM

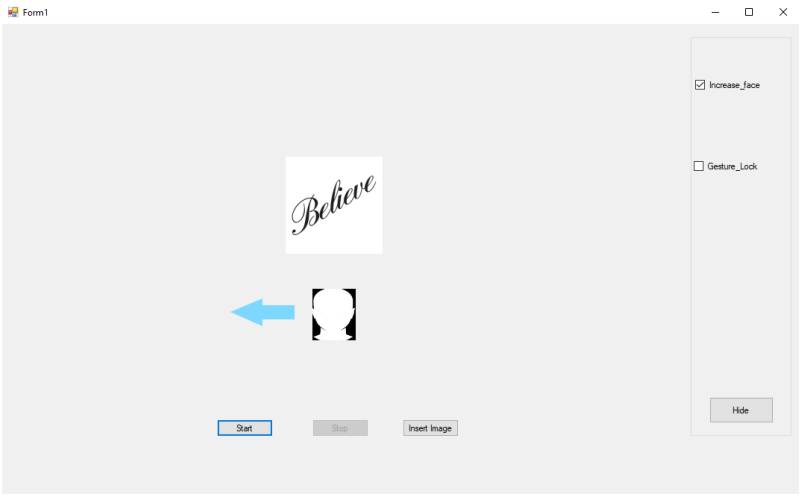

FIGURA 29 - DIMINUIÇÃO DA IMAGEM

É importante referir que ambas as interações são independentes entre si, tanto nível de implementação. Mas é possível combinar as várias funcionalidades aos mesmo tempo (figura 30). Contudo as emoções e gestos são prioritários aos movimentos devido ao funcionamento arquitetura da framework da Realsense.

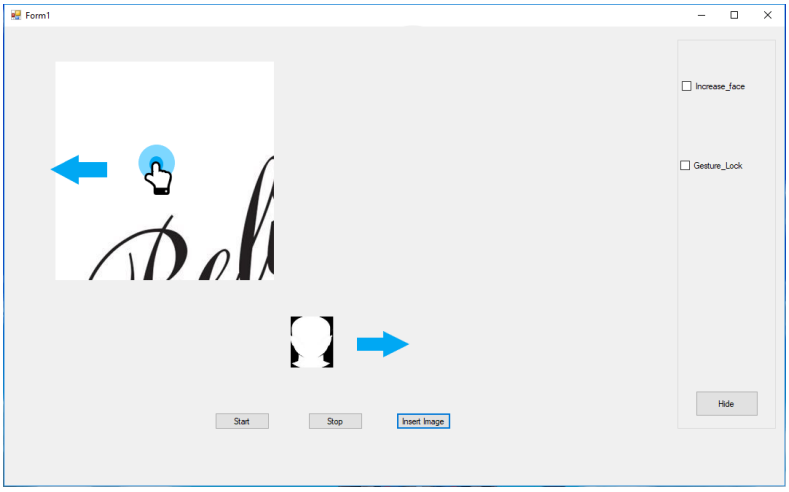

FIGURA 30 - COMBINAÇÃO DE DUAS FUNCIONALIDADES

3.3 Frame háptico

O feedback háptico é também um fator importante para este trabalho e ajuda-nos em duas vertentes: diminuição de erros humanos e na confortabilidade do utilizador, tal como demostrado dos trabalho referidos anteriomente.

Neste projeto o feedback háptico é conseguido através de um frame, com as dimensões de 48 cm por 32 cm, que fixa um material flexível e transparente, como o acetato (figura 31). Com a flexibilidade do material é possível explorar a profundidade, ou seja, quando o utilizador pressiona o acetato, a câmara deteta a profundidade do gesto. Isto permite múltiplas funcionalidades que depende da pressão (figura 32) que o utilizador aplica no acetato. Uma das funcionalidades que utiliza esta mecânica é a seleção. O framework reconhece que o utilizador está a fazer uma seleção até a chegar uma certa profundidade.

O frame serve também como guia para utilizador, dando feedback dos limites da captação da câmara, assim levando a reduzir erros de utilização.

O outro uso do frame é de servir como base, onde o utilizador pode apoiar as mãos, assim diminuindo o cansaço do utilizador após várias horas de uso de uma aplicação com gestos livres (mid-air gestures).

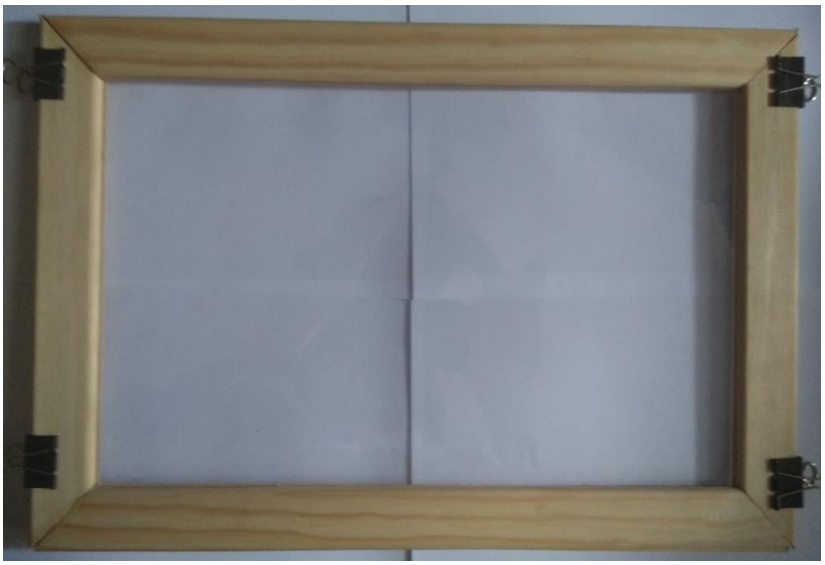

FIGURA 31 - FRAME MONTADO

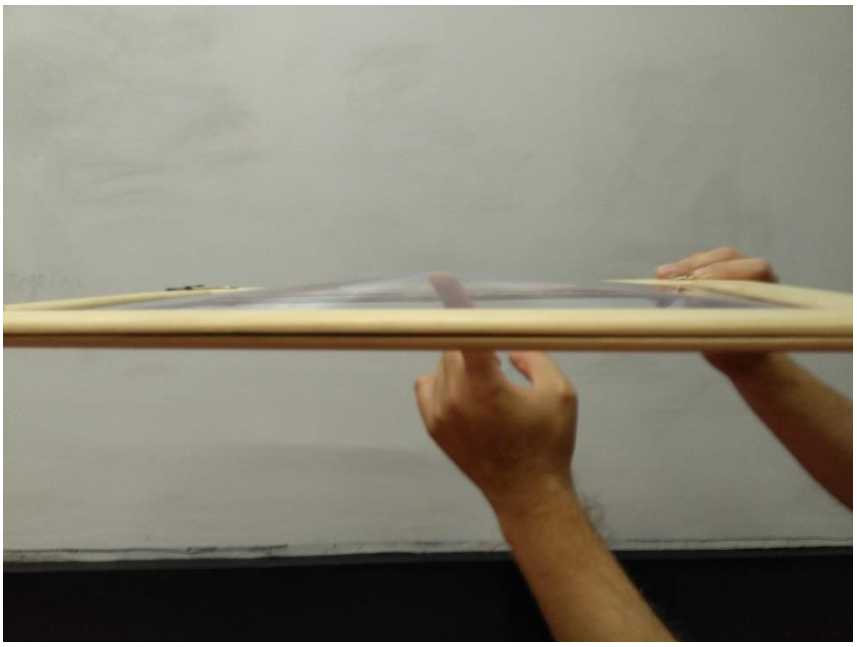

FIGURA 32 - LIMITE DA PROFUNDIDADE DO FRAME

A montagem do frame é composto por dois quadros de madeira e entre eles contem um acetato transparente. O motivo de ter dois quadros, é o fácil acesso ao acetato tendo assim a possibilidade de o remover sem causar danos. Ao contrário de um único quadro, o acetato iria sofrer danos na tentativa de o fixar na estrutura de madeira. Para fixar estes dois quadros e o acetato são usadas molas para prender papel, duas em cada lado, sendo assim de fácil montagem/desmontagem.

Contudo o uso da frame experimental também traz algumas desvantagens. A principal desvantagem é o próprio acetato, este reflete parte da luz infravermelho, o que dificulta o processo da captação tanto para as mãos como para a face, contudo o material escolhido cumpriu requerimento iniciais.

3.4 Protótipo final

Nas imagens 33 e 34 é possível visualizar o protótipo final. A configuração é simples, colocar a câmara e o frame ao nível da face, mas é possível estar mais acima ou em baixo, no entanto traz algumas dificuldades a nível da captação da face. O portátil é usado para o processamento dos dados recebidos pela câmara e na ligação com o projetor Optoma (“Optoma S340 DLP SVGA Business

Projector," 2015.). O projetor é usado para aumentar a imagem ficar maior tornando assim mais confortável usar a aplicação em pé.

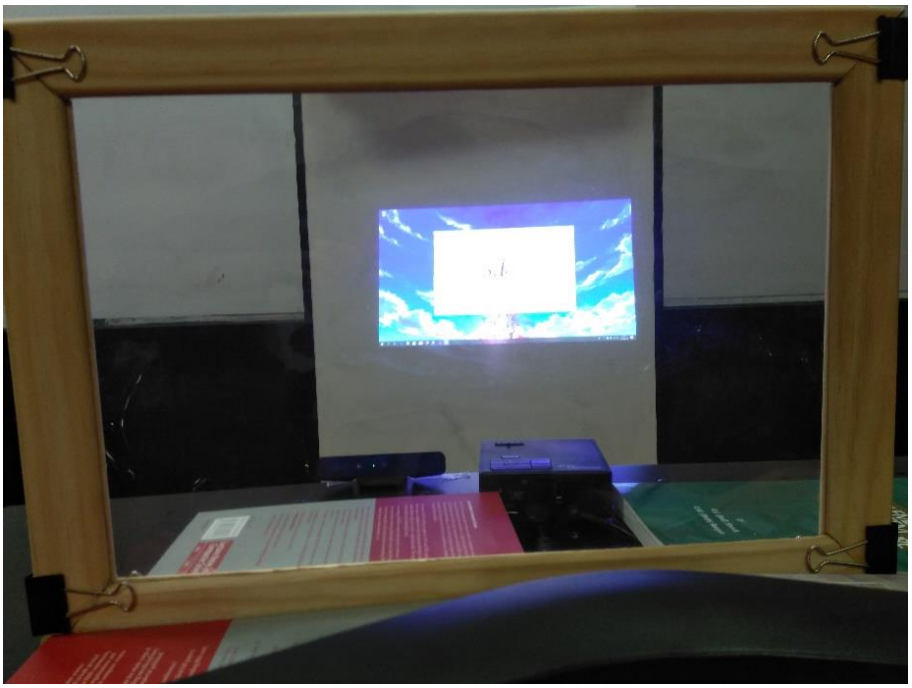

FIGURA 33 - O PROTÓTIPO FINAL

FIGURA 34 - INTERAÇÃO COM O PROTÓTIPO FINAL

4

Avaliação

4. Avaliação

Em seguida falaremos sobre as principais oito avaliações realizadas, dos módulos, dos problemas que surgiam desses testes e nas suas possíveis soluções, ao longo do desenvolvimento deste projeto. A primeira avaliação avalia realizada foi determinar quais os dados eram possíveis obter dos pontos da face e pontos da face, a segunda avaliação consiste nos movimentos do rato com dados da articulação, a terceira avaliação foi a tentativa se era possível colocar dois módulos da mesma pipeline sem ter nenhum problema. A quarta avaliação a posição da imagem captada pelos os sensores da câmara, a quinta avaliação realização foi as replicar as funções dos ratos (exemplo um clique) através das interações gestuais a sexta avaliação a replicação de outras funções através dos gestos e expressões. Na última avaliação foi testado os limites da captação da câmara.

4.1 Tipos de Dados

Primeira avaliação realizada, consiste na tentativa de obter os tipos de dados relativamente dos pontos das mãos e os pontos da face. Através da framework, é possível obter as coordenadas de todos os pontos das mãos, num plano de 2D e separadamente num plano 3D. É possível escolher quais os pontos que queremos recolher os dados, permitindo uma melhor performance da câmara e a redução dos recursos necessários.

4.2 Movimento do rato

O segundo teste foi verificar o comportamento do rato, com os dados inseridos da articulação escolhida. Como esperar os movimentos do rato não condiziam com os movimentos da mão, devido ao ruído dos dados captados pela câmara. Para resolver este problema foram implementados simples algoritmos para tentar reduzir o ruído, mas sem muito sucesso. Decidimos então utilizar o algoritmo fornecido pela Intel RealSense. Com a utilização foi possível verificar uma melhoria considerável nos movimentos do rato.

4.3 Múltiplos módulos no mesmo pipeline

Terceira avaliação, foi na tentativa de introduzir head tracking na mesma pipeline do modulo da hand tracking. Esta acabou por trazer alguns problemas no nível do software. O principal problema eram os constantes “crashes” do sistema, devido da saturação de informação pelo passado pelo o pipeline. A solução para este problema foi criar mais um pipeline, mas por consequência, levou ao aumento do consumo dos recursos, tanto a câmara e para o computador.

4.4 Posição da imagem em relação aos sensores

Quarta avaliação realizada sobre os pontos da face e os pontos da mão, foi detetada uma translação entre as imagens obtidas na deteção de movimento das mãos e da face. Este problema foi detetado da tentativa de colocar os pontos das mãos e dos pontos da face na imagem. A resolução deste problema, foi na utilização de modo de visualização por infravermelhos, com este modo a câmara utiliza a mesma lenta da deteção de movimento das mãos, assim corrigindo o problema.

4.5 Interações com rato

Na quinta avaliação foi efetuado um teste de usabilidade informal com diferentes utilizadores com o objetivo de saber qual o melhor método se fazer um gesto de pressionar e mover. Foram feitas várias tentativas para explorar quais eram as melhores combinações para fazer esta ação mais natural, contudo, não importa quais as combinações criadas, as pessoas continuam a preferir fazer esta ação, na maneira tradicional, como os smartphones e os tablets.

O principal comentário extraído dos vários comentários foi o facto de ser mais natural em comparação a utilização do rato.

4.6 Interações com gestos e expressões faciais

Sexta avaliação realizada, foi nos gestos e nas expressões, ambos semelhantes na maneira que são implementados e nos seus problemas. Tal como os problemas dos dedos, já mencionados, as expressões e os gestos também contêm semelhantes problemas, ainda por adição, quando realizamos um gesto ou

uma expressão, a framework deteta essas emoções ou gestos em múltiplos frames, pois cada emoção ou gestos leva vários segundos para ser completado. Para resolver este problema é construído um sistema de exclusão de frames. Este sistema consiste na deteção da ação pretendida no primeiro frame e ignora os seguintes frames, o sistema não vai verificar os seguintes frames, até o utilizador deixar de realizar esta ação, o sistema liberta-se e esta a verificar o próximo gesto ou expressão.

4.7 Limite da captação da câmara

Por última avaliação, foi testado os limites da captação da câmara. Neste teste foram colocadas seis pessoas à frente da câmara, como a captação da face com todos os pontos. Inicialmente foram apenas colocadas quatros pessoas a frente da câmara, por consequência, foi verificado uma descida dos frame por segundos (fps) para 20, tendo em conta apenas uma face detetada pela câmara, os quadros por segundos eram de 30. Com seis pessoas a frente da câmara os quadros por segundo era de 15 estáveis (tabela 4). Também verificado um tempo de demora (delay), proximamente de 5 a 10 segundos, para a deteção das faces nas últimas duas pessoas. Outra observação foi o fato da câmara não conseguir acompanhar completamente os movimentos (esquerda, direita, cima, baixo e girar) da cabeça dos utilizadores e facilmente perdia a captação mesmo estar próximo da câmara. Na figura 35 podemos visualizar a aplicação a captar as cabeças dos utilizadores.

Este teste foi realizado em dois computadores diferentes, modo há verificar se existia alguma diferença nos frames. Contudo não foi verificado nenhuma diferença, logo concluindo so o hardware da câmara influencia na captação das imagens.

Tabela 4 - Frames por segundos por pessoa

Número de pessoas	Frames por segundo
4	20
6	15

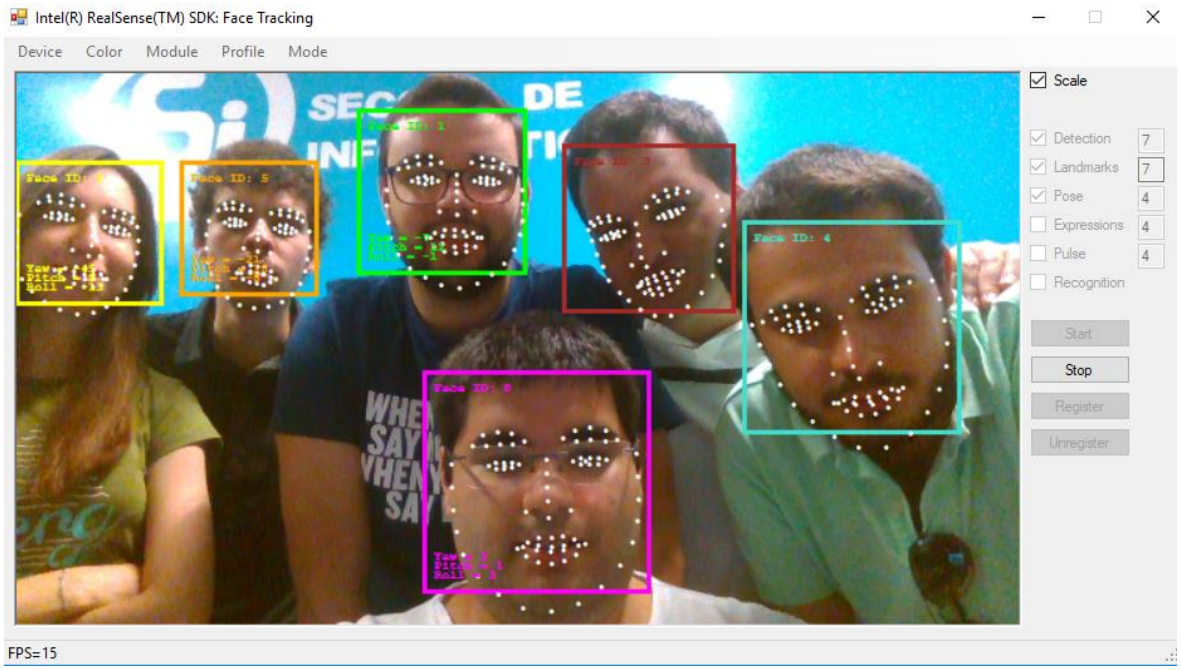

FIGURA 35 - CAPTAÇÃO DAS FACES DOS UTILIZADORES

5

Conclusão

5. Conclusão

O objetivo deste trabalho foca-se na tentativa de criar um cenário de aplicação sem os meios tradicionais (teclado e rato) e explorar interações faciais e gestuais através de uma câmara de profundidade.

Em primeiro ligar, destaca-se a separação dos módulos da câmara, nomeadamente a deteção de movimento da face e mão, em duas threads assim garantindo boa performance e não haja interferência da informação enviada por pipelines dos módulos.

Para a deteção de movimento da face e das mãos, as configurações iniciais dos módulos são muito semelhantes entre si, diferenciando na parte das configurações, no módulo da deteção de movimento da face, as principais configurações consistem na prioridade e o número de faces que são detetadas, enquanto no módulo da deteção de movimento das mãos consiste na primeira mão detetada e no modo que a mão é detetada, em que mão completa é escolhida neste projeto.

Depois das configurações dos módulos, foram criadas funções para o tratamento dos dados de cada módulo. Na primeira função do módulo da deteção de movimento das mãos, permite mover do rato, a segunda função trata dos gestos.

A primeira função permite obter os dados das pontos e com esses dados, introduzi-los no rato, assim permitindo o movimento, contudo a câmara capta dados extra, que acabam por prejudicar o movimento do cursor. Para contornar este problema foi utilizado um algoritmo suavização da Intel RealSense, que permite reduzir esses dados, denominado por ruído de dados. Também esta função trata de outras operações do rato, como o clique e o arrastar dos objetos.

Na segunda função ocupa-se de detetar os gestos realizados pelo utilizador, esta função funciona basicamente como uma interrupção, quando o utilizador realiza certo gesto, uma funcionalidade é executada no projeto.

No caso do tratamento dos dados do módulo da deteção de movimento da face, foram criadas duas funções. A primeira função trata das posições dos pontos da face e recolhendo os dados da posição dos pontos da face, permitindo o utilizador

aumentar e diminuir a imagem. A segunda função é focada da deteção das emoções do utilizador.

Os objetivos definidos inicialmente foram cumpridos com sucesso. Mas, o desenvolvimento deste protótipo requere trabalho futuro para ultrapassar as limitações encontradas.

5.1 Limitações

A maior limitação encontrada no projeto, é a versão utilização da framework da IntelSense. Este projeto utiliza a framework Intel Realsense SDK 2016 R3, que obviamente pelo título saiu em 2016. Contudo a versão mais recente da Intel, a Realsense SDK R2.0 não inclui os métodos de captação e movimento e outras funcionalidades presentes na versão anterior deixados para uma integração com processamento de imagem (OpenCV). No entanto, esta possui uma melhor performance, dando a possibilidade ao utilizador usar o multi-threading, o que leva a possibilidades de usar múltiplos módulos de captação e movimento bem como a captação maior número de faces e mãos, sem muito impacto na performance. Já a versão da framework de 2016 R3, isto já não é possível, confirmado pela a avaliação realizada.

Outra limitação é a versão dos drivers da câmara. A Intel declarou, no final do 2016, que a câmara já esta no final da fase da vida, sendo a D435i o modelo mais atual, e por consequência já não há suporte total para câmara utilizada no projeto (SR300). O próprio sistema operativo (Windows 10), também não suporta driver que trabalha com a framework da câmara, e devido a esta desatualização existem alguns problemas a nível do software da própria câmara.

A utilização do frame também apresentação limitações, como já foi referido a utilização do acetato como meio de dar feedback ao utilizador, causa alguns problemas na deteção das mãos e principalmente a face. Através do acetato ainda consegue captar as mãos, contudo apresenta atraso nos movimentos, e na captação da face é muito inconsistente, acabando por muito das vezes na paragem da captação da face.

Uma restrição colocada neste projeto é o número de utilizadores que podem interagir com este. Como desenvolvimento inicial optou-se por limitar a interação com um único utilizador. No entanto, desenvolvimentos futuros devem ter em conta múltiplos utilizadores.

5.2 Trabalho futuro

Neste projeto foi desenvolvido um protótipo com várias funcionalidades são provas de conceito. Em adição destaco alguns objetivos para o melhoramento do projeto:

1. Introdução de mais gestos e emoções para interação com o sistema.
2. Introdução de novos métodos e meios para o melhoramento do desempenho do projeto.
3. Adicionar outros formatos de documentos no projeto. Por exemplo, documentos com texto, imagens (PDF) e vídeos.
4. Mais testes na escolha do melhor material para o Frame.
5. Interações com múltiplos utilizadores.

6. Bibliografia

Andolina, S., Klouche, K., Cabral, D., Ruotsalo, T., Jacucci, G., 2015. InspirationWall: Supporting Idea Generation Through Automatic Information Exploration, in: Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition, C&C '15. ACM, New York, NY, USA, pp. 103–106. https://doi.org/10.1145/2757226.2757252

Augmented reality table - Simple 3D Brick-game - YouTube [WWW Document], 2012. URL https://www.youtube.com/watch?time_continue=1&v=2CbiOikirrg (accessed 9.8.18).

BlasterX Senz3D - - Creative Labs (United States) [WWW Document], 2016. URL https://us.creative.com/p/web-cameras/blasterx-senz3d (accessed 9.7.18).

Das, R., Shivakumar, K.B., 2016. Augmented World: Real Time Gesture Based Image Processing Tool with Intel Real Sense™ Technology. Int. J. Signal Process. Image Process. Pattern Recognit. 9, 63–84. https://doi.org/10.14257/ijsip.2016.9.1.07

Dezentje, P., Cidota, M.A., Clifford, R.M.S., Lukosch, S.G., Bank, P.J.M., Lukosch, H.K., 2015. Designing for Engagement in Augmented Reality Games to Assess Upper Extremity Motor Dysfunctions, in: 2015 IEEE International Symposium on Mixed and Augmented Reality - Media, Art, Social Science, Humanities and Design. Presented at the 2015 IEEE International Symposium on Mixed and Augmented Reality - Media, Art, Social Science, Humanities and Design (ISMAR-MASH'D), IEEE, Fukuoka, Japan, pp. 57–58. https://doi.org/10.1109/ISMAR-MASHD.2015.24

Face Landmark Data [+JS,UWP] [WWW Document], 2016. URL https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_face_face_landmark_data.html (accessed 9.8.18).

Han, J., Gold, N.E., 2014. Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design [WWW Document]. Proc. Int.

Conf. New Interfaces Music. Expr. URL http://nime2014.org/proceedings/papers/485_paper.pdf (accessed 9.26.18).

Hand Joints [WWW Document], 2016. URL https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_hand_hand_joints.html (accessed 9.8.18).

Home - Leap Motion [WWW Document], 2013. URL https://www.leapmotion.com/ (accessed 9.7.18).

How it Works - Touchless Touch [WWW Document], 2015. URL http://www.touchlesstouch.com/howitworks.php (accessed 9.8.18).

Ideum update 50-inch multitouch table [Video] [WWW Document], 2009. . SlashGear. URL https://www.slashgear.com/ideum-update-50-inch-multitouch-table-video-2052993/ (accessed 9.8.18).

Intel® RealSense Technology | Intel® Software [WWW Document], 2016. URL https://software.intel.com/en-us/realsense (accessed 9.8.18).

Kinect – Desenvolvimento de aplicações do Windows [WWW Document], 2011. URL https://developer.microsoft.com/pt-pt/windows/kinect (accessed 9.7.18).

Mann, A., 2011. Scientists Hack Kinect to Study Glaciers and Asteroids. Wired.

McWilliams, A., 2013. How a Depth Sensor Works - in 5 Minutes | Andrew McWilliams [WWW Document]. URL https://jahya.net/blog/how-depth-sensor-works-in-5-minutes/ (accessed 9.6.18).

Multitouch table uses a Kinect for a 3D display | Hackaday [WWW Document], 2012. URL https://hackaday.com/2012/04/30/multitouch-table-uses-a-kinect-for-a-3d-display/ (accessed 9.8.18).

Museums to get high-res multi-touch table from Ideum, 2009. . Geek.com.

Oikonomidis, I., Kyriazis, N., Argyros, A., 2011. Efficient model-based 3D tracking of hand articulations using Kinect, in: Procedings of the British Machine Vision Conference 2011. Presented at the British Machine Vision Conference 2011, British Machine Vision Association, Dundee, pp. 101.1-101.11. https://doi.org/10.5244/C.25.101

Optoma S340 DLP SVGA Business Projector [WWW Document], 2015. Optoma. URL http://optomaeurope.com/product/S340 (accessed 9.27.18).

Ott, R., Gutierrez, M., Thalmann, D., Vexo, F., 2005. Improving User Comfort in Haptic Virtual Environments through Gravity Compensation, in: First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. Presented at the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE, Pisa, Italy, pp. 401–409. https://doi.org/10.1109/WHC.2005.78

Peschke, J., Göbel, F., Gründer, T., Keck, M., Kammer, D., Groh, R., 2012. DepthTouch: an elastic surface for tangible computing, in: Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12. Presented at the the International Working Conference, ACM Press, Capri Island, Italy, p. 770. https://doi.org/10.1145/2254556.2254706

Pham, T., Pathirana, P., Trinh, H., Fay, P., 2015. A Non-Contact Measurement System for the Range of Motion of the Hand. Sensors 15, 18315–18333. https://doi.org/10.3390/s150818315

Retro review: Microsoft's 2008 Surface "coffee table" in 2017 [WWW Document], 2017. . Window Cent. URL https://www.windowscentral.com/microsoft-surface-pixelsense-table (accessed 9.8.18).

Roth, H., Vona, M., 2012. Moving Volume KinectFusion, in: Procedings of the British Machine Vision Conference 2012. Presented at the British Machine Vision Conference 2012, British Machine Vision Association, Surrey, pp. 112.1-112.11. https://doi.org/10.5244/C.26.112

Ruotsalo, T., Klouche, K., Cabral, D., Andolina, S., Jacucci, G., 2016. Flexible Entity Search on Surfaces, in: Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, MUM '16. ACM, New York, NY, USA, pp. 175–179. https://doi.org/10.1145/3012709.3012732

Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Krupka, E., Fitzgibbon, A., Izadi, S., Kohli, P., 2015. Accurate, Robust, and Flexible Real-time Hand Tracking.

Silva, E.S., Abreu, J. de, Almeida, J., Teichrieb, V., Ramalho, G.L., 2013. A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments.

Sridhar, S., Oulasvirta, A., Theobalt, C., 2013. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data, in: 2013 IEEE International Conference on Computer Vision. Presented at the 2013 IEEE International Conference on Computer Vision (ICCV), IEEE, Sydney, Australia, pp. 2456–2463. https://doi.org/10.1109/ICCV.2013.305

Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., Pauly, M., 2015. Robust Articulated-ICP for Real-Time Hand Tracking. Comput. Graph. Forum 34, 101–114. https://doi.org/10.1111/cgf.12700

Talldrinks, 2011. Microsoft Kinect Turntable 3D Scanner.

Tscheligi, M., Houde, S., Kolli, R., Marcus, A., Muller, M., Mullet, K., 1995. Creative Prototyping Tools: What Interaction Designers Really Need to Produce Advanced User Interface Concepts, in: Conference Companion on Human Factors in Computing Systems, CHI '95. ACM, New York, NY, USA, pp. 170–171. https://doi.org/10.1145/223355.223485

Varcholik, P.D., Laviola, J.J., Jr., Hughes, C.E., 2012. Establishing a Baseline for Text Entry for a Multi-touch Virtual Keyboard. Int J Hum-Comput Stud 70, 657–672. https://doi.org/10.1016/j.ijhcs.2012.05.007

Vilaça, R., Ramos, J., Silva, V.C.A., Sepúlveda, J., Esteves, J.S., 2017. Mobile platform motion control system based on human gestures. Int. J. Mechatron. Appl. Mech. 2017, 267–273.

Wasenmüller, O., Stricker, D., 2017. Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision, in: Chen, C.-S., Lu, J., Ma, K.-K. (Eds.), Computer Vision – ACCV 2016 Workshops. Springer International Publishing, Cham, pp. 34–45. https://doi.org/10.1007/978-3-319-54427-4_3

Windows USER, 2018. . Wikipedia.

Zhang, Z., 2012. Microsoft Kinect Sensor and Its Effect. IEEE Multimed. 19, 4–10. https://doi.org/10.1109/MMUL.2012.24