

**DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÕES LIVRE DE  
MARCADORES UTILIZANDO SENSORES DE PROFUNDIDADE**

**ÉLISSON MICHAEL FERNANDES MEIRELLES ARAÚJO**

**UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE – UENF**

**CAMPOS DOS GOYTACAZES – RJ**

**MARÇO – 2015**

# DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÕES LIVRES DE MARCADORES UTILIZANDO SENSORES DE PROFUNDIDADE

**ÉLISSON MICHAEL FERNANDES MEIRELLES ARAÚJO**

Dissertação apresentada ao Centro de Ciência e Tecnologia, da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção de título de Mestre em Engenharia e Ciência dos Materiais.

Orientador: Prof. D.Sc. Angelus Giuseppe Pereira da Silva

CAMPOS DOS GOYTACAZES – RJ

MARÇO – 2015

DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÕES LIVRES DE  
MARCADORES UTILIZANDO SENSORES DE PROFUNDIDADE

**ÉLISSON MICHAEL FERNANDES MEIRELLES ARAÚJO**

Dissertação apresentada ao Centro de  
Ciência e Tecnologia, da Universidade Estadual do  
Norte Fluminense, como parte das exigências para  
obtenção de título de Mestre em Engenharia e  
Ciência dos Materiais.

Aprovada em \_\_\_/\_\_\_/\_\_\_\_\_.

Comissão Examinadora:

---

Prof. Ítalo de Oliveira Matias (D.Sc., Sistemas Computacionais) – UCAM

---

Prof. Luis Antonio Rivera Escriba (D.Sc., Informática) – UENF

---

Prof.<sup>a</sup> Annabell Del Real Tamariz (D.Sc., Engenharia Elétrica) – UENF

---

Prof. Marcelo Neves Barreto (D.Sc., Petróleo e Energia) – IFF

---

Prof. Angelus Giuseppe Pereira da Silva (D.Sc., Ciências Técnicas) – UENF

Orientador

## **AGRADECIMENTOS**

À Universidade Estadual do Norte Fluminense Darcy Ribeiro e ao programa de Pós-Graduação em Engenharia e Ciência dos Materiais, por proporcionarem um ensino gratuito de qualidade;

Aos professores Angelus Giuseppe e Ítalo Matias, pela orientação, atenção e apoio na condução deste trabalho;

Aos professores Luís Rivera e Annabell Tamariz, responsáveis pela minha base em computação, necessária neste projeto;

Ao professor Marcelo Neves e membros da banca, pelas suas contribuições;

À minha família, a minha mãe e, em especial, meus irmãos Krishynan Shanty e Brandon Nyorai por terem servido de “cobaias” no protótipo desse projeto;

Aos meus amigos Felipe Barros e Luciano por também terem participado da gravação do vídeo de apresentação do protótipo;

À minha namorada Aline, pela participação e por tudo de bom que ela representa na minha vida.

## RESUMO

Modelos digitais em três dimensões de objetos do mundo real são importantes em muitas aplicações como, por exemplo, prototipagem virtual, jogos digitais, efeitos especiais em filmes e design 3D. A tarefa de modelagem desses objetos geralmente cabe a artistas altamente treinados que utilizam quase sempre algum software específico para modelagem 3D. No entanto, os últimos avanços nas técnicas de escaneamento mais acessíveis através de sensores 3D como, por exemplo, o Kinect, tornaram possível que objetos no mundo real sejam identificados, segmentados e medidos em tempo real, sem a necessidade de marcadores ou outros recursos especiais. Esse trabalho apresenta um provador de roupas virtual que usa as medidas extraídas do usuário em frente ao sensor, para projetar, através de realidade aumentada, roupas no usuário, como se ele as estivesse vestindo. Para isso, é necessário reconhecer o usuário que deve ser rastreado no campo de visão do sensor, identificar as partes do corpo que precisam ser medidas e, com essas medidas, projetar um modelo de roupa 3D previamente criado na parte do corpo correta do usuário, que se visualiza em um monitor através de uma câmera, como se fosse um espelho. Recursos como física interativa de tecidos nos modelos 3D das roupas são utilizados com o intuito de fazer o ambiente virtual o mais imersivo possível.

Palavras chave: sensores 3D, computação gráfica, realidade aumentada.

## **ABSTRACT**

Three-dimensional digital models of real world objects are important in many applications such as, for instance, virtual prototyping, digital games, special effects in movies and 3D design. The task of modeling these objects generally lies to highly trained artists that usually use some specific software for 3D modeling. However, the recent advances in scanning techniques through more accessible 3D sensors like, for example, Kinect, made possible that real world objects be identified, segmented and measured in real time without the need of markers or other special means. This paper presents a virtual fitting room that uses the extracted measures from the user in front of the sensor to project, through augmented reality, clothes on the user as if he were wearing them. To do this, it is necessary to recognize the user who should be tracked in the sensor's field of view, identify the parts of the body that need to be measured and with these measurements project a 3D model of clothing created previously in the correct body part of the of the user, which sees himself on a display like a mirror. Features such as physical interactive 3D models of tissues on the clothes is used in order to make the most immersive virtual environment possible.

Key words: 3D sensors, computer graphic, augmented reality

## LISTA DE FIGURAS

Figura 2-1 Matrizes de uma imagem <i>bitmap</i> .....	5
Figura 2-2 Tons de Cinza .....	5
Figura 2-3 Zoom em imagem e valores de pixels coloridos.....	6
Figura 2-4 Câmera estéreo.....	7
Figura 2-5 Profundidade calculada por estereoscopia.....	7
Figura 2-6 Câmera ToF .....	8
Figura 2-7 Fluxo de uma câmera baseada em luz estruturada infravermelho ..	9
Figura 2-8 Efeito da luz solar no padrão do Kinect .....	10
Figura 2-9 Sensores baseados em luz infravermelho estruturada.....	11
Figura 2-10 Luz infravermelha estruturada e mapa de profundidade .....	12
Figura 2-11 Visualização 3D de um mapa de profundidade .....	12
Figura 2-12 Processo de estimação das posições das articulações.....	13
Figura 2-13 Articulações estimadas de um mapa de profundidade .....	14
Figura 2-14 Exemplo de realidade aumentada .....	15
Figura 2-15 Diferentes marcadores para realidade aumentada.....	16
Figura 2-16 O problema da sobreposição.....	17
Figura 2-17 Fotos de amostra de Provedores Virtuais .....	20
Figura 3-1 Reconstrução 3D de uma cena real .....	22
Figura 3-2 Distância entre os pontos através do teorema de Pitágoras .....	26
Figura 3-3 Pontos escolhidos para medição de largura e altura.....	27
Figura 4-1 Metade da parte da frente do modelo de uma camisa .....	29
Figura 4-2 Aplicação do espelhamento para simetria vertical.....	29
Figura 4-3 Replicação e reorientação da parte das costas.....	31
Figura 4-4 Costura e adaptação dos polígonos ao manequim .....	31
Figura 4-5 Variação na contagem de triângulos de modelos 3D .....	32
Figura 4-6 Escolha dos pontos para criação do esqueleto .....	33
Figura 4-7 Visualização do esqueleto finalizado .....	34
Figura 4-8 Mapa de peso da perna direita .....	35
Figura 4-9 Posicionamento da câmera e espelho na cena .....	36
Figura 4-10 Orientação incorreta da luz direcionada .....	37
Figura 4-11 Direcionamento da luz escolhida.....	37

Figura 4-12 Algumas imagens utilizadas como texturas das vestimentas .....	38
Figura 4-13 Visão Perspectiva e Ortográfica .....	38
Figura 4-14 Comparação entre o modelo de bermuda mais folgado e justo ..	39
Fonte: (Próprio autor)Figura 4-15 Roupas disponibilizadas no provador.....	39
Figura 4-16 Configuração do componente Skinned Cloth .....	41
Figura 4-17 Situação de múltiplos usuário em frente ao sensor .....	43
Figura 4-18 Botões seletores de roupa usados na interface.....	44
Figura 4-19 Linhas de colisão entre as mãos do usuário e os botões .....	44
Figura 5-1 1º Modelo de Camisa – Testando ajuste de tamanho .....	45
Figura 5-2 1º Modelo de Camisa - Testando movimentação .....	45
Figura 5-3 Modelo de camisa com física de tecidos configurada .....	46
Figura 5-4 Testando a física de tecidos .....	46
Figura 5-5 Testando modelo com mangas longas.....	47
Figura 5-6 Testando rotação e movimentação do modelo.....	47
Figura 5-7 Testando o modelo de bermuda com física.....	48
Figura 5-8 Dificuldade de alinhamento em roupas mais justas.....	49
Figura 5-9 Usuário vestindo roupas reais muito folgadas .....	49
Figura 6-1 Imagem de contorno do usuário .....	51

# SUMÁRIO

<b>ÉLISSON MICHAEL FERNANDES MEIRELLES ARAÚJO</b> .....	iii
LISTA DE FIGURAS .....	5
CAPÍTULO 1 .....	1
INTRODUÇÃO .....	1
1.1 – Objetivo .....	2
1.2 – Justificativa .....	2
1.3 – Organização .....	3
CAPÍTULO 2 .....	4
FUNDAMENTAÇÃO TEÓRICA .....	4
2.1 – Visão Computacional .....	4
2.2 – Imagem Digital .....	4
2.3 – Câmeras de profundidade .....	6
2.3.1 – Câmera Estéreo .....	6
2.3.2 – Câmera ToF .....	8
2.3.2 – Câmera baseada em luz estruturada .....	8
2.4 – Kinect .....	10
2.5 – Mapa de Profundidade .....	11
2.6 – Utilidade de um mapa de profundidade .....	13
2.7 – Realidade aumentada .....	15
2.8 – Trabalhos relacionados .....	16
• KinectShop .....	18
• Bodymetrics .....	18
• Fitnect .....	19
• TryLive Eyewear .....	19
CAPÍTULO 3 .....	21

METODOLOGIA .....	21
3.1 – Descrição Básica do Sistema .....	21
3.2 – Ferramentas utilizadas .....	21
3.2.1 – SDK da Microsoft.....	22
3.2.2 – OpenNI .....	23
3.2.3 – Unity3D.....	24
3.2.4 – Marvelous Designer.....	24
3.2.5 – Maya.....	25
3.3 – Fluxo do programa .....	25
3.3.1 – Extração das medidas .....	26
3.3.2 – Interface do programa .....	27
3.3.3 – Alinhamento das roupas virtuais.....	27
CAPÍTULO 4.....	28
DESENVOLVIMENTO .....	28
4.1 – Criação das vestimentas 3D.....	28
4.2 – Deformação das roupas de acordo com movimentos do usuário.....	32
4.3 – Desenvolvimento da cena de realidade aumentada.....	35
4.4 – Lógica de programação.....	41
4.5 – Interface de interação.....	43
CAPÍTULO 5.....	45
RESULTADOS .....	45
CAPÍTULO 6.....	50
CONCLUSÃO .....	50
1 – Trabalho Futuro.....	50
REFERÊNCIAS BIBLIOGRÁFICAS .....	52

## CAPÍTULO 1

### INTRODUÇÃO

Nos últimos anos, o desenvolvimento dos sensores de profundidade através de luz estruturada revolucionou a interação com as máquinas em direção a um caminho intuitivo, chamado de NUI (*Natural User Interface*), ou seja, uma ‘interface natural’, que substitui o mouse e teclado, periféricos que durante anos foram os principais meios de entrada de informação nos computadores.

Esses sensores possibilitaram que tarefas como segmentação de objetos em relação ao resto de uma cena fossem facilmente realizadas, resolvendo, de forma rápida, problemas complexos na área de processamento de imagens. Em conjunto com novos algoritmos, também permitiram que gestos com as mãos e comandos envolvendo linguagem corporal fossem mais facilmente traduzidos.

Outro motivo pelo qual eles fizeram sucesso foi o preço, eles são muito mais baratos do que os dispositivos de escaneamento 3D anteriores, de forma que seja acessível para usuários convencionais. O que facilitou a comercialização em massa.

O Kinect, sensor de profundidade desenvolvido pela Microsoft em conjunto com a PrimeSense, custa atualmente US\$ 99.99, na loja online da Microsoft (Microsoft Store, 2015), câmeras ToF (*Time of Flight*), por exemplo, custavam aproximadamente mais de dez vezes esse valor quando esse projeto teve início.

Os sensores de profundidade abriram portas para o desenvolvimento de, por exemplo, sistemas de reconhecimento de ambientes ao redor de máquinas em robótica e automação, interação e controle de diversos dispositivos através de gestos, escaneamento de objetos do mundo real para virtualização e utilização em simuladores 3D, entre outras.

O trabalho possui como foco o desenvolvimento de um protótipo de provador virtual de roupas. O protótipo, que representa um espelho onde o usuário se visualiza com roupas virtuais, faz uso tecnologias de segmentação e detecção de objetos, reconhecimento de gestos, extração de medidas e realidade aumentada. Tecnologias que poderiam ser empregadas em conjunto ou separadamente nos exemplos de sistemas mencionados anteriormente.

## 1.1 – Objetivo

O objetivo desse trabalho é extrair as medidas de um usuário em frente a uma câmera e fazer uso delas para criar um provador de roupas virtual. Nele, os trajes serão projetados com o uso de realidade aumentada. O usuário estará visualizando a si mesmo em um monitor, de forma espelhada para simular um “espelho virtual”.

O usuário será capaz de interagir através de gestos com o sistema, escolhendo um modelo de roupa que, através do espelho virtual, fará com que possa conferir como aquela roupa ficaria ‘vestida’ nele. Os objetivos específicos são:

- Identificação do usuário que será medido dentro do campo de visão do sensor;
- Reconhecimento das partes do corpo do usuário em que diferentes roupas serão projetadas;
- Extração das medidas relevantes do corpo desse usuário;
- Exportação dessas informações para o software responsável pela projeção do vestuário virtual;
- Criação dos modelos 3D das roupas;
- Sobreposição da roupa virtual no usuário;
- Configuração da componente responsável pela deformação física em tempo real nos modelos de roupa virtuais;
- Programação da interface de seleção dos trajes;

## 1.2 – Justificativa

Segundo Weiss (Weiss *et al.*, 2011), sistemas comerciais de escaneamento custam entre US\$ 35,000 e US\$ 500,000 e o tamanho e custo desses *scanners* os tornam inviáveis em muitas potenciais aplicações.

Um sistema que permita escanear medidas cria oportunidades para um grande número de novas aplicações, além disso, um provador virtual, faria com que compras de vestuários pela internet fossem realizadas de forma muito mais eficiente, uma vez que as medidas do cliente seriam extraídas, o tamanho de roupa mais

adequado dentro do estoque seria exibido e o cliente poderia tomar melhores decisões através do espelho virtual que simularia ele vestindo a roupa.

Esta tecnologia também permite o desenvolvimento de aplicativos de controle de máquinas através de gestos, bem como o escaneamento de objetos e cenas, podendo ser aplicada em setores como prototipagem rápida, treinamento de pessoal, inspeção, automação, simulação, entre outras.

### **1.3 – Organização**

Além da Introdução, descrita anteriormente, o capítulo 2 descreve os fundamentos teóricos das componentes principais deste trabalho, explicando os conceitos de imagem digital, como funcionam os diferentes sensores de profundidade, o que é realidade aumentada e um resumo de alguns trabalhos desenvolvidos no mesmo segmento.

No capítulo 3, as ferramentas, *softwares* e tecnologias utilizadas no desenvolvimento do provador virtual são explicados em detalhes. Entre eles estão os softwares Marvelous Designer, Maya, Unity3D e as bibliotecas para desenvolvimento utilizando os sensores de profundidade.

O capítulo 4 descreve em detalhes o processo de implementação de cada etapa do sistema, começando pela criação dos modelos virtuais das roupas, seguindo para a preparação delas para que possam se deformar e ajustar aos usuários.

Os resultados e conclusões são expostos nos capítulos 5 e 6, respectivamente. Nestes, são exibidas algumas imagens de usuários testando o protótipo. Algumas limitações são descritas e ilustradas, assim como algumas sugestões para melhorias do sistema.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

Esse capítulo tem como objetivo explicar os principais conceitos teóricos necessários para compreensão do desenvolvimento do protótipo desenvolvido nesse projeto.

#### 2.1 – Visão Computacional

Visão computacional é o ramo da computação que trabalha com métodos para processar, analisar e extrair informações de imagens digitais com o objetivo de construir sistemas de computadores que entendam essas imagens. Nesse campo da computação, as propriedades do mundo, como forma, iluminação e cor são reconstruídas virtualmente. (Bradski & Kaehler, 2008)

Hoje, visão computacional é utilizada em diversas aplicações, que incluem reconhecimento de escrita, inspeção de máquinas, construção de objetos 3D, análise de imagens de exames médicos, segurança em automóveis, efeitos especiais em filmes, captura de movimentos para aplicações em animações 3D, segurança, reconhecimento de digitais, entre outras. (Szeliski, 2010)

#### 2.2 – Imagem Digital

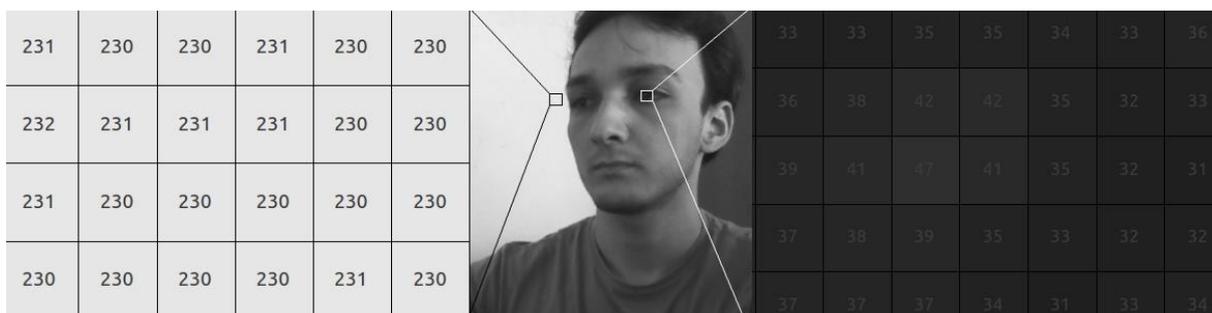
Uma imagem bidimensional é representada digitalmente por números binários codificados de forma que seja possível armazenar, transferir, reproduzir através de dispositivos eletrônicos. Existem dois tipos fundamentais de imagem digital, uma é do tipo vetorial, que é descrita por posição e tamanho de formas como linhas, curvas, círculos e retângulos e a outra é do tipo rastreio, ou *bitmap*, que apresenta uma matriz de pixels que correspondem ponto a ponto à imagem que está representando. Imagens vetoriais geralmente são utilizadas em logos, fontes e ilustrações. A qualidade delas não é afetada, quando a imagem é aumentada.

Imagens *bitmap* são a forma mais comum de representar imagens ricas em detalhes.

Os pixels assumem valores numéricos de acordo com o brilho dentro de um conjunto finito. Em uma imagem monocromática, por exemplo, só há uma cor representando a imagem, em seus diferentes tons. Se os pixels possuírem formato de inteiros de 8 bits, o brilho poderá ser quantizado em apenas 256 ( $2^8$ ) valores discretos. (Gonzalez & Woods, 2002)

As Figuras 2-1 e 2-2 mostram que valores mais próximos de 0 (completamente preto) representam pixels mais escuros enquanto pixels mais claros assumem valores próximos de 255 (totalmente branco).

**Figura 2-1 Matrizes de uma imagem *bitmap***



**Exemplo de matrizes de valores dos pixels em uma imagem *bitmap* em tons de cinza.  
Fonte: (Próprio autor)**

**Figura 2-2 Tons de Cinza**

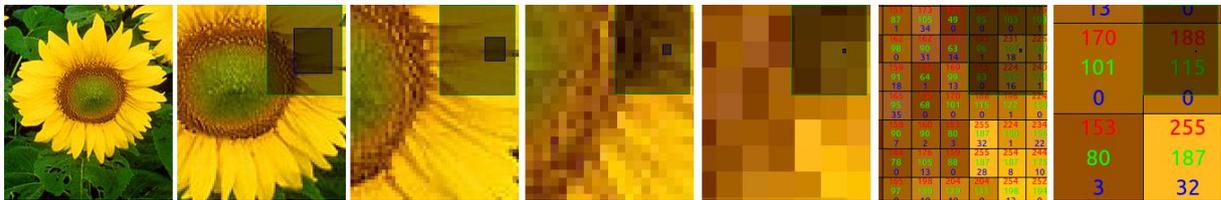


**Imagem de 256 pixels de largura com efeito em degradê gerado pelo software de edição de imagens GIMP. Fonte: (Próprio autor)**

Uma imagem colorida é tipicamente uma tripla de cores decompostas. Um modelo comum é o RGB, onde a imagem é decomposta nas cores, vermelho, verde e azul respectivamente, estabelecendo um valor entre o máximo possível de reprodução daquela cor e o mínimo, ou seja, a ausência total dela. Sendo que a

soma dos valores resulta em um ponto colorido na imagem final. A Figura 2-3 ilustra um zoom em uma determinada parte de uma imagem colorida até a visualização dos valores de seus pixels.

Figura 2-3 Zoom em imagem e valores de pixels coloridos



Fonte: (Próprio autor)

Esses números que representam a imagem digital muitas vezes revelam pouca informação ou contêm ruídos e distorções. Por esse motivo, sistemas baseados em visão computacional precisam, na prática, de outras informações descritivas do ambiente, ou outros sensores para solucionar as limitações impostas pelo dispositivo de captura. (Bradski & Kaehler, 2008)

Um exemplo de limitação comum é o problema de calcular a distância entre os objetos da cena em relação à câmera, criando assim um mapa de profundidade. Diferentes técnicas e sensores são utilizados para resolver esse problema.

## 2.3 – Câmeras de profundidade

O dispositivo de captura que produz a imagem ou mapa de profundidade costuma ser chamado de câmera de profundidade ou câmera 3D. O mapa de profundidade também pode ser chamado de mapa de disparidade.

### 2.3.1 – Câmera Estéreo

Com duas câmeras alinhadas horizontalmente, ou com uma câmera estéreo (Figura 2-4), é possível estimar a profundidade de um ponto a partir do triângulo (ilustrado na Figura 2-5) formado entre esse mesmo ponto e as duas lentes,

simulando assim um fenômeno natural chamado estereoscopia, que ocorre com o ser humano quando uma cena qualquer é observada com os dois olhos.

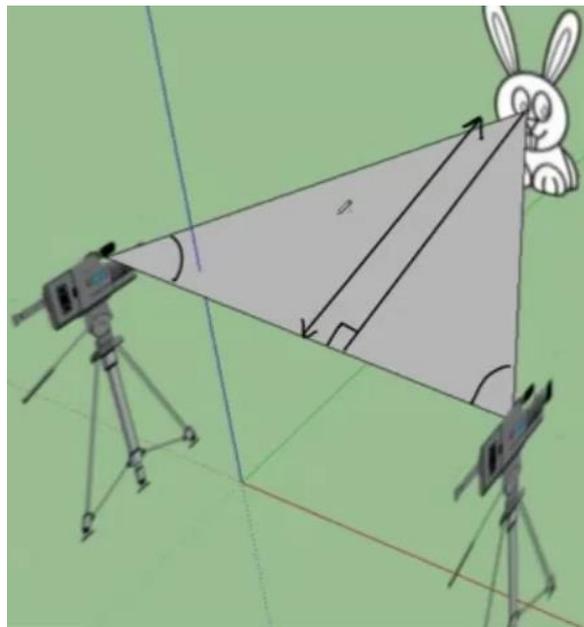
**Figura 2-4 Câmera estéreo**



**Exemplo de uma câmera estéreo, a Bumblebee.**

Para comparar as imagens, é necessário um processo complexo de calibração, no qual é necessária uma amostra de pontos visíveis e corretamente correspondidos no espaço de interseção dos diferentes campos de visão, definidos pela posição e orientação das câmeras. (Loaiza *et al.*, 2010)

**Figura 2-5 Profundidade calculada por estereoscopia**



**Estimativa de profundidade a partir de duas lentes calibradas em relação a um mesmo ponto.  
Fonte: (CuriousInventor, 2015)**

### 2.3.2 – Câmera ToF

A profundidade de um ponto na cena é calculada por uma câmera ToF (Figura 2-6) a partir do tempo que a luz emitida pela câmera viaja até o destino e retorna ao sensor receptor. Essas câmeras geram imagens com resolução muito baixas, mas a uma alta taxa de quadros por segundos.

Figura 2-6 Câmera ToF



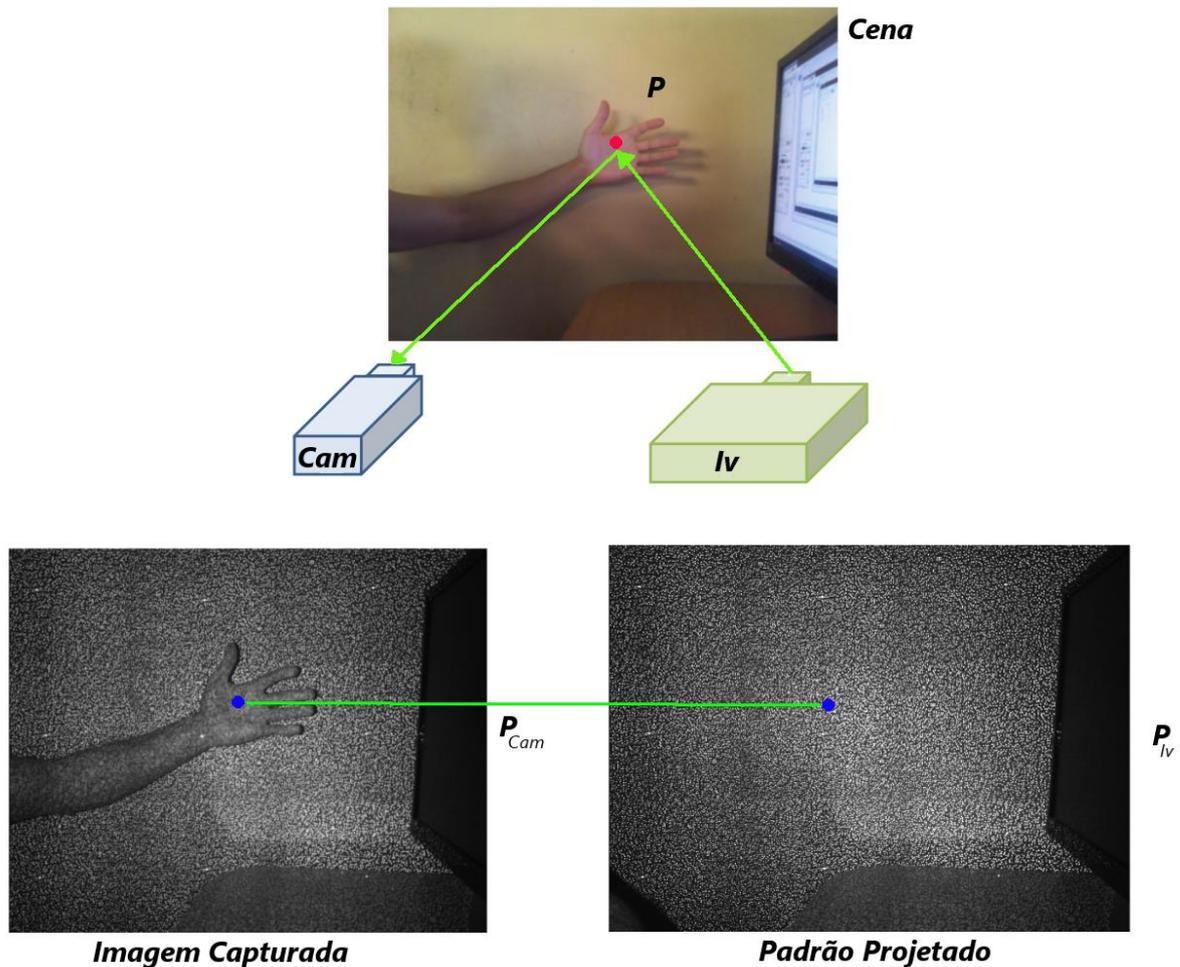
Exemplo de uma câmera ToF da Panasonic.

### 2.3.2 – Câmera baseada em luz estruturada

Esse tipo de câmera possui um projetor que ilumina a cena com um padrão de luz infravermelha estruturada. A câmera, alinhada horizontalmente em relação ao projetor, captura esses pontos que não são visíveis ao olho humano. Quando um objeto entra na cena, ele distorce o padrão desses pontos e essa variação é reconhecida como mais próximo ou distante (Figura 2-7).

Essas câmeras possuem hardware e software específicos para efetuarem a tradução e correspondência da distância dos pontos do padrão que emitem em tempo real.

Figura 2-7 Fluxo de uma câmera baseada em luz estruturada infravermelho



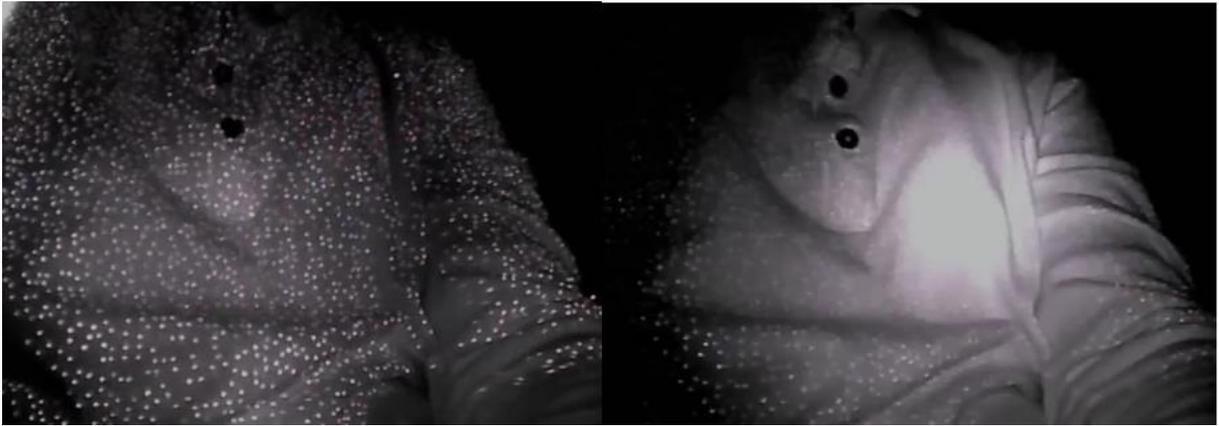
O pixel  $P_{Iv}$  é codificado no padrão que é projetado na cena e obtido pela Câmera(Cam). O ponto 3D associado à  $P_{Iv}$  é  $P$  e o ponto correspondente é  $P_{Cam}$ . Um algoritmo de correspondência estima os pontos conjugados (Mutto *et al.*) Fonte: (Próprio autor)

Por utilizarem infravermelho, são considerados dispositivos para serem utilizados dentro de ambientes fechados. A luz do sol, por exemplo, poderia 'queimar' os pontos, deixando um conjunto deles muito claro, comprometendo assim a leitura da distância da área 'queimada'. Esse problema pode ser conferido na Figura 2-8.

Outro problema é que materiais transparentes, reflexivos e absorventes não refletem bem a luz infravermelha. (Alhwarin, Ferrein, & Scholl, 2014)

O custo desse tipo de câmera é muito menor do que os tipos descritos anteriormente.

Figura 2-8 Efeito da luz solar no padrão do Kinect



A imagem da direita mostra como a luz solar compromete a leitura do sensor.  
Fonte: (Curiousinventor, 2015)

## 2.4 – Kinect

O Kinect é um sensor baseado em luz estruturada, desenvolvido pela PrimeSense, em conjunto com a Microsoft em novembro de 2010, como um acessório para o console Microsoft Xbox 360.

O principal objetivo do acessório era ser uma ferramenta de interação entre o usuário e o console através de gestos e comandos de voz.

As diferenças entre o Kinect e outros sensores baseados em luz estruturada são que o Kinect contém:

- Uma câmera RGB;
- Um conjunto de microfones;
- Um acelerômetro, que diz ao sistema qual o lado que está virado para baixo de acordo com a gravidade;
- Uma base motorizada que permite que o sensor vire 58 graus na horizontal, 45 na vertical, 70 graus na diagonal;
- Um SDK (*System Development Kit*)<sup>1</sup> oficial da Microsoft.

---

<sup>1</sup> O kit de desenvolvimento da Microsoft (SDK – *System Development Kit*), é uma biblioteca que traz diversas funções prontas para o Kinect, agilizando assim, o processo de desenvolvimento de sistemas que utilizem o sensor.

A Figura 2-9 mostra o sensor Kinect ao lado de outros sensores do mesmo tipo.

**Figura 2-9 Sensores baseados em luz infravermelho estruturada**



**O Microsoft Kinect, Asus Xtion e PrimeSense Carmine respectivamente. Fonte: (Próprio autor)**

O sensor captura imagens em uma resolução de 640 por 480 pixels a 30 fps (*frames* por segundo). Cada pixel gerado pelo Kinect possui informação de cor e profundidade da cena, ou seja, ele é capaz de gerar informação visual e geométrica e ambas podem se complementar. Esse tipo de estrutura de imagem é denominada RGBD (RGB + Depth). Os pixels de profundidade são formatados em 11 bits, ou seja, a sensibilidade é de 2048 níveis de distância. Já na câmera RGB os pixels são formatados em 8 bit por canal. A distância de operação é de 0,40 metros a 3,5 metros. As informações geradas pelo periférico consomem aproximadamente 70% da taxa de transmissão da porta USB 2.0 que ele estiver usando. (Cruz *et al.*, 2012)

A estrutura de imagem RGBD permite implementar procedimentos que são difíceis, senão impossíveis, utilizando apenas cores. Dentre as diversas novas possibilidades, um exemplo importante é o rastreamento em tempo real do esqueleto humano, um modelo usado em interações baseadas em gestos, nos jogos lançados para o Kinect

## **2.5 – Mapa de Profundidade**

Um mapa de profundidade, (também chamado de imagem de disparidade), é uma imagem ou canal que contém informações referentes à distância das superfícies dos objetos, de acordo com um ponto de vista em uma cena. É possível traduzi-lo em uma imagem em tons de cinza, por exemplo, onde pixels mais brancos representam pixels de objetos que estão mais próximos e pixels mais escuros,

objetos que estão mais distantes da câmera. Um mapa de profundidade de 8 bits possibilitaria representar um total de 256 diferentes distâncias (Figura 2-10).

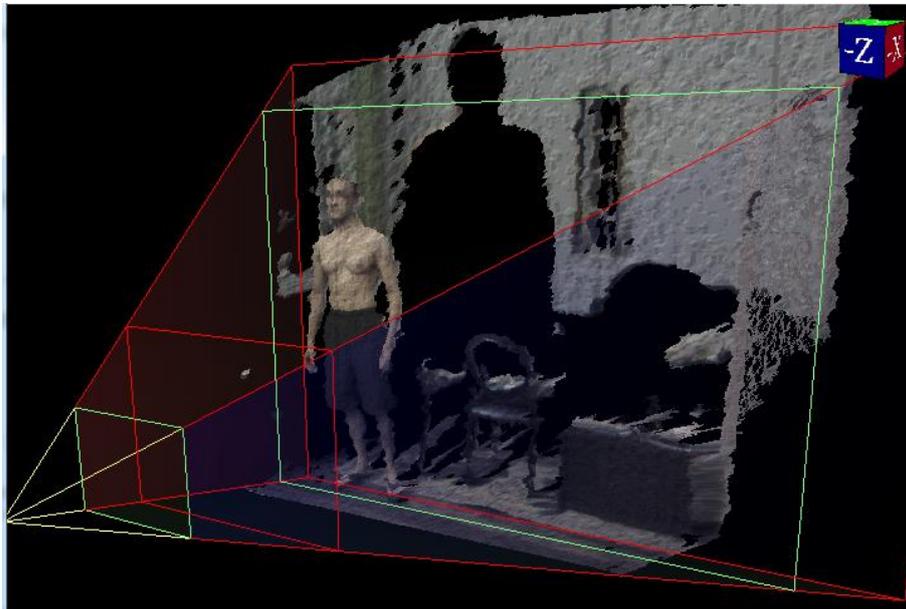
**Figura 2-10 Luz infravermelha estruturada e mapa de profundidade**



**Exemplo de mapa de profundidade obtido através de luz infravermelha estruturada. Na esquerda a imagem da câmera RGB, no centro a imagem recebida da câmera infravermelho e o mapa de profundidade na direita. Fonte: (Próprio autor)**

O mapa de profundidade também pode ser exibido dentro de um espaço tridimensional, o resultado é ilustrado na Figura 2-11.

**Figura 2-11 Visualização 3D de um mapa de profundidade**



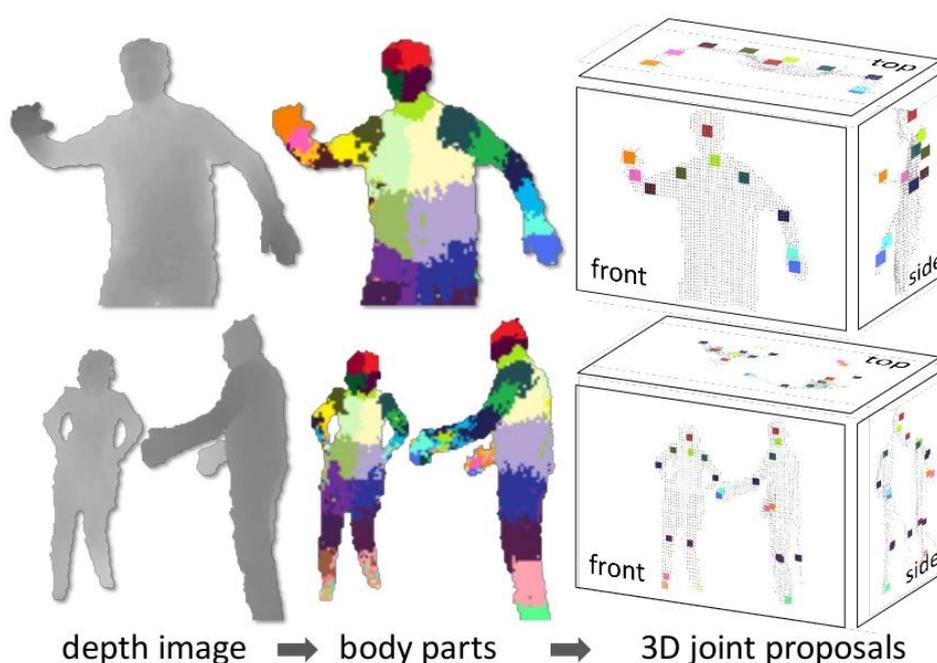
**As manchas pretas no fundo são os pontos cegos do sensor. Fonte: (Próprio autor)**

## 2.6 – Utilidade de um mapa de profundidade

A partir de um mapa de profundidade, é possível estabelecer posições de certas partes do corpo de um usuário detectado em frente ao sensor. Isso é possível com um algoritmo desenvolvido por Fitzgibbon (Fitzgibbon, et al., 2013). Neste trabalho, tendo como entrada a informação de profundidade adquirida pelo Kinect, regiões especiais do corpo são reconhecidas e agrupadas em pixels de acordo com uma probabilidade baseada em um banco de dados de aproximadamente 500 mil imagens. As imagens no banco de dados incluem pessoas de todas as alturas, medidas e peso executando sequências de ações, como por exemplo, dançar, chutar, correr, navegar em menus, etc

Posições de pontos de articulações são definidas baseadas nesses grupos, utilizando florestas de decisões como classificadores. O algoritmo efetua o reconhecimento em cada frame independentemente do frame anterior e é executado em 5 ms por frame (200 frames por segundo) na placa de vídeo do Xbox 360, um exemplo de hardware barato e comum atualmente. A Figura 2-12 ilustra a ideia principal por trás do trabalho de Fitzgibbon .

Figura 2-12 Processo de estimação das posições das articulações



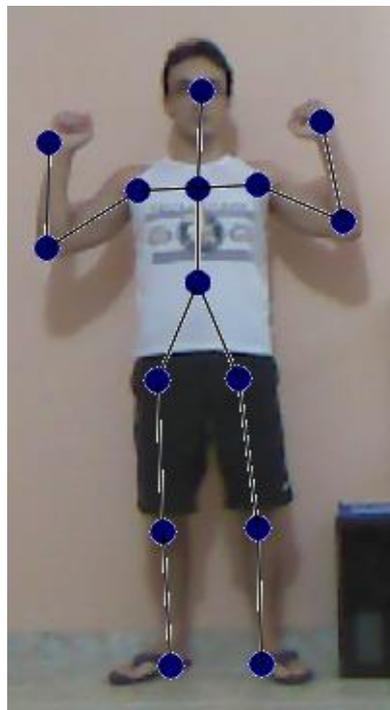
Diferentes etapas no processo de estimação das posições 3D de articulações a partir de mapas de profundidade do corpo humano. Fonte: (Fitzgibbon, et al., 2013)

Comparado a outros métodos, o trabalho de Fitzgibbon se tornou estado de arte, mostrando resultados eficientes e estáveis em tempo real e é a parte mais importante do Kinect, pois permitiu que jogos, por exemplo, de esportes, onde é necessário que comandos sejam reconhecidos rapidamente, pudessem ser lançados para o console.

A localização desses pontos terá um papel importante na interação entre os modelos virtuais de roupa com o usuário no provador desenvolvido nesse projeto. Eles auxiliarão no processo de fixação da projeção de cada tipo de roupa virtual na parte correta do corpo em que ela deve ser 'vestida' pelo usuário. A Figura 2-13 mostra a localização de articulações em formato de pontos sobrepostos em cima da imagem obtida pela câmera RGB do Kinect.

É comum se referir a esses pontos como articulações e o conjunto de deles como esqueleto do usuário. Esses pontos substituirão os marcadores necessários para a realidade aumentada do provador, explicada na sessão 2.7.

**Figura 2-13** Articulações estimadas de um mapa de profundidade



**Posição de alguns pontos de articulações estimados através de um mapa de profundidade. OpenCV foi usado para desenhar os pontos e retas sobre a imagem original. Fonte: (Próprio autor)**

## 2.7 – Realidade aumentada

Um sistema de realidade aumentada ‘amplia’ o mundo real com objetos virtuais, ou seja, criados por computador. A aparência para o usuário é a de que os objetos coexistem no mesmo espaço no mundo real. Esse tipo de sistema poderia ser definido através das seguintes propriedades, segundo Azuma (Azuma, *et al.*, 2001):

- Combina objetos reais e virtuais em um mesmo ambiente real;
- É interativo e roda em tempo real;
- Alinha (sincroniza) objetos reais e virtuais uns com os outros.

Outros pesquisadores expandiram essa definição, indo além dessa visão.

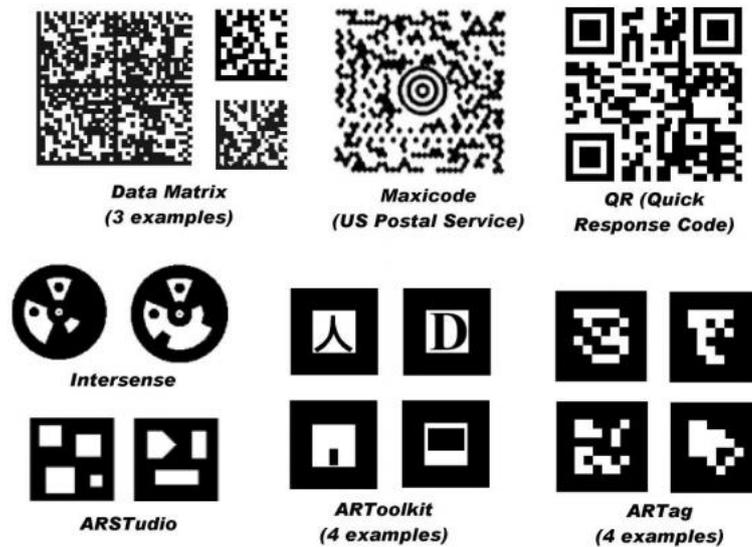
A Figura 2-14 mostra uma situação onde um modelo de árvore tridimensional virtual é projetado em uma cena no mundo real. Um cartão é utilizado como marcador para identificação do local onde o objeto 3D deve ser posicionado. A Figura 2-15 exhibe exemplos de diferentes tipos de marcadores.

Figura 2-14 Exemplo de realidade aumentada



A árvore é um objeto virtual sobreposto no mundo real através da localização do cartão marcador. Fonte: (PS Vita, 2013)

Figura 2-15 Diferentes marcadores para realidade aumentada



Fonte: (Fiala, 2005)

O provador desenvolvido nesse projeto utiliza os pontos de articulações do esqueleto, mencionado na sessão 2.6, como localização para projeção dos objetos virtuais, nesse caso os modelos de roupas 3D, substituindo os marcadores. O sistema pode ser caracterizado como livre de marcadores por esse fato.

## 2.8 – Trabalhos relacionados

Presle (Presle, 2012) utilizou a engine Unity para sobrepor modelos 3D de camisas, calças e saias criados no software de modelagem Cinema 4D. A *engine*<sup>2</sup> também foi responsável pelo cálculo das deformações físicas do vestuário virtual. Para detecção e rastreamento do usuário, o Kinect foi escolhido como sensor de profundidade. A biblioteca usada foi a primeira versão do OpenNI. Outra funcionalidade adicionada foi a de substituir o fundo do vídeo por uma imagem.

Um problema comumente encontrado em provadores virtuais é o da sobreposição. Uma vez que a imagem da câmera é exibida em um plano, os modelos de roupa 3D ficam na frente desse plano bloqueando algumas vezes

<sup>2</sup> *Engine* é um programa ou sistema que executa tarefas principais ou funções essenciais para outros programas. Costuma ser o programa base que coordena as funções utilizadas por outros programas desenvolvidos através dele.

incorretamente partes do corpo do usuário. A Figura 2-16 mostra esse problema que não foi resolvido no trabalho de Presle.

**Figura 2-16 O problema da sobreposição**



**A mão do usuário é ocultada pela projeção do modelo 3D de saia. Fonte: (Presle, 2012)**

Isikdogan e Kara (Isikdogan & Kara, 2012) desenvolveram um provador virtual usando o sensor Kinect, o SDK da Microsoft e fotos frontais em duas dimensões de camisas. A imagem da camisa é sobreposta no vídeo obtido pela câmera RGB do Kinect. Também foi utilizada a segmentação da cor da pele, de forma a resolver o problema da sobreposição mencionado anteriormente. Os resultados apresentam alguns lugares onde a pele não foi reconhecida.

A câmera RGB do Kinect oferece uma resolução de 480 pixels, o que é uma resolução baixa para os padrões de hoje em dia. A grande maioria dos aparelhos de televisões atualmente possuem resolução HD (720p) ou Full HD (1080p). Para solucionar esse problema Giovanni e sua equipe (Giovanni *et al.*, 2012) desenvolveram um sistema onde o espelho virtual do provador utiliza uma outra camera HD em conjunto com o sensor Kinect. Portanto foi necessário um trabalho de calibração entre os dois dispositivos.

No trabalho de Giovanni, eles também testaram a eficácia do reconhecimento das articulações do esqueleto entre o OpenNI e o SDK da Microsoft. Os resultados mostraram que o SDK da Microsoft confunde os membros do lado direito e esquerdo

do corpo, quando o usuário está de costas para o sensor. Isso não aconteceu durante os testes com o OpenNI.

Também é importante citar que a inclusão de novos modelos de trajes exige um trabalho intenso e isso foi descrito como uma limitação do sistema. Outra limitação mencionada foi que as roupas virtuais são escaladas uniformemente de acordo com a distância do usuário em relação ao sensor. Uma pessoa obesa, por exemplo, teria sua largura muito maior do que a altura e uma escala uniforme do vestuário virtual não seria suficiente para cobrir todo o corpo dessa pessoa com a roupa projetada.

Pachoulakis e Kapetanakis (Pachoulakis & Kapetanakis, 2012) escreveram uma avaliação sobre os provedores virtuais existentes, KinectShop, Bodymetrics, Fitnect, VIPodium, Imagine That, TryLive Eyewear e AM4U. Alguns desses trabalhos serviram de inspiração para o provedor desenvolvido nesse projeto. Esses serão descritos a seguir. A Figura 2-17 mostra fotos de alguns desses provedores em funcionamento.

- **KinectShop**

KinectShop é uma plataforma online criada pela equipe Emerging Experiences da Razorfish. Nesse provedor virtual, os usuários podem criar álbuns de fotos com as roupas virtuais e compartilhá-los com os amigos via redes sociais, coletar aprovações ou reprovações e armazenar lista de interesse em smartphones. As redes sociais também podem ser utilizadas para alertar os clientes de promoções dos itens na lista de interesses em lojas próximas.

- **Bodymetrics**

Bodymetrics (Bodymetrics, 2015) em parceria com PrimeSense, criou um scanner 3D de corpo humano composto de oito sensores de profundidade da PrimeSense localizados em diferentes partes dentro de uma cabine. Juntos eles calculam cem medidas do usuário. Posteriormente essas medidas podem ser utilizadas por 'estilistas' para encontrar roupas com o tamanho e formato mais adequado para o cliente. Também criaram uma versão para ser utilizada em casa com apenas um sensor 3D.

Em um dos programas que utiliza as medidas, um sistema de cores foi usado em alguns locais dos trajes onde o tecido ficaria apertado no corpo do usuário, com o intuito de informar que o cliente deveria procurar um tamanho maior daquela roupa.

- **Fitnect**

O provedor virtual Fitnect (Fitnect, 2015) usa física interativa em tempo real nos modelos de roupas 3D para criar uma experiência mais convincente. No entanto, espaços vazios na sobreposição das roupas indicam que as medições não são tão sofisticadas quanto as do Bodymetrics, sugerindo que o aplicativo deve ser utilizado para conferir combinações de roupas e estilos mais do que verificação de tamanhos e formatos precisos.

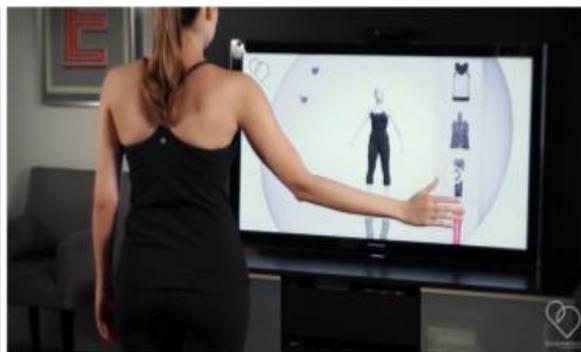
- **TryLive Eyewear**

Desenvolvido por Total Immersion (TotalImmersion, 2015), essa aplicação usa realidade aumentada para fazer o usuário testar joias, relógios, calçados e óculos. É necessário um computador com câmera, um tablet ou smartphone. Um algoritmo de reconhecimento de face detecta, por exemplo, o local onde o óculos virtual deve ser projetado no rosto do usuário. O usuário pode experimentar então diferentes armações e lente de vários ângulos e enviar fotos para redes sociais e receber feedback.

Figura 2-17 Fotos de amostra de Provadores Virtuais



(a) KinectShop from Razorfish Emerging Experiences



(b) Bodymetrics



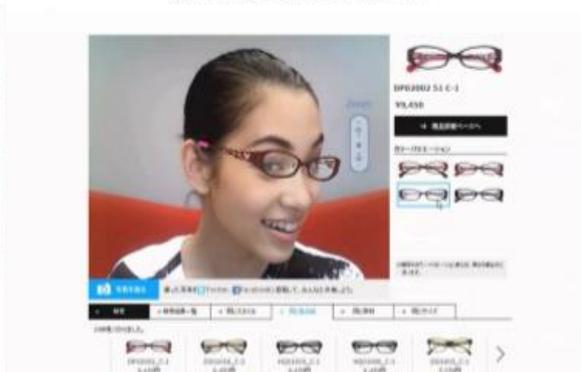
(c) Fitnect



(d) Fitting Reality's VIPodium



(e) Imagine That



(f) Total Immersion's TryLive

Screen Shots dos programas em funcionamento. Fonte: (Pachoulakis & Kapetanakis, 2012)

## CAPÍTULO 3

### METODOLOGIA

Nesse capítulo as ferramentas, *softwares* e tecnologias utilizadas no desenvolvimento do provador virtual são explicados em detalhes. Entre eles estão os softwares Marvelous Designer, Maya, Unity3D e as bibliotecas para desenvolvimento utilizando os sensores de profundidade.

#### 3.1 – Descrição Básica do Sistema

Abaixo são exibidas as partes que compõem o provador de roupas virtual:

- Microsoft Kinect
  - Obtém dados da câmera de vídeo RGB
  - Captura informações de profundidade
- Monitor ou Televisão
  - Exibe o vídeo RGB em modo espelhado
  - Apresenta a interface de comandos de interação
  - Mostra a interface de seleção de roupas
- Computador
  - Executa os algoritmos de rastreamento do esqueleto
  - Sobrepõem os trajes no vídeo exibido
  - Processa a simulação física dos tecidos
  - Sincroniza as informações do projetor infravermelho com o vídeo
  - Armazena os modelos das roupas
  - E qualquer outra funcionalidade relacionada ao software

#### 3.2 – Ferramentas utilizadas

Duas grandes bibliotecas são mais utilizadas atualmente no desenvolvimento de aplicações utilizando o Kinect. São elas o SDK da Microsoft e a OpenNI da

Primensense. Ambas oferecem alguns recursos que diferenciam uma em relação à outra, mas a escolhida para esse projeto foi a da Microsoft.

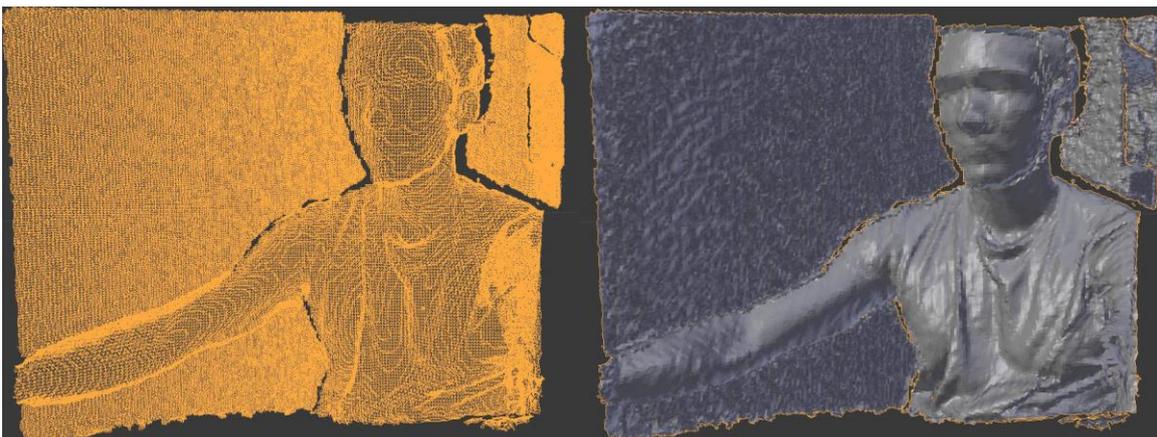
O Unity3D foi escolhido como plataforma para construção do cenário de realidade aumentada por já existir suporte para trocas de informações com o sensor, sendo o maior interesse o de se obter acesso aos pontos de articulações.

Para construção dos modelos virtuais das roupas, foi utilizado o Marvelous Designer 3, em conjunto com Maya 2015, que serviu para a tarefa de mapeamento do esqueleto que é responsável pela movimentação delas.

### 3.2.1 – SDK da Microsoft

As linguagens suportadas pelo SKD da Microsoft (KinectSDK, 2013) são C++, C# ou Visual Basic. Ele possui capacidade de reconhecimento da parte de cima de esqueletos de usuários sentados, reconhecimento de voz, rastreamento de face e reconstrução 3D das cenas como diferencial. A Figura 3-1 exibe o modelo 3D da mesma cena da Figura 2-10. As imperfeições no objeto tridimensional construído são minimizadas se for possível rotacionar o sensor ao redor do objeto de interesse.

Figura 3-1 Reconstrução 3D de uma cena real



Na esquerda malha exportada utilizando o Kinect Fusion, tecnologia de reconstrução 3D do SDK Microsoft e na direita o respectivo modelo 3D. Fonte: (Próprio autor)

A Microsoft também disponibilizou o produto 'Kinect para Windows', que consiste no hardware do sensor Kinect em uma versão que possui um modo de captura que detecta objetos 40 centímetros mais próximos do sensor.

Esse SDK permite acessos a funcionalidade exclusivas do Kinect, além de garantir 100% de compatibilidade com esse sensor. A Microsoft tem atualizado frequentemente, lançando em 21/10/2014 a versão 2.0 do SDK, compatível com o novo Kinect 2, versão do sensor com poder de processamento e resolução da imagem superior (Kinect2, 2015). O interesse em poder substituir mais facilmente o sensor para um de tecnologia mais avançada, torna esse SDK a escolha mais adequada para o desenvolvimento desse projeto.

### 3.2.2 – OpenNI

(OpenNI, 2013) (*Open Natural Interaction*) é um framework multiplataforma de código aberto. Ele pode ser utilizado com qualquer sensor 3D, em qualquer sistema operacional. É possível estender as funcionalidades do SDK através de *middlewares*<sup>3</sup>. Um deles, muito utilizado, permite que gestos e posturas dos usuários sejam reconhecidas, tanto do corpo quanto das mãos. É importante mencionar que frequentemente é necessário apenas rastrear a mão do usuário, que substitui o mouse na hora de mover um cursor, por exemplo. Esse SDK fornece funções específicas para esse tipo de rastreamento.

O SDK permite que vídeos no formato RGBD sejam gravados e executados. Esse tipo de vídeo guarda informações referentes à profundidade de cada pixel na cena e ajudam no desenvolvimento, uma vez que o uso contínuo do sensor é desnecessário e ser possível gravar cenários específicos para testar a aplicação. É interessante ter um vídeo em que o fundo é bem movimentado, ou outro em que há diversos usuários interagindo ao mesmo tempo com o sistema, e poder 'rodar' a mesma situação diversas vezes na hora de encontrar e consertar erros de programação em geral.

---

<sup>3</sup> *Middewares* são programas que fazem a mediação entre software e demais aplicações. Eles mascaram a heterogeneidade entre diferentes protocolos de comunicação, fornecendo um modelo de programação mais produtivo para os programadores de aplicativos.

Algumas dessas melhorias não existiam na SDK da Microsoft no começo do projeto, o que tornava essa escolha mais interessante na época. Porém, no decorrer do projeto, as atualizações que a Microsoft efetuou na SDK trouxeram as funcionalidades de gravação de vídeos, detecção de gestos, entre outras.

### **3.2.3 – Unity3D**

O Unity3D é uma engine gráfica muito utilizada por desenvolvedores de jogos eletrônicos. É possível utilizá-la nos sistemas operacionais Windows e Mac OS. Existe uma versão gratuita, mas a versão paga (Pro) possui algumas funcionalidades extras. Permite que aplicações para dispositivos móveis sejam criadas. Recentemente também permite que as aplicações sejam importadas para sistemas Linux. (Unity3D, 2013)

A engine facilita a criação de cenas virtuais, composta de diversos objetos 3D. Os modelos podem ser importados a partir de diversos formatos de outros softwares responsáveis por modelagem 3D. Além da cena, a engine permite o ajuste do comportamento dos objetos, as propriedades físicas, sons, texturas e todo o fluxo de controle da aplicação.

Outro grande interesse na utilização dessa engine, em relação a outras, é devido a um componente dela chamado 'Tecido Interativo', o qual é de grande relevância em provadores virtuais. Esse componente permite que diversos comportamentos específicos de tecidos possam ser ajustados com a finalidade de se obter uma simulação mais real das roupas virtuais. Um exemplo de comportamento ajustável seria quanto o tecido pode ser esticado antes que se rasgue.

### **3.2.4 – Marvelous Designer**

Marvelous Designer é uma ferramenta profissional para a criação de roupas virtuais, onde é possível desenhar, recortar, costurar os modelos 3D para exportação e utilização em outros softwares posteriormente. (MarvelousDesigner, 2015)

Com esse software é possível adaptar um modelo de roupa a um manequim previamente criado, ou selecionando algum dos manequins padrões que o programa traz (dois modelos de homens, dois de mulheres e uma criança). Isso facilita não só a tarefa de criação dos modelos de roupas, como a capacidade de adaptar um mesmo modelo de roupa a manequins com proporções e medidas diferentes.

Dentro das funcionalidades, é importante mencionar o mapa de calor que exhibe onde a roupa está apertada no manequim, a possibilidade de trazer animações para o manequim de forma a visualizar a simulação da movimentação da roupa criada em tempo real. É possível também configurar a quantidade de polígonos, o tipo de material da roupa (elástico, couro, etc) e parâmetros como capacidade de esticar, dobrar etc.

### **3.2.5 – Maya**

Maya é um software 3D completo, pois oferece um conjunto de recursos para animação computadorizada 3D, modelagem, simulação, composição e renderização em uma plataforma com capacidade de expansão. O software disponibiliza diversas ferramentas avançadas que facilitam o trabalho de, no caso desse projeto, ligar os modelos de roupa 3D a um esqueleto que possa ser movimentado no Unity3D. Durante esse processo, o software também disponibiliza as ferramentas para balanceamento dos pesos de cada junta do esqueleto. (Maya, 2015)

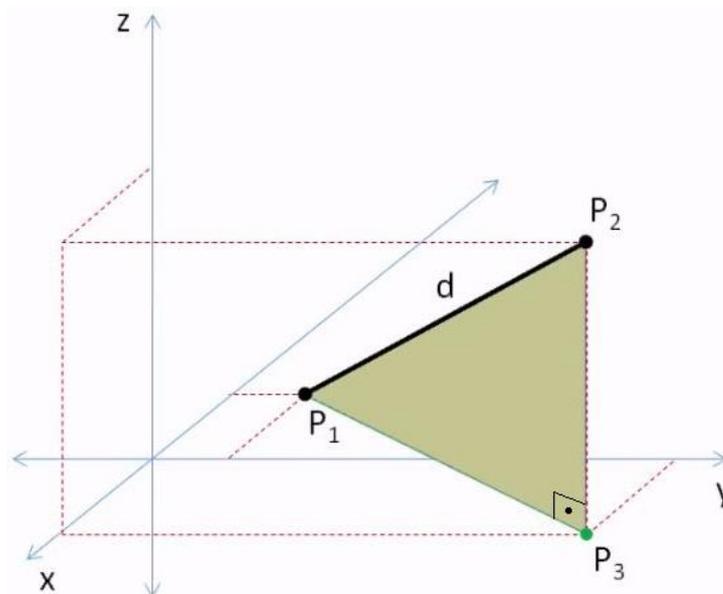
### **3.3 – Fluxo do programa**

Ao inicializar o programa, o Kinect ficará aguardando a detecção de um usuário em frente ao sensor. Apenas um usuário é rastreado por vez, nesse caso, o mais próximo do aparelho. Após reconhecimento de um usuário, o programa irá mapear os pontos das juntas do esqueleto. Esses pontos serão esferas invisíveis no ambiente 3D. Sua visibilidade só é necessária no caso de tratamento de erros. Os modelos de roupas já estarão construídos nessa etapa.

### 3.3.1 – Extração das medidas

Uma vez que diferentes usuários terão diferentes proporções de largura e altura, essas medidas são necessárias para a adaptação dos modelos virtuais das roupas. Elas são obtidas a partir do mapeamento dos pontos do esqueleto. Neles, utiliza-se a função de cálculo de distância  $D$  entre dois pontos  $P_1(x,y,z)$  e  $P_2(x,y,z)$  posicionados nos ombros direito e esquerdo, respectivamente. Como a distância entre esses dois pontos é uma reta, podemos construir um triângulo retângulo usando os segmentos de reta  $P_1P_3$  e  $P_2P_3$  conforme a Figura 3-2. Calcula-se então a hipotenusa de acordo com o teorema de Pitágoras.

Figura 3-2 Distância entre os pontos através do teorema de Pitágoras

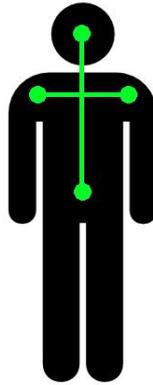


O segmento  $d$  é a hipotenusa do triângulo retângulo formado pelo terceiro ponto.  
Fonte: (G.R.1.618..., 2015)

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Essa distância é utilizada para ajuste da largura da roupa virtual. No caso da altura, o mesmo cálculo é realizado trocando-se os pontos para os da cabeça e cintura respectivamente. A Figura 3-3 exibe o posicionamento dos pontos usados para extração da largura e altura dos usuários reconhecidos.

Figura 3-3 Pontos escolhidos para medição de largura e altura



Fonte: (Próprio autor)

### 3.3.2 – Interface do programa

Após o reconhecimento do usuário também é exibida a interface de interação com o programa. Ela consiste em dois botões que trocam o modelo de roupa atualmente exibido para os modelos anteriores e seguintes respectivamente. Esses botões são posicionados do lado esquerdo e direito da cabeça do usuário e se movimentam de acordo com ela, usando como informação as coordenadas (x, y, z) do ponto do esqueleto mapeado na cabeça do usuário durante a calibração dele. O capítulo 4 contém imagens dos botões e da interface.

### 3.3.3 – Alinhamento das roupas virtuais

Um dos maiores desafios de provedores virtuais é o alinhamento correto da sobreposição dos modelos 3D no vídeo espelhado do usuário, uma vez que, qualquer mínimo desvio é visível e atrapalha o realismo da simulação.

Após a calibração do usuário, o ponto da cintura do esqueleto ligado à roupa é alinhado à frente do ponto do esqueleto do usuário. As distâncias da altura e largura obtidas anteriormente garantem um alinhamento aproximado nas bordas da roupa. A movimentação da roupa é feita de acordo com os ângulos formados pelas juntas do esqueleto do usuário, que são então copiadas no esqueleto da roupa, fazendo com que o esqueleto da roupa se movimente igual ao esqueleto do usuário.

## CAPÍTULO 4

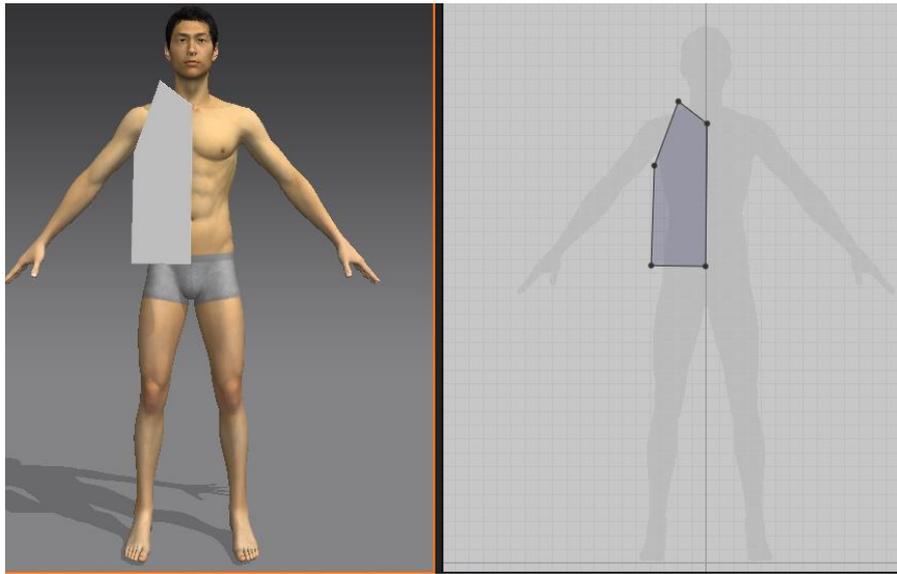
### DESENVOLVIMENTO

Esse capítulo descreve em detalhes o processo de implementação de cada etapa do sistema, começando pela criação dos modelos virtuais das roupas, seguindo para a preparação delas para que possam se deformar e ajustar aos usuários.

#### 4.1 – Criação das vestimentas 3D

Os modelos das roupas precisam estar finalizados antes de serem utilizados pelo Unity. Elas são criadas usando a ferramenta Marvelous Designer, como descrito na sessão 3.2.4. Para o desenvolvimento é necessário que as roupas sejam criadas usando como base um modelo de manequim. Pode-se escolher um dos cinco modelos que já acompanha o *software* ou efetuar uma importação de algum modelo desenvolvido em outros programas específicos para modelagem 3D.

Escolhido o manequim, desenha-se com retas e pontos os polígonos em duas dimensões que servirão de base para a frente e costas da roupa. Uma vez que o corpo humano apresenta simetria vertical, desenha-se um lado da roupa de acordo com o eixo escolhido e aplica-se um espelhamento. Isso evita possíveis diferenças entre os dois lados do modelo da roupa. A Figura 4-1 demonstra o desenho de uma camiseta desenhada usando como base a metade direita do corpo do manequim.

**Figura 4-1 Metade da parte da frente do modelo de uma camisa**

Fonte: (Próprio autor)

A Figura 4-2 mostra o resultado do espelhamento vertical. O lado esquerdo da figura mostra a roupa renderizada em sua forma final e o lado direito, o polígono responsável por ela.

**Figura 4-2 Aplicação do espelhamento para simetria vertical**

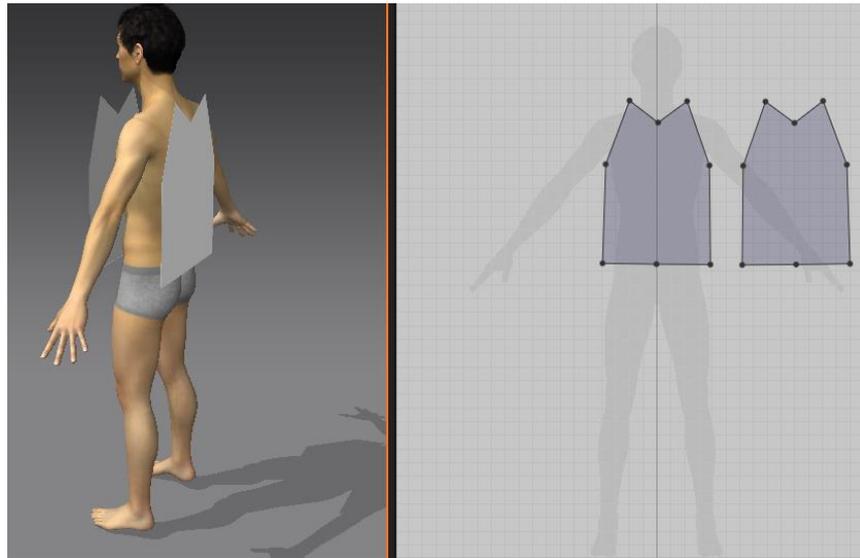
Fonte: (Próprio autor)

Esse polígono só é transformado em um modelo de roupa utilizável após ser costurado a outro polígono que representa a parte das costas da vestimenta. Nessa parte, é possível copiar a parte da frente, criar a parte das costas do início ou replicar e modificar a parte frontal. Ao copiar a parte frontal, é importante observar e acertar a orientação da parte do interior e exterior da roupa, pois o programa aplica uma iluminação diferenciada nas texturas internas e externas das roupas.

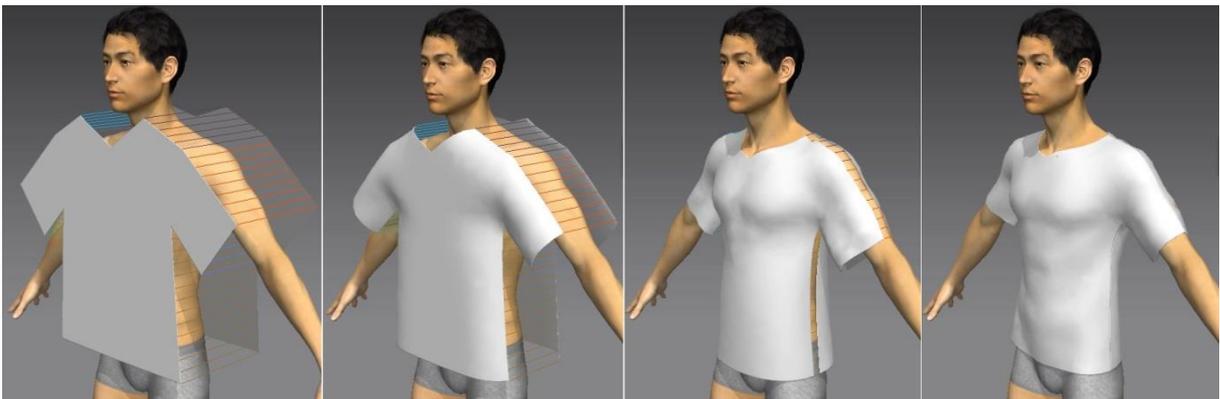
Na Figura 4-3 é possível observar a localização da parte das costas que, nesse caso, foi replicada e espelhada horizontalmente da parte frontal.

Nessa etapa, é realizada a costura entre os dois ou mais polígonos, respeitando-se os espaços livres de, por exemplo, pescoço, braços e cintura, no caso de um camisa. A ferramenta usa as linhas de costura como guia para adaptação dos polígonos ao redor do manequim, detectando e tratando colisões entre a roupa e o manequim.

Deixar os polígonos, dos desenhos das roupas, mais próximos do manequim, durante a etapa de ajuste da roupa no manequim, garante resultados melhores na renderização do modelo final da vestimenta. Se a etapa falhar em encaixar algumas partes da roupa no corpo do manequim, é possível reajustar e corrigir manualmente as partes específicas após a adaptação automática. A Figura 4-4 ilustra a costura e etapa de adaptação.

**Figura 4-3 Replicação e reorientação da parte das costas**

Fonte: (Próprio autor)

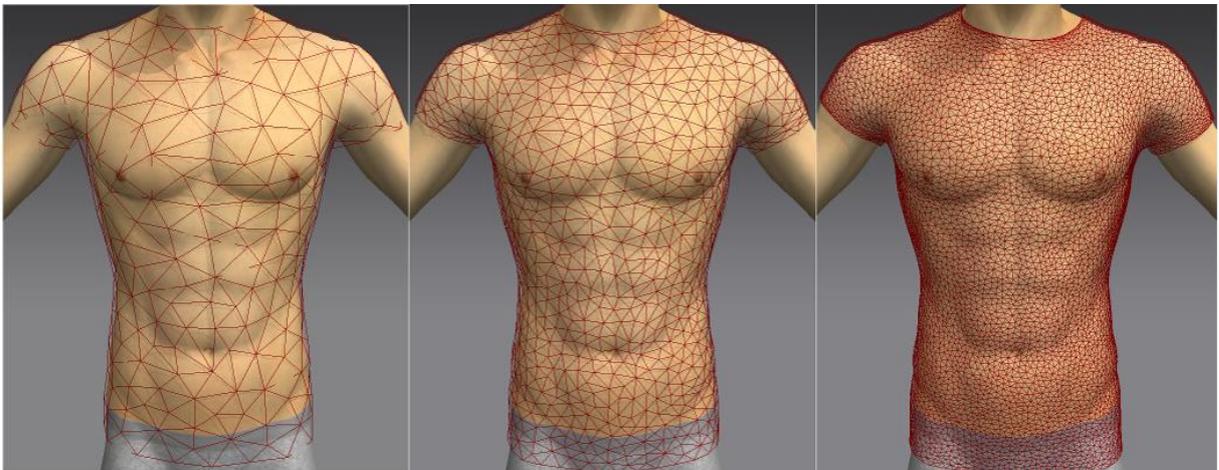
**Figura 4-4 Costura e adaptação dos polígonos ao manequim**

Fonte: (Próprio autor)

A contagem de triângulos do modelo 3D também precisa ser avaliada durante sua criação. Modelos com poucos triângulos garantem o melhor desempenho, mas as vestimentas ficam menos “reais”, o que influencia diretamente a imersão do provador. Por outro lado, o excesso de triângulos faz com que menos quadros sejam renderizados por segundos, o que faz com que a movimentação no provador fique “arrastada” devido ao atraso no processamento. Naturalmente que, com o avanço do poder de processamento das placas de vídeo, modelos com cada vez mais triângulos poderão ser utilizados, gerando roupas cada vez mais realistas e com

aparência mais próxima daquelas dos seus modelos reais. Os modelos virtuais das vestimentas criadas nesse projeto foram desenvolvidos dentro de uma contagem de triângulo na faixa de 2000 a 3000. Esses valores foram suficientes para garantir um bom custo benefício entre qualidade e desempenho no hardware escolhido para a máquina do projeto, que consiste de um processador Intel Core i7 2.40 Ghz com 8 GB de memória RAM e placa de vídeo GeForce GT 640M. A Figura 4-5 exibe um mesmo modelo de camisa com poucos triângulos na figura da esquerda, com a quantidade utilizada nesse projeto no meio e muitos triângulos na imagem da direita.

**Figura 4-5 Variação na contagem de triângulos de modelos 3D**



Fonte: (Próprio autor)

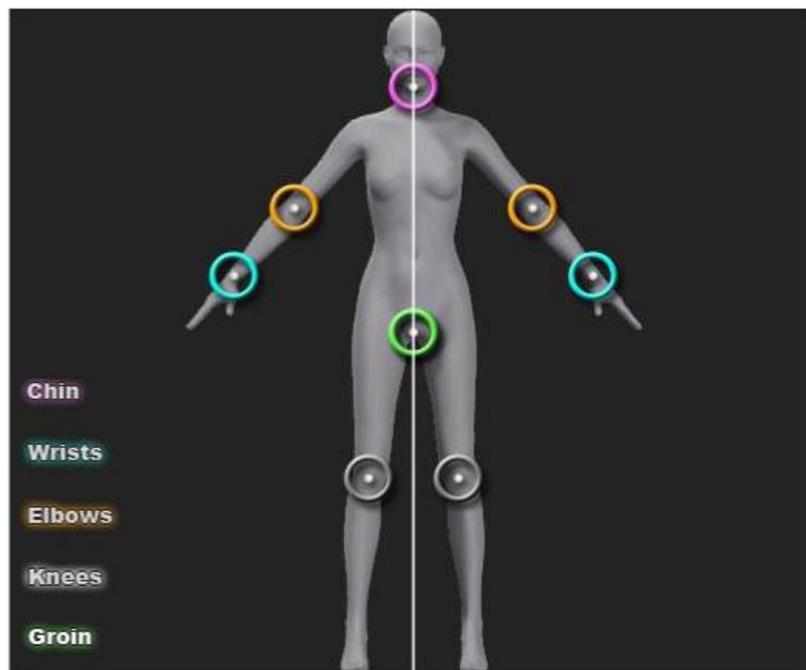
#### **4.2 – Deformação das roupas de acordo com movimentos do usuário**

Os modelos construídos pelo Marvelous Designer são estáticos e não podem ser animados ou movimentados. Uma vez que é preciso que a roupa se movimente em sincronia com o usuário no provador virtual, ela precisa ser capaz de se deformar adequadamente, de acordo com os movimentos dele.

Para que isso seja possível, é necessário atribuir um RIG ao modelo 3D da roupa. O RIG, também chamado de esqueleto, é composto de ossos e articulações que armazenam informações de como cada parte da malha 3D deve se comportar durante as deformações causadas pela movimentação dela. Por exemplo, toda a parte da malha referente ao braço direito da roupa deve se deformar quando o usuário movimentar ou rotacionar seu braço direito.

O software Maya possui funcionalidades para ligação e ajuste dos modelos de roupa aos seus esqueletos e foi a ferramenta escolhida junto com (Mixamo, 2015) para esse processo. Mixamo facilita o processo de atribuição de esqueletos a malhas com formato humano e foi utilizada para atribuição do esqueleto no manequim usado durante a construção das vestimentas. Na Figura 4-6 observa-se o processo de escolha dos pontos de referência e de calibração para criação do esqueleto.

Figura 4-6 Escolha dos pontos para criação do esqueleto



Fonte: (Próprio autor)

Com o esqueleto calibrado, é possível adaptar a malha, também chamada de *skin* ou pele, a diferentes deformações das animações. Os ossos desse esqueleto também podem guardar informações de restrições como, por exemplo, o pescoço que não pode girar 360 graus em nenhum eixo. A Figura 4-7 ilustra um esqueleto executando a animação de correr e a malha deformada de acordo.

Figura 4-7 Visualização do esqueleto finalizado



Fonte: (Próprio autor)

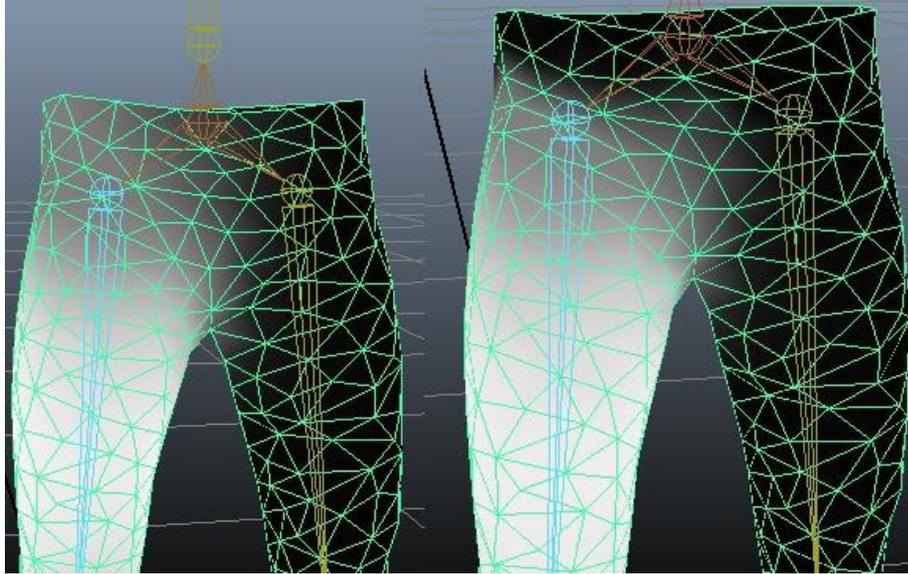
Não é possível usar essa funcionalidade diretamente nos modelos de roupa, pois não possuem todos os pontos de calibração. Uma camisa, por exemplo, não possuirá, na maioria das vezes, pontos de referência para a cabeça, cintura e pernas. O esqueleto é, portanto, atribuído ao manequim para que possa servir de estrutura inicial para todos os modelos de roupas que foram construídos para ele no Marvelous Designer.

O software Maya efetua a ligação desse esqueleto a cada peça de roupa e possui ferramentas que possibilitam ajustar os pesos de cada osso em relação à malha, facilitando ajustes em deformações incorretas.

O balanceamento de pesos é feito através de um mapa na malha que exhibe tonalidades de branco variando do totalmente preto (não causa deformação alguma) ao totalmente branco (causa muita deformação). As cores são apresentadas separadamente para cada junta escolhida. Portanto, ao selecionar o osso da perna direita, são exibidas as tonalidades apenas dos locais onde esse osso deforma a malha. Na Figura 4-8 é possível consultar o mapa da perna direita antes e depois de

um ajuste na parte da cintura, que antes era deformada incorretamente quando o usuário levantava a perna, por exemplo.

Figura 4-8 Mapa de peso da perna direita



Fonte: (Próprio autor)

Pode ser necessário ajustar os pesos nas partes internas das malhas em alguns casos.

Ajustado os modelos de roupas aos seus respectivos esqueletos, eles estão prontos para serem importados e utilizados pelo Unity.

### 4.3 – Desenvolvimento da cena de realidade aumentada

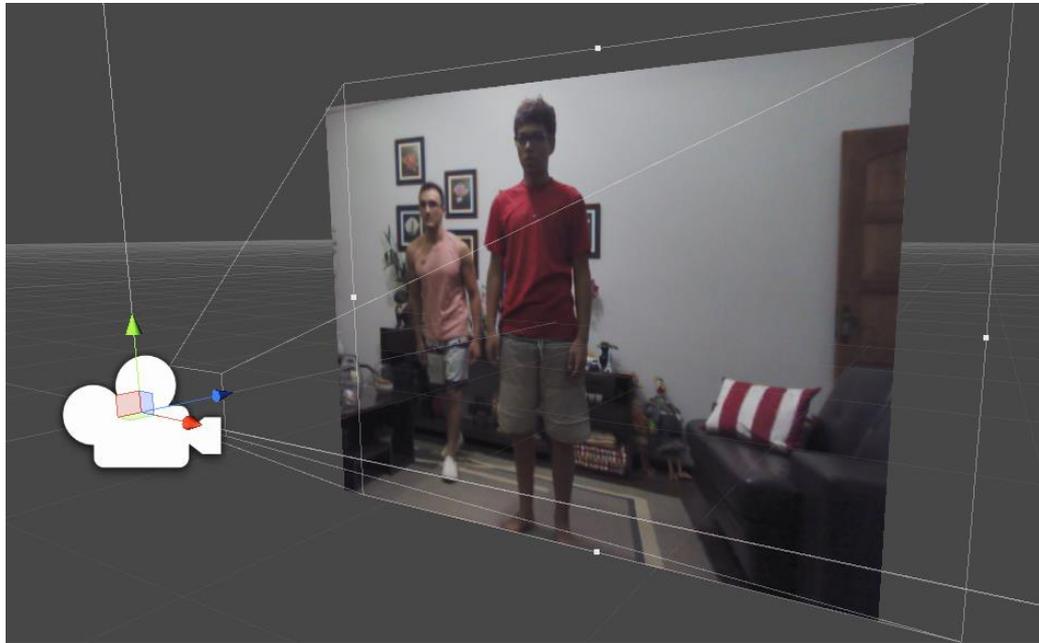
O Unity3D é o integrador de todas as partes. É responsável por juntar os modelos 3D, processar a lógica de programação, receber informações do Kinect e calcular qualquer física na cena. Cada ambiente desenvolvido no Unity é chamado de cena. Um programa, simulação ou jogo pode conter várias cenas. Um exemplo prático disso é uma cena que exibe um prédio por fora que não deve conter nenhuma informação do que há dentro do prédio. Por motivos de desempenho, uma outra cena do interior dele é criada e carregada de acordo com a necessidade ou não de se “entrar” nesse prédio.

A cena principal do provador virtual é formada por um plano, uma câmera, uma fonte de luz direcionada, os modelos de roupa importados do Maya e a interface de interação do usuário com o provador (GUI - *Graphical User Interface*).

O plano funciona como um espelho exibindo o fluxo de vídeo fornecido pela câmera RGB do Kinect. Os *frames* precisam estar invertidos horizontalmente.

A Figura 4-9 mostra a relação entre a câmera da cena e o espelho. Nessa parte é necessário que a área de visualização da câmera cubra todas as margens do plano, caso contrário a imagem ficará cortada ou muito distante.

**Figura 4-9** Posicionamento da câmera e espelho na cena



**Fonte:** (Próprio autor)

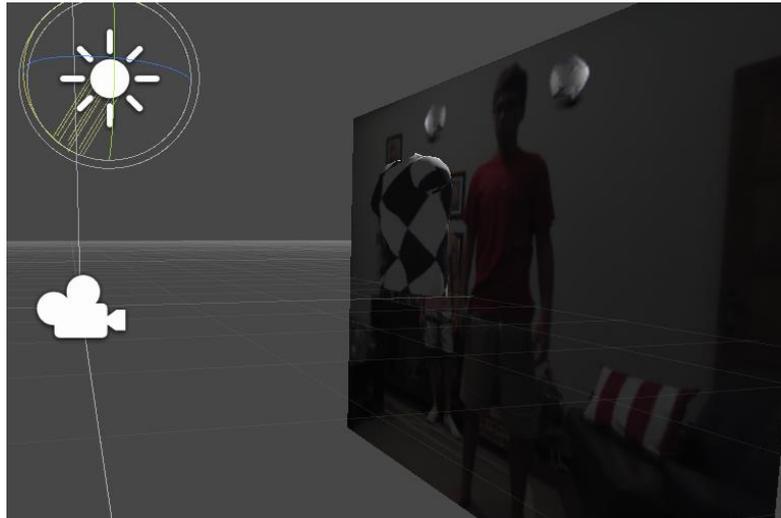
Sem a correta iluminação na cena, não será possível visualizar o vídeo no espelho, ou ele ficará muito escuro, como na Figura 4-10. Se a luz ficar diretamente sobre a cena, os modelos de roupa ficarão muito brancos e a iluminação ficará com uma aparência artificial.

Pode-se controlar a iluminação do ambiente virtual de acordo com a quantidade de luz na cena real, de forma que as roupas sempre tenham uma iluminação de acordo com o ambiente da captura das imagens. Isso não foi implementado nesse projeto e ficará como sugestão de melhoria.

Unity3D também foi responsável pela aplicação das texturas nos modelos de roupa, embora isso pudesse ter sido realizado pelo Maya ou Marvelous Designer.

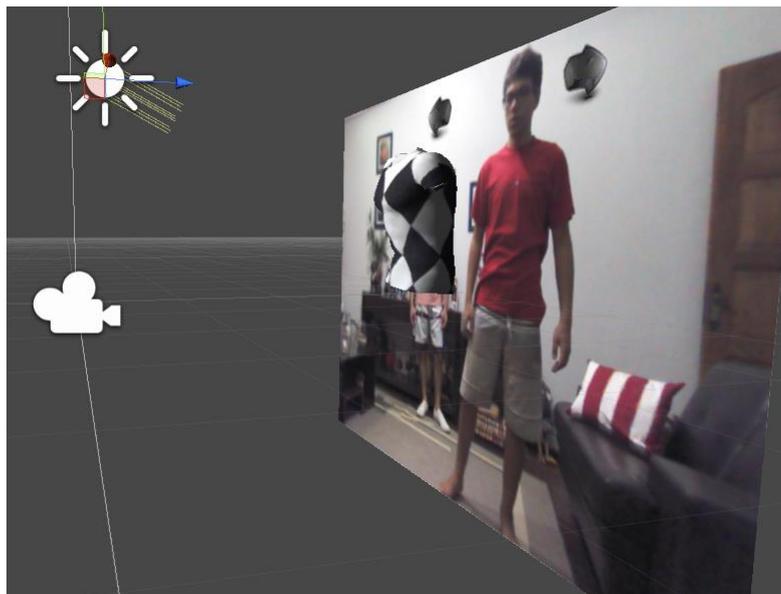
As texturas são imagens planas, com ou sem efeitos e filtros, aplicadas às malhas. Na Figura 4-12 é possível conferir três exemplos de imagens escolhidas como texturas para roupas no provador virtual desse projeto.

**Figura 4-10 Orientação incorreta da luz direcionada**



Fonte: (Próprio autor)

**Figura 4-11 Direcionamento da luz escolhida**



Fonte: (Próprio autor)

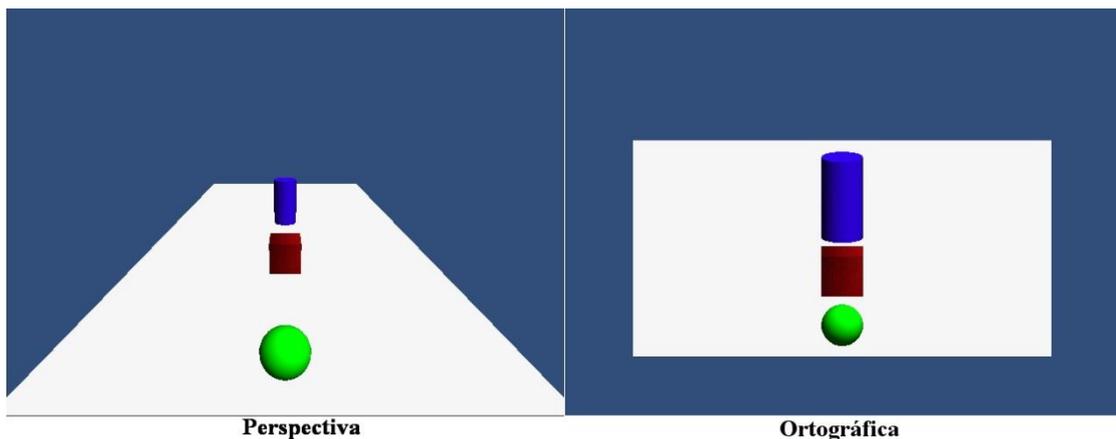
Figura 4-12 Algumas imagens utilizadas como texturas das vestimentas



Fonte: (Próprio autor)

A câmera foi configurada para projetar a cena ortograficamente. Dessa forma, não haverá necessidade de se ajustar a distância dos objetos em relação à câmera. No modo original, a projeção é em perspectiva. Por isso objetos mais distantes parecem estar menores, enquanto os mais próximos parecem maiores. Ao invés de alinhar as roupas através dessa aproximação e afastamento, elas ficarão em uma posição de profundidade fixa em frente ao espelho e terão sua largura e altura ajustadas conforme descrito em detalhes a seguir. A Figura 4-13 exibe uma mesma cena sendo projetada com as vistas perspectiva e ortográfica, respectivamente.

Figura 4-13 Visão Perspectiva e Ortográfica



Cena construída no Unity3D com o objetivo de exibir as diferenças entre os sistemas de projeção.

Fonte: (Próprio autor)

O provedor desse projeto apresenta seis roupas conforme vistas na Figura 4-15. Dessas, duas foram desenvolvidas para ficarem mais folgadas que o

manequim e configuradas para usar o componente de cálculo de física de tecidos disponível do Unity.

**Figura 4-14** Comparação entre o modelo de bermuda mais folgado e justo



**Fonte: (Próprio autor)** Figura 4-15 Roupas disponibilizadas no provador



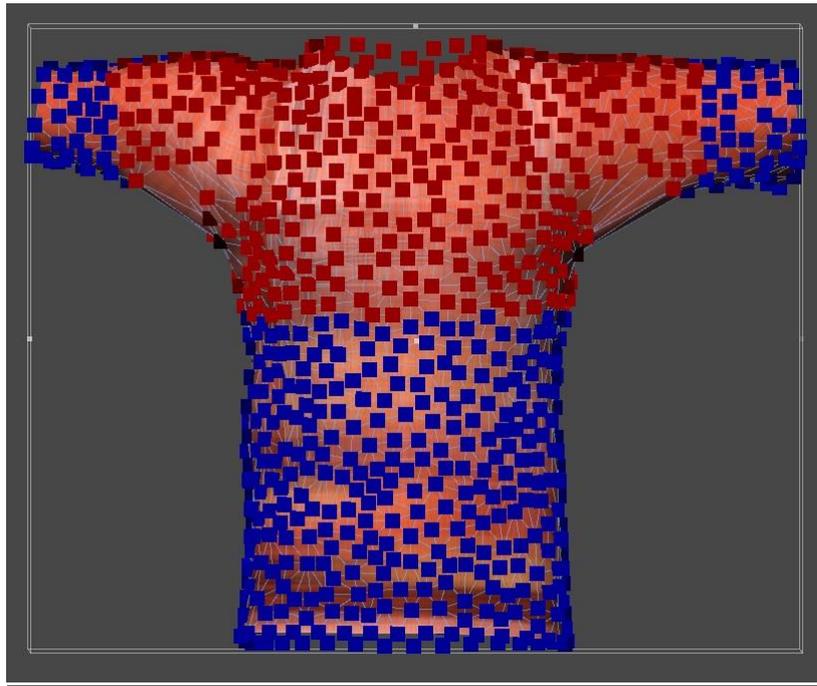
**Fonte: (Próprio autor)**

Os modelos de roupa, os quais serão configurados para terem cálculos de física de tecidos, precisam ter um componente a mais no Unity. Existem duas opções, o *Interactive Cloth* e o *Skinned Cloth*.

O *Interactive Cloth* funciona calculando, em cada frame, as colisões de cada triângulo da malha com qualquer outro objeto da cena. Pode ser utilizado em qualquer malha, mas é computacionalmente complexo e requer muito poder de processamento conforme aumenta-se o número de triângulo e, conseqüentemente, pontos de colisão. Apesar de ter sido utilizado no trabalho de Presle (Presle, 2012) e ser uma opção muito interessante para esse projeto, não é recomendada a sua utilização em roupas, pela documentação online do Unity (ManualUnity, 2015), onde é sugerido que seja usado apenas em cortinas e outros tecidos que não sejam vestuário.

O componente *Skinned Cloth*, adiciona simulação de roupas a uma malha de pele. Malha de pele é usada nos personagens, conforme visto anteriormente no manequim e requer uma estrutura de ossos (esqueleto). Esse componente é responsável pelos cálculos de física na região onde a malha é deformada, quando um osso responsável por essa região é movimentado. O resultado depende da velocidade do movimento realizado. É possível configurar quão livremente cada vértice pode se afastar em relação ao osso, uso de gravidade, se haverá colisão com vértices do próprio tecido, a espessura, capacidade de dobrar e esticar, etc. O fato desse componente usar um esqueleto, faz com que seja mais facilmente utilizado pelo Kinect. Um ponto negativo seria a ausência de interatividade com outros objetos em cena, mas como nesse provador virtual não haverá outros objetos 3D interagindo com os modelos de roupa, isso não será um problema, tornando essa a escolha ideal para esse trabalho. A Figura 4-16 exibe o processo de configuração desse componente. Vértices com diferentes cores possuem diferentes configurações, possibilitando que, por exemplo, partes mais justas como ombros e peito, possam estar configuradas diferentes de partes mais folgadas, como a parte do abdômen e cintura.

Figura 4-16 Configuração do componente Skinned Cloth



Fonte: (Próprio autor)

#### 4.4 – Lógica de programação

Finalizada a construção da cena, é necessário escrever os códigos que controlarão o fluxo do programa.

Ao iniciar o provador, todos os modelos de roupa são armazenados em um vetor e uma função desliga a renderização de todas as vestimentas, exceto a primeira roupa, que fica no índice 0 do *vetor* e é a primeira a ser exibida. Botões na interface ficam responsáveis por alterar o valor desse índice, tomando cuidado para não ultrapassar o limite do vetor, de forma que, ao chegar no último índice, retorne-se para o começo do *vetor* e conseqüentemente à primeira roupa. A cada troca, desliga-se a renderização da roupa anterior e ativa-se a da próxima ou vice-versa, caso o botão pressionado seja o de retornar.

Outra função garante que as roupas e a interface só serão renderizadas no espelho, se um usuário for reconhecido em frente ao Kinect. Na situação de duas ou mais pessoas serem detectadas, aquela que estiver mais próxima do sensor será o usuário do qual os comandos gestuais serão interpretados. O SDK do Kinect permite

que mais pessoas interajam ao mesmo tempo, mas não é de interesse desse projeto que duas pessoas usem o mesmo provedor simultaneamente. Também há outras configurações de escolha de qual usuário será o principal, no caso de múltiplas pessoas reconhecidas. A configuração de usuário mais próximo atendeu satisfatoriamente. Na Figura 4-17 confere-se o resultado desse tipo de cenário.

Reconhecido um usuário principal, diversas esferas não renderizadas são mapeadas nas juntas do esqueleto reconhecido pelo Kinect. Todas com tamanhos iguais, com exceção das mãos, que possuem esferas maiores que serão utilizadas na interação com a interface, explicado em detalhes na sessão 4.5 – Interface de interação.

As esferas dos ombros, cabeça e cintura são usadas para extração da altura e largura usadas na configuração dessas escalas na roupa, conforme descrito em 3.3.1 – Extração das medidas. Os pontos dos pés poderiam ser utilizados, mas esses não são imediatamente visualizados assim que o usuário é reconhecido em frente ao sensor e muitas vezes não ficam dentro da área de visualização da câmera.

Nessa etapa, é importante verificar diferenças que possam existir entre as medidas do esqueleto original das roupas (o do manequim) e o esqueleto construído pelo Kinect. Essas diferenças devem ser ajustadas ou ignoradas, caso contrário, a escala ficará incorreta e a roupa não vestirá o usuário adequadamente.

Figura 4-17 Situação de múltiplos usuário em frente ao sensor



Fonte: (Próprio autor)

#### 4.5 – Interface de interação

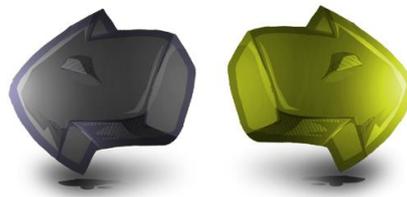
A Figura 4-18 demonstra as imagens exibidas nos botões de seleção de roupa, o botão da esquerda é o utilizado para retornar a roupa anterior e sua cor é a de desativado, enquanto o botão da direita seleciona a próxima roupa e sua cor é de quando está sendo pressionado. Ambos são posicionados e movimentam-se de acordo com a esfera localizada na junta da cabeça do esqueleto reconhecido pelo Kinect.

A detecção de que o usuário pressionou um botão é efetuada a partir de uma função que verifica a cada frame se as esferas invisíveis mapeadas nas mãos do usuário colidem com as margens dos quadrados em torno dos botões. Ao colidir, a função de troca de roupa é executada e a textura do botão muda de cor com a finalidade de dar feedback ao usuário quanto ao pressionamento bem sucedido. Um novo pressionamento só é registrado se a esfera sair da área de colisão e entrar

novamente, caso contrário o programa interpretaria um pressionamento para cada frame que a mão estivesse dentro da área de colisão do botão. Observando a Figura 4-19, é possível enxergar essas linhas de colisão. Na mão direita é visível a interseção com o botão direito e conseqüente pressionamento, enquanto isso não ocorre na mão esquerda.

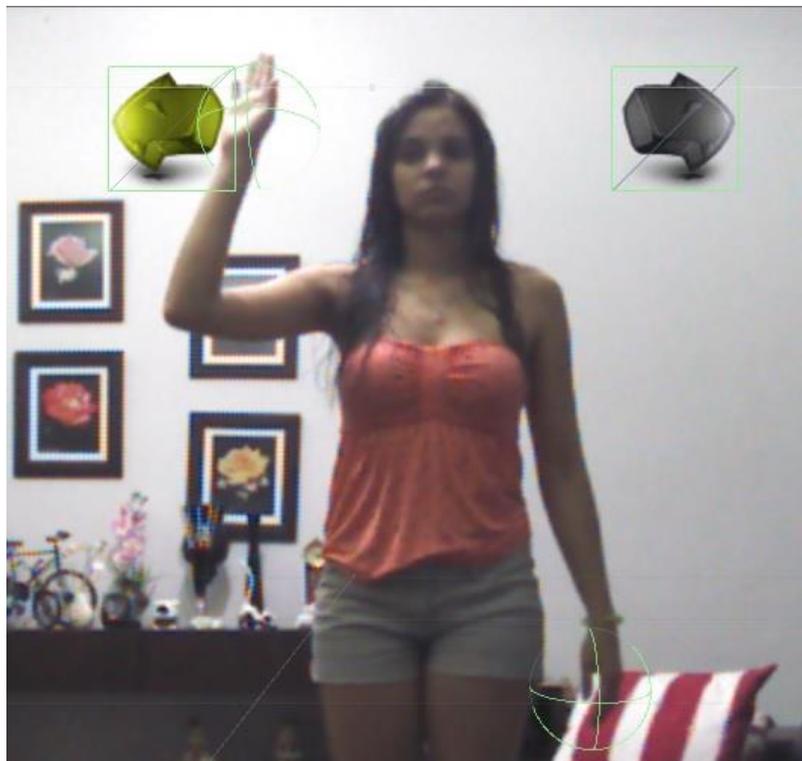
A interface também informa o nome da vestimenta atual no canto direito superior da tela. Outras informações, como preço ou disponibilidade, também poderiam aparecer nessa área.

**Figura 4-18 Botões seletores de roupa usados na interface**



Fonte: (Próprio autor)

**Figura 4-19 Linhas de colisão entre as mãos do usuário e os botões**



Fonte: (Próprio autor)

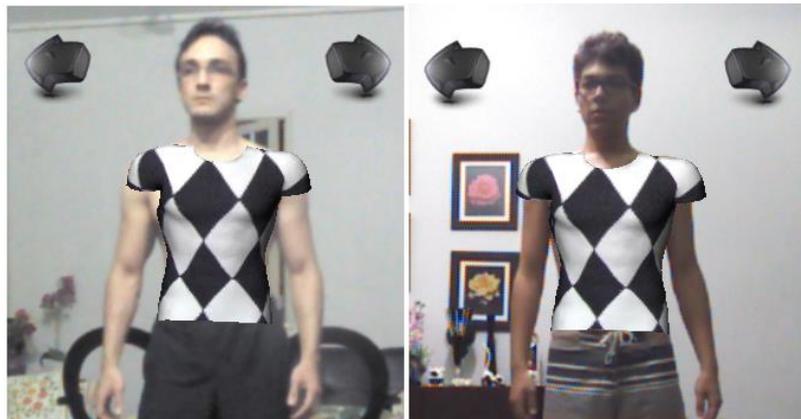
## CAPÍTULO 5

### RESULTADOS

As figuras abaixo comparam os resultados das adaptações dos diferentes modelos de roupas aos usuários com proporções diferenciadas em diversas poses.

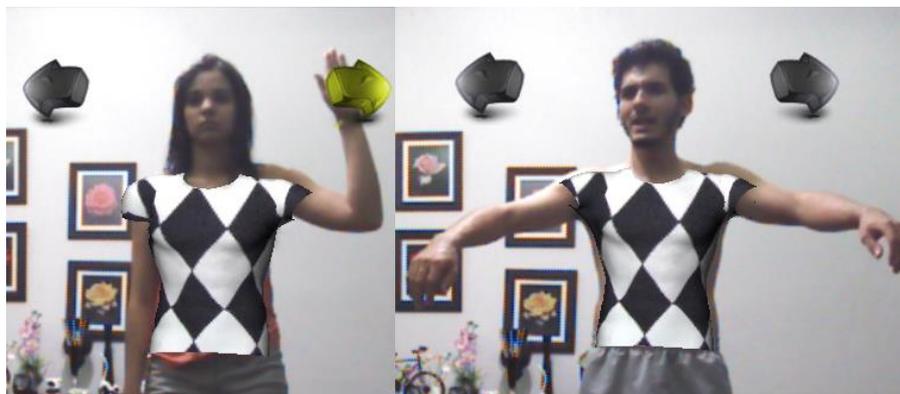
Enquanto se movem, é possível conferir a deformação da malha da roupa, principalmente nos braços e pernas, ficando quase sempre na mesma posição do usuário. Foi realizado o *upload*, para o Youtube, de um vídeo com algumas cenas de usuários testando o provador. O link pode ser encontrado na referência (ProvadorVirtual, 2015).

Figura 5-1 1º Modelo de Camisa – Testando ajuste de tamanho



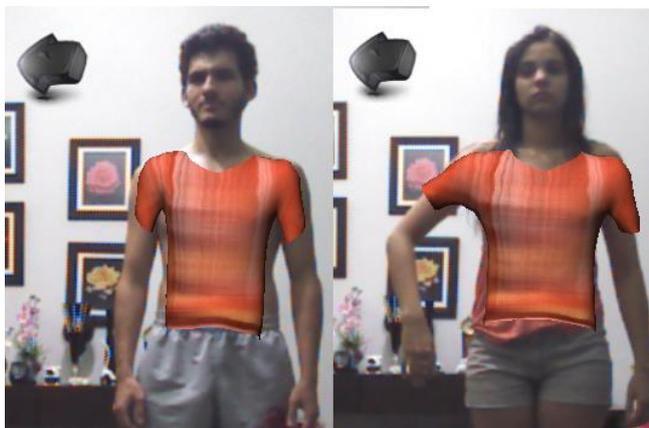
Fonte: (Próprio autor)

Figura 5-2 1º Modelo de Camisa - Testando movimentação



Fonte: (Próprio autor)

Figura 5-3 Modelo de camisa com física de tecidos configurada



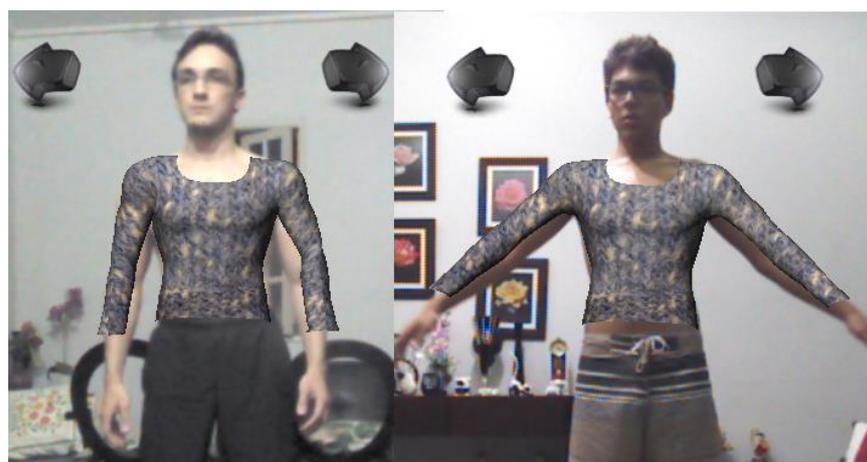
Fonte: (Próprio autor)

Figura 5-4 Testando a física de tecidos



Fonte: (Próprio autor)

**Figura 5-5 Testando modelo com mangas longas**



**Fonte: (Próprio autor)**

**Figura 5-6 Testando rotação e movimentação do modelo**



**Fonte: (Próprio autor)**

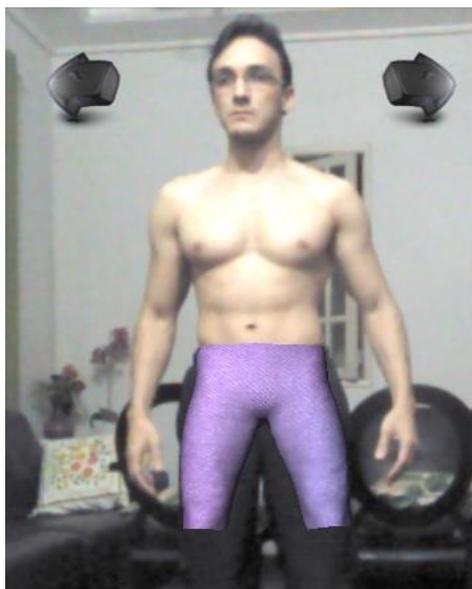
Figura 5-7 Testando o modelo de bermuda com física



Fonte: (Próprio autor)

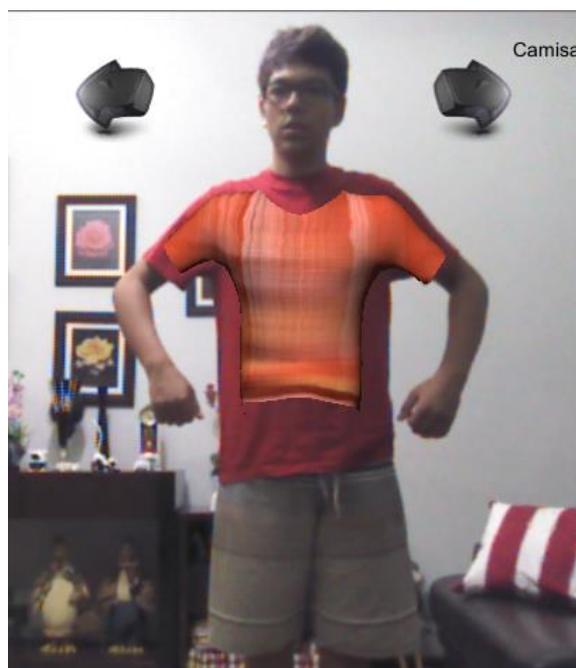
Os modelos mais folgados são mais facilmente adaptados do que os justos. Uma vez que não são extraídas as medidas da largura dos braços e pernas, a circunferência desses membros dificilmente seria a mesma do manequim. Uma roupa mais folgada garante que um número maior de circunferências possa ser vestida do que uma mais justa. A Figura 5-8 mostra esse problema e torna clara a importância do usuário estar vestindo roupas reais mais coladas ao corpo, enquanto usa o provador virtual. Caso contrário, ficaria visível a sobra da roupa real atrás da sobreposição virtual como na Figura 5-9.

Figura 5-8 Dificuldade de alinhamento em roupas mais justas



Fonte: (Próprio autor)

Figura 5-9 Usuário vestindo roupas reais muito folgadas



Fonte: (Próprio autor)

## CAPÍTULO 6

### CONCLUSÃO

O provador virtual apresentado nesse projeto aparenta ser uma boa solução para uma maneira rápida e fácil de se experimentar roupas sem vesti-las. O sensor Kinect forneceu a tecnologia ideal para que essa implementação fosse bem sucedida. Comparado a outras tecnologias que usam marcadores ou conjuntos de hardware específicos, essa tecnologia é mais barata e permite que soluções de mercado possam ser criadas rapidamente.

Além disso, é um sistema simples que pode ser utilizado em casa, pois só é necessário um computador equipado com o sensor. Isso possibilitaria a utilização desse sistema para lojas que vendem suas roupas pela internet. O que por si só seria uma grande vantagem sobre a atual experiência de compras online.

A tecnologia também abre portas para um série de novas funcionalidades em outros projetos como: sistemas de reconhecimentos de gestos em diversas áreas, realidade aumentada sem marcadores, modelagem de objetos 3D facilitada, mapeamento de cenários para navegação em robótica e auxílio de medição através de processamento de imagens.

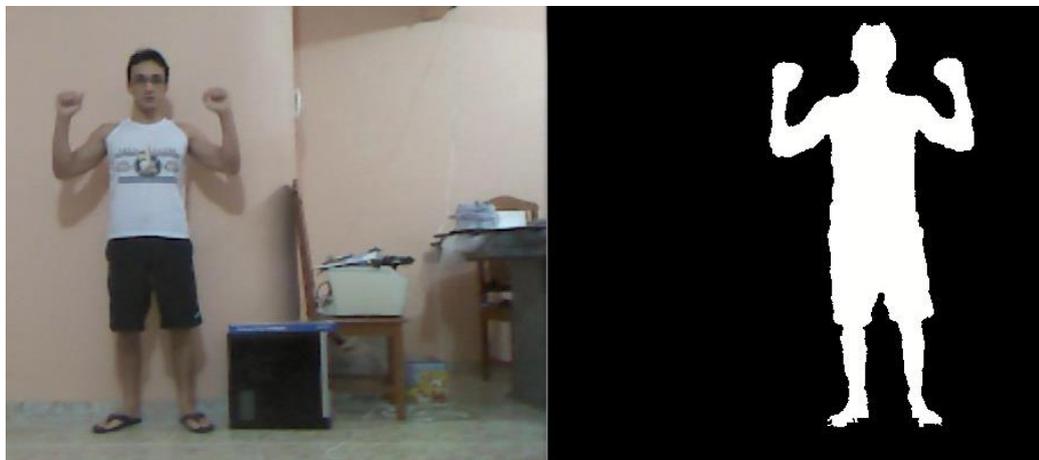
#### **1 – Trabalho Futuro**

Uma sugestão de melhoria imediata para o provador virtual é extrair mais medidas úteis do usuário com a finalidade de melhor ajustar os modelos de roupas virtuais. Inclusive as medidas de locais onde não há pontos de juntas do esqueleto construído pelo Kinect, como a circunferência da cintura, braços e coxas.

Essas medidas extras podem ser obtidas através da utilização de uma imagem de contorno do usuário que está interagindo com o sistema. Um exemplo desse mapa pode ser visto na Figura 6-1. Ao contabilizar a distância em pixels brancos de uma medida já conhecida do usuário como, por exemplo, a distância entre os ombros direito e esquerdo, pode-se aplicar regra de três para saber a

distância do meio da cintura (ponto obtido no esqueleto) até a margem esquerda ou direita do contorno.

Figura 6-1 Imagem de contorno do usuário



Fonte: (Próprio autor)

Outra oportunidade de melhoria seria verificar a quantidade de brilho na cena para poder adequar a iluminação do ambiente virtual. Isso faria com que as roupas ficassem mais claras ou mais escuras de acordo com a iluminação da cena real.

Como o objetivo desse trabalho era a extração e utilização das medidas do usuário, o problema da sobreposição, comumente encontrado em provadores virtuais, não foi resolvido. Isso significa que como as roupas estão sendo sobrepostas em frente ao plano de vídeo, não é possível tapar nenhuma parte delas como aconteceria no mundo real se posicionássemos os braços à frente das vestimentas.

É importante observar que a resolução da câmera do kinect não é de alta definição (HD ou Full HD) e isso faz com que a imagem tenha uma resolução baixa na maioria dos monitores e televisores atuais. O novo kinect resolve esse problema e isso causará uma melhoria significativa na qualidade das imagens reais exibidas no espelho do provador virtual. Com resolução maior, aumenta-se a precisão das medidas, outro ponto positivo na hora da substituição do sensor em projetos futuros.

**REFERÊNCIAS BIBLIOGRÁFICAS**

- Alhwarin, F., Ferrein, A., & Scholl, I. (2014). IR Stereo Kinect: Improving Depth Images by Combining Structured Light with IR Stereo. In: *PRICAI 2014: Trends in Artificial Intelligence* (pp. 409-421). Switzerland: Springer International Publishing .
- Azuma, R., Baillot, Y., Feiner, S., Julier, S., & MacIntyre, B. (Novembro/Dezembro de 2001). Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, pp. 34-47.
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV*. Sebastopol, CA: O REILLY.
- Cruz, L., Lucio, D., & Velho, L. (2012). Kinect and RGBD Images: Challenges and Applications. *XXV SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, 36-49.
- Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., et al. (2013). *Real-Time Human Pose Recognition in Parts from Single Depth Images*. Microsoft Research Cambridge & Xbox Incubation.
- Giovanni, S., Choi, Y. C., Huang, J., Khoo, E. T., & Yin, K. (2012). Virtual Try-on using Kinect and HD camera. In: *Motion in Games: Lecture Notes in Computer Science* (pp. 55-65). Rennes: Springer Berlin Heidelberg.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*. New Jersey: Prentice Hall.
- Isikdogan, F., & Kara, G. (2012). *A Real Time Virtual Dressing Room Application using Kinect*. CMPE537 COMPUTER VISION COURSE PROJECT.
- Loaiza, M. E., Raposo, A. B., & Gattass, M. (2010). Calibração de múltiplas câmeras baseado em um padrão invariante. *XII Symposium on Virtual and Augmented Reality*, 212-221.
- Mutto, C. D., Zanuttigh, P., & Cortelazzo, G. M. (2013). *Time-of-Flight Cameras and Microsoft Kinect: A user perspective on technology and applications*. Springer.

Pachoulakis, I., & Kapetanakis, K. (Agosto de 2012). Augmented Reality Platforms For Virtual Fitting Rooms. *The International Journal of Multimedia & Its Applications (IJMA) Vol.4, No.4*, 35-46.

Presle, P. (2012). *A Virtual Dressing Room based on Depth Data*. Vienna University of Technology: Faculty of Informatics.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Weiss, A., Hirshberg, D., & Black, M. J. (2011). *Home 3D Body Scans from Noisy Image and Range Data*. Tubingen, Germany: Perceiving Systems Dept., Max Planck Institute for Intelligent Systems.

PS Vita – WAAR (*Wide-Area Augmented Reality*); <http://www.youtube.com/watch?v=wUxSwsy3PZo> em 16/11/2013 página mantida pelo YOUTUBE.

OpenNI – *Open Source SDK for 3D sensors*; <http://openni.org/> em 29/11/2013 página mantida pela OpenNI.

KinectSDK – *Kinect for Windows | Voice, Movement & Gesture Recognition Technology*; <http://www.microsoft.com/en-us/kinectforwindows/> em 29/11/2013 página mantida pela Microsoft.

Unity3D – *Unity - Game Engine*; <http://unity3d.com/> em 29/11/2013 página mantida pela Unity Technologies.

Blender – *Free and Open 3D Creation Software*; <http://www.blender.org/> em 29/11/2013 página mantida pela Blender Foundation.

Maya – *3D Animation Software*; <http://www.autodesk.com/products/autodesk-maya/> em 29/11/2013 página mantida pela AUTODESK.

Cinema4D – *3D for the Real World*; <http://www.maxon.net> em 29/11/2013 página mantida pela MAXON.

3Ds Max – *Software de Modelagem e Renderização 3D*; <http://www.autodesk.com.br/products/autodesk-3ds-max/> em 29/11/2013 página mantida pela AUTODESK.

- Makehuman – *Open Source tool for making 3D characters*; <http://www.makehuman.org/> em 29/11/2013 página mantida pela MakeHuman team.
- Kinect2 – *Kinect for Windows SDK 2.0*; <http://www.microsoft.com/en-us/download/details.aspx?id=44561> em 22/01/2015 página mantida pela Microsoft.
- MarvelousDesigner; <http://www.marvelousdesigner.com/> em 22/01/2015 página mantida pela Virtual Fashion Inc.
- Mixamo – *3D Animation Online Service, 3D Characters, and Character Rigging*; <http://www.mixamo.com/> em 04/02/2015 página mantida pela Mixamo Inc.
- Microsoft Store – Buy Kinect Sensor for Xbox360; [http://www.microsoftstore.com/store/msusa/en\\_US/pdp/Kinect-for-Xbox-360/productID.253169000](http://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-for-Xbox-360/productID.253169000) em 23/02/2015 página mantida pela Microsoft.
- Totallmersion – *Augmented Reality Software and Solutions*; <http://www.t-immersion.com/> em 25/03/2015 página mantida pela Total Immersion.
- Fitnect – *3D Virtual fitting dressing room*; <http://www.fitnect.hu> em 25/03/2015 página mantida pela Fitnect Interactive.
- Bodymetrics – *The Science of Fit*; <http://www.bodymetrics.com> em 25/03/2015 página mantida pela BodymetricsLtd.
- G.R.1.618... – *The Distance Formula in 3D*; <https://youtu.be/c5vodU05l8Y> em 24/03/2015 página mantida pelo Youtube.
- CuriousInventor – *How the Kinect Depth Sensor Works in 2 Minutes*; <https://youtu.be/uq9SEJxZiUg> em 25/03/2015 página mantida pelo Youtube.
- ManualUnity – Unity – Manual: Interactive Cloth; <http://docs.unity3d.com/es/current/Manual/class-InteractiveCloth.html> em 25/03/2015 página mantida por Unity Technologies.

ProvadorVirtual – [Kinect Unity3D] Virtual Fitting/Dressing Room | Provador Virtual de Roupas (Clothes); <https://youtu.be/yFGmWeiqSKI> em 25/03/2015 página mantida pelo Youtube.

Fiala, M. (2005). Artag, a fiducial marker system using digital techniques. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition.