



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Carlos Diego Carvalho Chaves

**AUTOMATIZAÇÃO DO IMAGE MARKING PROCEDURE PARA
ANÁLISE DO ESQUEMA CORPORAL UTILIZANDO O KINECT**

Palmas - TO

2014

Carlos Diego Carvalho Chaves

**AUTOMATIZAÇÃO DO IMAGE MARKING PROCEDURE PARA
ANÁLISE DO ESQUEMA CORPORAL UTILIZANDO O KINECT**

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.Sc. Fabiano Fagundes

Palmas - TO

2014

Carlos Diego Carvalho Chaves

**AUTOMATIZAÇÃO DO IMAGE MARKING PROCEDURE PARA
ANÁLISE DO ESQUEMA CORPORAL UTILIZANDO O KINECT**

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.Sc. Fabiano Fagundes

Aprovado em: ____/____/____

BANCA EXAMINADORA

Prof. M.Sc. Fabiano Fagundes

Centro Universitário Luterano de Palmas - CEULP

Prof. M.Sc. Fernando Luiz de Oliveira

Centro Universitário Luterano de Palmas - CEULP

Prof. M.Sc. Pierre Soares Brandão

Centro Universitário Luterano de Palmas - CEULP

Palmas - TO

2014

Dedico este trabalho primeiramente a Deus, o único que é digno de honra e glória, e a meus pais: Carlos Teixeira Chaves e Jucélia Carvalho Vieira Chaves, que sempre me ensinaram e apoiaram no caminho que devo andar, com amor, carinho, compreensão e motivação.

AGRADECIMENTOS

Primeiramente agradeço à Deus, que é a única razão de existirmos, por ter possibilitado que eu chegasse até aqui. Sou eternamente grato a Ele pela mão estendida a mim nos momentos difíceis ao longo dos anos que se passaram. Sempre serei mui grato a Ti, Senhor, pois sei que sem Ti não teria conseguido chegar até aqui. Muito obrigado.

Agradeço aos meus pais por tudo que fizeram por mim. Sempre me motivaram a seguir em frente, ajudando-me a tomar as melhores decisões possíveis diante das mais diversas situações. Suas contribuições foram imprescindíveis para que eu pudesse realizar este sonho. Obrigado por todo incentivo, amor e carinho. Amo vocês.

Agradeço a minha noiva Leidiane, pela paciência, compreensão, motivação e carinho. Suas orações sempre foram importantes, e seu otimismo sempre me encorajou a seguir cada vez mais em frente. Obrigado por estar presente todo este tempo, e por ter me ajudado a superar mais este objetivo de vida. Te Amo.

Não posso deixar de agradecer aos meus irmãos, Carla, Pablo e Fábio, a meus primos, Andrew e Giulliano, a todos meus familiares e irmãos em Cristo. Vocês foram compreensíveis e pacientes quando eu estava ausente e sem paciência. Muito obrigado.

Agradeço a meu “chegado” Gleyson Rocha Moura, amigo para todas as horas. Estávamos sempre juntos, tanto em horas difíceis quanto em momentos de alegria. Tu me apoiaste nas diversas situações durante todo este tempo. Obrigado.

Agradeço também a meu orientador, Fabiano Fagundes, que contribuiu significativamente em minha vida acadêmica. Sua paciência, bondade e boa vontade foram determinantes para que eu conseguisse concluir este curso. Seus ensinamentos me mostraram que a vida não é fácil, e me preparou para encarar o futuro fazendo a diferença. Obrigado por sua paciência, seu zelo e por sua influência em minha formação acadêmica, que certamente influenciará positivamente minha carreira profissional.

Não poderia deixar de agradecer também ao professor Edeilson, que com toda sua “transparência” e presteza me orientou em momentos difíceis e me ensinou que não existem coisas impossíveis, mas sim maneiras mais trabalhosas para alcançar o sucesso em algo que se deseja. Muito obrigado, caro amigo “Edy”.

Agradeço a todos professores do curso, Cristina, Fernando, Jack e Parcilene, que sempre contribuíram positivamente em minha formação acadêmica. Hoje entendo que as dificuldades impostas por vocês sempre objetivaram moldar não somente um acadêmico, mas também um profissional e cidadão.

Por fim, agradeço aos colegas de aula e a todos que me ajudaram a vencer mais esta batalha. Muito obrigado.

RESUMO

Alterações na forma e na composição corporal de uma pessoa podem ser causadas por diversos fatores, tais como: doença, comprometimento físico, transtornos alimentares, envelhecimento, dentre outros. Estas alterações resultam em modificações na Percepção Corporal (PC) e fazem com que o indivíduo estabeleça novas relações consigo mesmo e com os demais. Uma das formas de avaliar as modificações na Percepção Corporal é através do *Image Marking Procedure* (IMP), aplicado por um profissional da área da saúde. O objetivo deste trabalho é desenvolver uma aplicação que automatize o IMP utilizando o sensor Kinect, bem como o protocolo a ser seguido para utilização da mesma. Além disto, a aplicação deverá sistematizar a apresentação dos resultados, de forma que o profissional da saúde, geralmente fisioterapeuta, possa aferir o grau de distorção do Esquema Corporal (EC) com maior facilidade, eficiência e eficácia.

PALAVRAS-CHAVE: Saúde, Image Marking Procedure, Kinect.

LISTA DE FIGURAS

Figura 1 - Modelo normal da IEC, segundo Askevold (1975).....	19
Figura 2 – IR e IP sobrepostas com pontos interligados. Adaptado de Thurm (2007, p.35). ..	22
Figura 3 - Protocolo de um sistema de visão computacional	24
Figura 4 - <i>Joints</i> ao longo do corpo, mapeadas nativamente pelo Kinect.	26
Figura 5 - Ângulo de visão do Kinect (CARDOSO, 2013, p.24).....	27
Figura 6 - Ângulo de escuta do Kinect (CARDOSO, 2013, p.144).	28
Figura 7 - Arquitetura da Aplicação	33
Figura 8 - Interface Inicial da Aplicação.	34
Figura 9 - Interface de Resultados da Avaliação	38
Figura 10 - As 20 <i>Joints</i> do esqueleto (MICROSOFT, 2014).	40
Figura 11 - Marcação dos pontos que compõem a Imagem Real através do procedimento tradicional (amarelo) e automatizado (vermelho).	41
Figura 12 - Declaração das variáveis iniciais.	43
Figura 13 - Declaração do Método de inicialização dos comandos de voz para o Kinect.	44
Figura 14 - Método responsável por inicializar a gramática da aplicação.	45
Figura 15 - Método para reconhecimento em tempo real dos comandos de voz.	47
Figura 16 - Método responsável por capturar a posição de alguma articulação.....	48
Figura 17 - Trecho de código responsável por desenhar os pontos e o círculo na mão dos indivíduos.	49
Figura 18 - Método que captura a posição da mão direita do Avaliador e do Paciente.	50
Figura 19 - Método que lança ação para aplicação detectar paciente.	51
Figura 20 - Método para detectar paciente.	51
Figura 21 - Método responsável por detectar o ID do usuário que estiver com as duas mãos acima da cabeça.	52
Figura 22 - Método para obter esqueletos contidos em um quadro de imagem.....	53
Figura 23 - Método para capturar os pontos.....	54
Figura 24 - Método responsável por realizar o cálculo da média dos pontos.	55
Figura 25 - Método que realiza o cálculo do IPC.....	57
Figura 26 - Desenho dos pontos e imagens da tela.....	59
Figura 27 - Método para desenhar pontos no Canvas	60
Figura 28 - Método para realizar a ligação entre os pontos	60

LISTA DE ABREVIATURAS

AN - Anorexia Nervosa

BN - Bulimia Nervosa

EC - Esquema Corporal

IA - Inteligência Artificial

IC - Imagem Corporal

IEC - Imagem e Esquema Corporal

IP - Imagem Percebida

IR - Imagem Real

IMP - *Image Marking Procedure*

PC - Percepção Corporal

QV - Qualidade de Vida

SDK – Software Development Kit

TA - Transtornos Alimentares

VC - Visão Computacional

SUMÁRIO

INTRODUÇÃO.....	12
1 REFERENCIAL TEÓRICO.....	14
1.1 Percepção Corporal.....	14
1.1.1 Imagem Corporal.....	15
1.1.2 Esquema Corporal.....	16
1.1.3 Importância da Percepção Corporal.....	16
1.2 Image Marking Procedure.....	18
1.2.1 Materiais para Aplicação do IMP.....	20
1.2.2 Protocolo do IMP.....	21
1.2.3 Índice de Percepção Corporal.....	22
1.3 Visão Computacional.....	23
1.3.1 Estrutura de um Sistema de Visão Computacional.....	24
1.3.2 Microsoft Kinect Sensor.....	25
1.3.3 Sensor de Profundidade do Kinect.....	27
1.3.4 Sensor de Áudio.....	28
2 MATERIAIS E MÉTODOS.....	29
2.1 Local e Período.....	29
2.2 Materiais.....	29
2.3 Hardware.....	29
2.4 Softwares.....	29
2.5 Métodos.....	30
3 RESULTADOS E DISCUSSÃO.....	32
3.1 Visão geral da Aplicação.....	32
3.1.1 Arquitetura.....	32
3.1.2 AutoIMP.....	34
3.2 Utilizando o Kinect para Automatizar o IMP.....	39

3.3 Implementação.....	42
3.3.1 Inicialização do Sensor e declaração de variáveis chave.....	43
3.3.2 Comandos de Voz.....	44
3.3.3 Detecção da Mão.....	48
3.3.4 Detecção do Paciente.....	50
3.3.5 Captura dos Pontos da Imagem Percebida e Real.....	53
3.3.6 Cálculo da Média dos Pontos Capturados.....	54
3.3.7 Cálculo do IPC e classificação do Paciente.....	56
3.3.8 Projeção dos pontos na tela.....	58
3.4 Protocolo para Avaliação através do Procedimento Automatizado.....	61
4 CONSIDERAÇÕES FINAIS.....	64
REFERÊNCIAS.....	66

INTRODUÇÃO

Tecnologias como o acessório *Microsoft Kinect*, do console *Xbox*, são um exemplo de tecnologia de baixo custo com grande potencial. O *Microsoft Kinect* é um sensor 3D que permite mapear superfícies e reconhecer comandos de voz, possibilitando a interação homem-máquina através de gestos e comandos, dispensando a utilização de *joysticks*. O fabricante disponibiliza o Kit de Desenvolvimento gratuitamente aos desenvolvedores, além de manter uma comunidade que possibilita a discussão e o compartilhamento do conhecimento entre os desenvolvedores (MICROSOFT, 2014), pontuando positivamente o acessório no mercado.

O *Microsoft Kinect* possibilita a captura de imagens com qualidade superior a outros acessórios no mercado, que são, geralmente, mais caros. Isto tem motivado o desenvolvimento de aplicações na área da Visão Computacional – VC – para solução de problemas específicos nas mais diversas áreas do conhecimento, com destaque à área da saúde.

Segundo Mattos (2012), a área da saúde é uma grande consumidora de aplicações baseadas em sistemas de visão computacional. Assim, diversas pesquisas científicas vêm surgindo acerca da utilização do Kinect a favor da saúde, destacando-se as da área da Fisioterapia, na qual tem sido notável o surgimento de pesquisas científicas acerca da utilização do Kinect para auxílio dos profissionais Fisioterapeutas no desempenho de suas funções relacionadas às alterações em partes do corpo humano.

Alterações em partes do corpo humano podem ser causadas por diversos fatores, tais como: doença ou comprometimento físico, transtornos alimentares, chegada da idade, dores crônicas, dentre outros. Segundo Ferreira *et al.* (2012), estas alterações resultam em modificações na Percepção Corporal – PC – e faz com que o indivíduo estabeleça novas relações consigo mesmo e com os demais.

A PC é construída a partir do convívio social, através da vivência e dos valores da sociedade e é influenciada pelas sensações, ou seja, é formada uma estrutura corporal com base em influências do meio social e sensações advindas do sistema nervoso (PEREIRA, 2010). “Estudos que avaliam a Percepção Corporal têm focado suas pesquisas nos dois componentes que a constroem: 1. A imagem corporal, relacionada às atitudes e sentimentos que os indivíduos têm em relação a seu próprio corpo; 2. O esquema corporal, relacionado com a precisão em perceber a dimensão corporal” (THURM, 2012, p.16).

A avaliação do Esquema Corporal – EC – revela a imagem que o indivíduo tem sobre si, em termos de dimensão corporal. Ela é utilizada por especialistas para acompanhamento da evolução do tratamento de patologias específicas, como é o caso da anorexia e bulimia nervosa, síndrome pós-poliomielite, distúrbio no esquema corporal de idosos, postura de pacientes hemiplégicos e hemiparéticos, dentre outras.

Atualmente, um dos procedimentos para se avaliar o EC é o *Image Marking Procedure* – IMP –, proposto por Askevold em 1975. Segundo Thurm (2007), o procedimento utiliza a habilidade do paciente de se projetar no espaço através de um auto desenho, que permite avaliar o EC sem a interferência visual, cognitiva e qualquer referencial externo, fornecendo informações menos tendenciosas (THURM, 2011).

A marcação dos pontos através do IMP é tradicional, sendo necessária a utilização de alguns materiais, a saber: papel *craft*, canetas hidrográficas, canetas esferográficas, régua, esquadro, fita métrica, fita crepe, bastão em centímetros, venda para os olhos e etiquetas para identificação dos indivíduos.

Dado o esforço para realização do procedimento tradicional, uma aplicação que automatize o IMP, isto é, a substituição de alguns dos tradicionais materiais por ferramentas automatizadas, pode trazer benefícios à comunidade, tal como facilitar o trabalho dos profissionais de áreas correlatas.

Visto que a literatura não revela a existência de alguma aplicação relacionada, o objetivo deste trabalho é desenvolver uma aplicação que automatize o procedimento de marcação do esquema corporal utilizando o *Console Kinect* e sistematizar a apresentação dos resultados, de forma que o profissional possa aferir o grau de distorção do EC através da aplicação com mais facilidade. Também faz parte do escopo deste trabalho, a elaboração de um protocolo para aplicação do procedimento automatizado.

O presente trabalho está organizado da seguinte forma: a princípio será apresentado o referencial teórico, que contempla os conceitos básicos para o desenvolvimento do trabalho. Em seguida, a seção 2 apresenta os materiais e métodos utilizados para realização deste trabalho. A seção 3 apresenta os resultados obtidos, com detalhes sobre o desenvolvimento e a aplicação propriamente dita, bem como o protocolo automatizado para aplicação do teste através da aplicação. Por fim, na seção 4 serão apresentadas as considerações finais, seguidas das referências bibliográficas.

1 REFERENCIAL TEÓRICO

Para o desenvolvimento deste trabalho, faz-se necessária a compreensão de conceitos que envolvem o EC e o procedimento de marcação, VC e Tecnologias Computacionais.

A seção a seguir apresenta a PC, conceito necessário para compreensão do EC.

1.1 Percepção Corporal

Thurm (2007) afirma que a PC corresponde ao conjunto de representações neurofisiológicas e espaciais que permite que o indivíduo reconheça conscientemente seu corpo e seus segmentos no tempo e no espaço, com informações vivenciadas e armazenadas no decorrer da vida. A PC dá-se no córtex cerebral, a partir das sensações recebidas pelas vias nervosas e pela vivência emocional.

A PC é construída a partir do convívio social, através da vivência e dos valores da sociedade e é influenciada pelas sensações, ou seja, é formada uma estrutura corporal com base em influências do meio social e sensações advindas do sistema nervoso (PEREIRA, 2010). As informações fornecidas por estes meios são armazenadas no córtex cerebral e formam um mapa neural, localizado na região temporoparietal – também conhecida por córtex somatossensorial, é uma região do cérebro ligada às ações e movimentos e a recepção de sensações através dos sensores, tal como tato - que é alimentado por aferências que interagem com o sistema motor e proporcionam capacidade para o indivíduo reconhecer seu corpo de olhos fechados (THURM, 2007).

“Estudos que avaliam a percepção corporal têm focado suas pesquisas nos dois componentes que a constroem: 1. A imagem corporal, relacionada às atitudes e sentimentos que os indivíduos têm em relação a seu próprio corpo; 2. O esquema corporal, relacionado com a precisão em perceber a dimensão corporal” (THURM, 2012, p.16). Tal divisão se dá devido ao fato de que, quando os conceitos que a envolvem são desmembrados, a capacidade de compreensão dos mesmos é estendida.

Logo, a PC, também conhecida por Imagem e Esquema Corporal – IEC –, pode ser decomposta em Imagem Corporal – IC – e EC, que serão vistos em mais detalhes nas seções que seguem. Uma das definições encontradas na literatura define a IC como a projeção do corpo na mente, ou seja, o modo pelo qual o corpo se apresenta. Em contrapartida, o EC é a percepção da localização precisa e do relacionamento entre as partes do corpo (MOURA *et al.*, 2012).

1.1.1 Imagem Corporal

Os estudos relacionados à IC não são recentes. Askevold (1975) é apresentado como um dos pioneiros quando dissertava sobre o esquema corporal de pacientes com anorexia nervosa – AN – e bulimia nervosa – BN. Posteriormente, surgiram novos estudos e Shilder (1994) definiu a IC como sendo a representação do corpo na mente, contemplando aspectos cognitivos, crenças e expectativas do indivíduo em relação ao seu próprio corpo e ao estado emocional. Em outras palavras, a forma como o sujeito se vê e como ele gostaria de se ver ou ser visto pelo outro de acordo com a aceitação e julgamento do próprio corpo (THURM, 2012).

Em outra definição, Ferreira *et al.* (2012, p.51) afirma que “a imagem corporal é a percepção do indivíduo em relação a sua posição de vida, no contexto da cultura e sistema de valores nos quais ele vive e em relação a seus objetivos, expectativas, padrões e preocupações”. Ou seja, além da IC ser uma representação mental do corpo, ela ultrapassa sua aparência real, pois o indivíduo mentaliza sua imagem. Isto reflete em sua experiência de vida, envolvendo suas crenças, sentimentos, atitudes, impressões e posição social. Assim, há uma integração entre os aspectos fisiológicos, neural e emocional, não desabonando o contexto social (BARROS, 2005; SOUTO, 1999 *apud* MOURA *et al.*, 2012).

Em outras palavras, a IC é a projeção do corpo armazenada na mente, que representa mentalmente os contornos e a forma do corpo, acrescidos de sentimentos de satisfação e insatisfação a estas características. Estes sentimentos são influenciados por experiências vividas, pelo juízo de valores, por estímulos visuais que, em sua maioria, são provenientes da mídia e da cultura no qual está inserido (THURM, 2007). Dessa forma, a IC está diretamente relacionada à consciência que o indivíduo tem sobre seu corpo, sendo influenciada pela forma de ver, pensar e agir, considerando o contexto social do qual está inserido, que contempla as regras e valores sociais e morais (FERREIRA *et al.*, 2012).

Um estudo realizado por Barbosa (2004) aponta que a IC é composta por diversos fatores que, em conjunto, propiciam o autoconhecimento do próprio corpo. Os fatores podem ser classificados em anatômicos, funcionais e subjetivos. Os fatores anatômicos envolvem o formato, percepção e aparência do corpo, enquanto os fatores funcionais contemplam a capacidade de gerenciar sua própria vida. Por fim, os fatores subjetivos envolvem a classe social, renda, gênero e idade que levam determinados grupos a valorizar certas áreas ou funções corporais (MOURA, 2012).

Assim, alterações em uma parte do corpo por motivo de doença ou comprometimento físico e alterações psicológicas que envolvam o indivíduo influenciarão diretamente em modificações na PC, o que implicará em novas relações na sociedade e consigo mesmo (FERREIRA *et al.*, 2012).

1.1.2 Esquema Corporal

O EC é um conceito que envolve a construção do próprio corpo na mente a partir de aspectos que envolvem a IC, acrescido da percepção da localização de cada segmento corporal através de aferências multimodais, tais como visão, tato, propriocepção e suas conexões com o sistema motor. Salzano (2011, p. 2) afirma que “o esquema corporal corresponde à representação dos segmentos corporais na dimensão espacial, originado por estímulos visuais, proprioceptivos, motores e táteis, sendo projetado e armazenado no córtex cerebral”. Assim, o EC corresponde ao componente perceptual ligado à dimensão do corpo, no que tangencia tamanho e forma, somado à representação dos segmentos corporais e a relação com o meio ao redor.

Apesar de semelhantes, a IC e EC se diferem, mas juntas compõem a PC. Enquanto a IC está diretamente ligada ao sistema pessoal emocional, o EC está relacionado ao sistema neural do indivíduo, sob responsabilidade da região cortical, a área temporoparietal (THURM, 2012).

Em suma, o EC corresponde à representação das partes do corpo no espaço e permite ao indivíduo reconhecer estas partes e a interação que elas têm com o meio, originando a percepção de sua dimensão corporal. Deve-se ressaltar que o EC depende dos valores e conceitos subjetivos da IC, pois é a representação corporal do indivíduo na mente, que é um fator chave para conhecimento dos segmentos corporais, que implicam diretamente na PC do indivíduo.

1.1.3 Importância da Percepção Corporal

A PC é influenciada por diversos fatores que a compõem como, por exemplo, a vivência e os valores da sociedade, bem como as sensações advindas do sistema nervoso. Logo, fatores emocionais, crenças, expectativas de vida, visão, tato, propriocepção e o sistema motor são variáveis que compõem e implicam na manutenção da PC. Assim, ações que envolvam alguma destas variáveis implicarão na mudança da PC como, por exemplo, o movimento e atividade física ao longo da vida, que desempenham fortes alterações nestes fatores e

implicam diretamente na Qualidade de Vida – QV – do indivíduo (PEREIRA, 2010; FONSECA, 2012).

Com o avançar da idade as dimensões corporais tendem a se tornarem aquém das medidas reais, ou seja, o indivíduo enxerga-se menor do que realmente é (PEREIRA, 2010). Esta afirmação pode ser refutável, considerando que, ao longo de sua vida, embora esteja com idade avançada, o indivíduo pode sofrer alterações positivas em seu EC, quando submetido à alimentação balanceada e constante atividade física. Assim, é possível perceber alterações no EC, conseqüentes do avanço da idade, que podem ser estimuladas e melhoradas (FONSECA, 2012).

Um estudo realizado em idosos mostra que a IEC também está diretamente relacionada à capacidade motriz e equilíbrio. Isto é, a qualidade de vida destes idosos (NOGUEIRA, 2012).

Em outro estudo, realizado por Thurm (2012), foi constatado que sujeitos com transtornos alimentares - TA, como os já citados AN e BN, apresentam como sintoma marcante o distúrbio da PC e uma insatisfação com o próprio corpo, pois, de modo geral, têm uma percepção além do real, ou seja, se veem e se sentem maiores do que realmente são.

A AN caracteriza-se por perda de peso intensa e intencional à custa de dietas rígidas e métodos purgativos na busca desenfreada pela magreza e por distorção da imagem corporal. Os pacientes acometidos por este transtorno apresentam insatisfação com seus corpos e se sentem obesas, apesar de magras, fato considerado como uma alteração da sua PC (THURM, 2012).

A BN se caracteriza quando o indivíduo ingere uma grande quantidade de alimentos, geralmente desenfreada (episódios bulímicos). Para contornar a situação e compensar o episódio, o indivíduo tem ações como o vômito autoinduzido e o uso de laxantes. “O efeito imediato provocado pelo vômito é o alívio do desconforto físico secundário a uma hiperalimentação e, principalmente, a redução do medo de engordar” (THURM, 2012, p.15).

Assim, a análise do EC tem um papel fundamental no tratamento de pacientes com TA, pois permite aferir o grau de distorção da PC do indivíduo em tratamento, permitindo que seja feito um acompanhamento para constatar a evolução do tratamento. Deve-se ressaltar que, de modo geral, não existe um profissional específico que cuide do distúrbio da

alimentação envolvendo a PC, pois é uma área multiprofissional, que envolve profissionais da área da Psicologia, Educação Física, Fisioterapia, dentre outras (THURM, 2012, p.15).

A Análise da IEC em pacientes que já foram acometidos de poliomielite pode evidenciar a síndrome pós-poliomielite, que está diretamente relacionada à capacidade motriz do indivíduo (FERREIRA, 2012).

Outro estudo revela que atletas de alto rendimento apresentam alterações significativas no EC, influenciada principalmente pela dor crônica, apresentado distorção em todos os segmentos corporais (THURM, 2007). Além disso, o esporte de alto rendimento é um conjunto complexo de elementos que envolvem o atleta, o técnico, o espectador, a torcida e os patrocinadores e, neste contexto, o atleta é o principal foco que será cobrado pela equipe esportiva e pela comissão técnica a alcançar a vitória. Para isso o atleta de alto rendimento foca seu objetivo na superação do próprio limite (RUBIO, 2004), que implica significativamente na PC, envolvendo psicologicamente o indivíduo.

“Distúrbios na IEC são sequelas comuns em hemiplégicos/hemiparéticos onde a dominância hemisférica (lobo parietal) relaciona-se com a imagem corporal e a capacidade perceptiva” (MARCOTTI, 2005; SILVA *et al.*, 2002 *apud* MOURA, 2012, p.2). Dessa forma, a análise da PC também permite aferir o grau de distorção da IEC, fornecendo subsídios para o tratamento.

Em suma, um sistema de PC íntegro é fator chave para uma boa formação da IC. Logo, a percepção da IC satisfatória é coadjuvante de uma boa simetria corporal (MOURA, 2012, p.11). Assim, na maioria dos casos, a análise da PC é utilizada para acompanhar o rendimento ou evolução do tratamento do qual o indivíduo está sendo submetido e assim poder identificar e mensurar os resultados para, juntamente com outros procedimentos, nortear a aplicação do tratamento.

1.2 Image Marking Procedure

As origens sobre avaliações da IEC não são conhecidas. Askevold (1975) foi, provavelmente, um dos primeiros a propor uma forma de visualizar e identificar possíveis alterações na IEC de um indivíduo.

Em seus estudos, Askevold reuniu uma equipe de fisioterapeutas e psicólogos para ser um grupo de referência, sugerindo que fosse seu procedimento bem como as medidas fossem um padrão normalidade, considerando que seu grupo teria uma boa consciência corporal pelo

trabalho que exerciam (THURM, 2007). Nas pesquisas sobre IEC, os sujeitos foram avaliados por meio de um auto desenho feito a partir de pontos anatômicos específicos, estimulados pela capacidade do indivíduo de entrar em contato com a estimulação presente no ambiente circundante. “Este procedimento foi chamado de *Image Marking Procedure*, ou Procedimento de Marcação do Esquema Corporal” (THURM, 2007, p.27).

Estudos de Thurm (2007), Ferreira (2012) e Salzano (2011), revelam que o IMP é um estudo direto e compatível para a análise do EC de modo objetivo. O IMP utiliza a habilidade do paciente de se projetar no espaço através de um auto desenho, que permite avaliar o EC sem a interferência visual, cognitiva e qualquer referencial externo, fornecendo informações menos tendenciosas. Por este motivo, ele se diferencia dos demais métodos de avaliação que fornecem informações subjetivas por usar a percepção não corporal, como dito por Thurm (2007, p.27): “Já a avaliação do esquema corporal obtida por meio de questionários e entrevistas fornece informações subjetivas, por utilizar a percepção não corporal”.

A Figura 1 apresenta o modelo normal da IEC, marcado através do IMP.

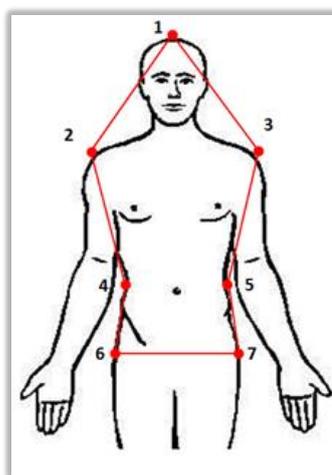


Figura 1 - Modelo normal da IEC, segundo Askevold (1975).

Na Figura 1 é apresentado o modelo normal da IEC, na qual os pontos enumerados correspondem as localizações de partes do corpo humano, apresentadas no Quadro 1.

Quadro 1 - Localização dos pontos do Teste de Askevold (1975, apud MOURA et al, 2012, p.4)

1	Ápice da cabeça
2	Articulação acrômio-clavicular esquerda
3	Articulação acrômio-clavicular direita
4	Curva da cintura esquerda (acima dos ápices das cristas ilíacas E)
5	Curva da cintura direita (acima dos ápices das cristas ilíacas D)

6	Trocanter maior do fêmur esquerdo
7	Trocanter maior do fêmur direito

Existem outros procedimentos de avaliação do EC que fornecem resultados semelhantes ao IMP, como é o caso do *Kinesthetic Size Estimating Apparatus*, ou Aparato Sinestésico de Estimativa de Largura Corporal (LAUTENBACHER E COLS., 1997), que consiste em “um bastão graduado em centímetros em que o sujeito deverá mostrar, por meio do posicionamento das mãos no bastão, a largura dos pontos anatômicos tocados em seu corpo, que são os mesmos tocados no IMP” (THURM, 2007, p.28).

Atualmente existem softwares que integram funcionalidades específicas para análise do EC, através do uso de câmeras específicas, como é o caso da Técnica de Distorção de Vídeo (*Vídeo Distortion Technnique*) em que o sujeito é fotografado ou filmado e a imagem é reproduzida no monitor, onde a dimensão da imagem é distorcida e o sujeito deve reconhecer a imagem que melhor representa seu corpo.

Em suma, os procedimentos para avaliação do EC são, por vezes, menos precisos que o IMP. Tal imprecisão, aliada ao alto custo por razão da aparelhagem e softwares, acabam tornando estes procedimentos inviáveis (THURM 2007), favorecendo a utilização do IMP, que pode ser realizado com a utilização de materiais de baixo custo e fácil acesso.

1.2.1 Materiais para Aplicação do IMP

Atualmente, a marcação dos pontos através do IMP se mantem de forma tradicional, tendo em vista não existirem tecnologias acessíveis para automatização do processo. Assim, para aplicação do protocolo é necessário a utilização de alguns materiais, são eles: papel *craft*, canetas hidrográficas, canetas esferográficas, réguas, esquadro, fita métrica, fita crepe, bastão em centímetros, venda para os olhos e etiquetas para identificação dos indivíduos.

De posse dos materiais necessários, o procedimento deve ser aplicado seguindo protocolos existentes, tal como feito por Ferreira (2012), que avaliou a Análise da IEC em indivíduos com diagnóstico de síndrome de pós-poliomielite e Thurm (2007), que avaliou os efeitos da dor crônica em atletas de alto rendimento em relação ao EC, Agilidade Psicomotora e estados de Humor.

1.2.2 Protocolo do IMP

O protocolo proposto por Askevold (1975) propõe, a princípio, informar o sujeito sobre o funcionamento do procedimento. Em seguida, o mesmo é marcado com lápis dermatográfico nas regiões corporais que correspondem aos pontos a serem analisados, a saber: articulações acrômio-clavicular direita e esquerda, curvas da cintura direita e esquerda, trocanteres maiores do fêmur direito e esquerdo. A marcação é importante por garantir que, quando da captura da Imagem Percebida – IP, os pontos sejam tocados no mesmo lugar, o que garante uma precisão maior para análise dos dados.

Em seguida, o paciente permanece em pé diante de uma folha de papel *craft* fixada à parede a uma altura de um palmo acima de sua cabeça. A distância do mesmo até o papel é determinada pelo comprimento de seu membro superior semiflexionado de forma que sua mão atinja o papel. Com isso, o paciente é orientado a realizar o teste de olhos vendados com instrução verbal de que deve imaginar que a folha de papel é um espelho e estaria se vendo nele. Assim, os pontos marcados podem ser tocados e o indivíduo projeta no papel diante de si, utilizando caneta preta, o ponto tocado.

Para melhor precisão nas medidas referentes à captura da IP, deve ser realizada mais de uma marcação sem que o indivíduo visualize a marcação anterior, evitando interferências visuais no procedimento, de forma que a marcação final do ponto seja a média entre as marcações. Por fim, o indivíduo é posicionado de costas próximo ao papel, para que sejam marcados, com caneta em outra cor, os pontos referentes à Imagem Real – IR. Para evitar erros de marcação dos pontos reais, o profissional pode estabilizar a região cervical com o apoio da mão impedindo o balanço postural que é a oscilação natural que o corpo apresenta quando está em postura ereta, conforme realizado por Thurm (2007).

Com a marcação dos pontos referente às IP e IR, pode ser feita a ligação entre os pontos, tornando visível a dimensão corporal, conforme apresentado na Figura 3.

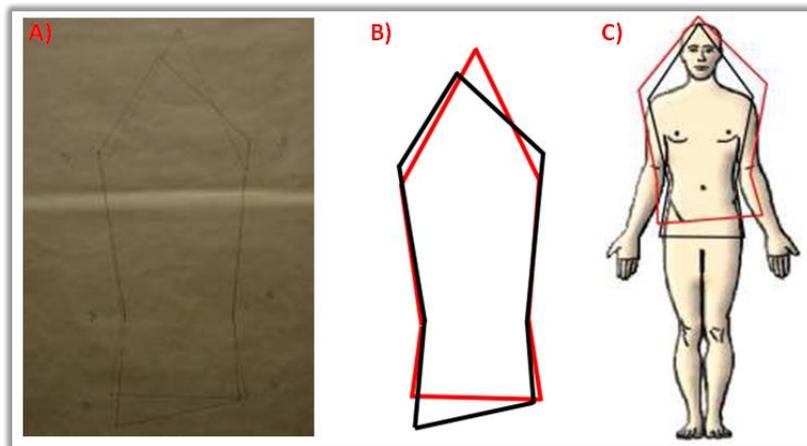


Figura 2 – IR e IP sobrepostas com pontos interligados. Adaptado de Thurm (2007, p.35).

Na Figura 3, o lado A mostra os pontos ligados, sobre um papel *craft*, enquanto o lado B apresenta as marcações sobrepostas digitalizadas. Por fim, o lado C permite a visualização dos desenhos referentes às IP e IR em relação ao corpo de um indivíduo.

O tamanho percebido e o tamanho real servem para avaliar o grau de distorção da Percepção Corporal de um Indivíduo. A sobreposição dos tamanhos, tal como na Figura 3, permite visualizar a disparidade entre os tamanhos sem a inferência de cálculos matemáticos, o que já pode indiciar distorção na IEC.

A disparidade entre os tamanhos pode ser calculada através do Índice de Percepção Corporal – IPC, o qual fornece um valor percentil que possibilita identificar o grau de distorção da IC, apresentado na seção a seguir.

1.2.3 Índice de Percepção Corporal

Segundo Askevold (1975), o IPC consiste em aplicar uma fórmula matemática sobre os dados referentes à avaliação do IMP. A fórmula para cálculo do IPC é representada pela média das três aferências do tamanho percebido pelo indivíduo dividido pelo tamanho real multiplicado por 100. O resultado da fórmula é expresso em porcentagem e representa a diferenciação do IPC dos segmentos corporais, envolvendo a IP e a IR (NOGUEIRA, 2012). Assim, o IPC permite avaliar o grau de distorção da PC de um indivíduo.

Salzano (2011), com base nos estudos de Askevold (1975), resume o uso IPC para classificar a PC de um indivíduo. Bonnier (1905) sugeriu as seguintes classificações para a PC: os sujeitos que se percebem maior do que realmente são, ou seja, acima de 100%, são classificados como hiperesquemáticos. Os indivíduos que se percebem menor do que

realmente são, ou seja, abaixo de 100% são considerados hipoesquemáticos. Os indivíduos que se percebem 100% são considerados normais.

Segheto *et al.* (2010) sugeriram uma nova classificação, considerando que a classificação sugerida por Bonnier (1905) não se enquadrava à realidade das avaliações aplicadas, pois dificilmente seria encontrado algum indivíduo classificado como normal, devido sempre existir diferença entre a IP e a IR, embora insignificante.

A classificação sugerida por Segheto *et al.* (2010) foi baseada na análise de percentis. Assim, os sujeitos que se perceberam entre 99,4% e 112,3% foram considerados com PC adequada; valores abaixo de 99,4% foram classificados com hipoesquematia e os acima de 112,3%, como hiperesquematia.

1.3 Visão Computacional

A VC, também conhecida por Visão Robótica, é a ciência responsável pela visão de uma máquina, compreendida pela interpretação do meio visualizado pela máquina, que permite extrair informações significativas capturadas por sensores de imagem e vídeo. As informações permitem à máquina a identificação de padrões em uma determinada imagem, reconhecendo objetos, bem como a possibilidade de classificação, categorização, manipulação e pensamento sobre o mesmo (MILANO; HONORATO, 2014).

Atualmente, a VC é considerada uma ciência relativamente recente e multidisciplinar. Milano e Honorato (2014) mencionam em sua publicação que uma das primeiras menções sobre VC foi através de Selfridge (1955), ao afirmar que a visão computacional é “olhos e orelhas do computador”, sendo compreendida como uma semelhança com o modo de percepção humana do meio, o que a torna uma ferramenta poderosa, no sentido de possibilitar o reconhecimento do ambiente em volta da máquina e, a partir de processamento, agir conforme alguma decisão tomada.

A transformação do conceito de visão humana em visão de uma máquina é uma tarefa difícil e dispendiosa no que tange os recursos computacionais. Entretanto, a forte evolução da capacidade de processamento dos computadores tem proporcionado um crescimento notável da visão computacional, à medida que vem proporcionando o processamento em tempo real, o que explica o motivo de ser uma ciência nova.

Em suma, a visão computacional tem integrado as áreas de Processamento de Imagens e Inteligência Artificial – IA, objetivando a criação de algoritmos capazes de interpretar o

conteúdo visual de imagens. A sua utilização é evidente em diversas áreas do conhecimento, principalmente as do segmento tecnológico, que envolvem a agronomia, astronomia, biologia, medicina, dentre outras. A sua finalidade é a análise de imagens, reconhecimento de padrões e controle inteligente, com aplicações práticas (NEVES, 2012).

Identificada a importância da VC para o desenvolvimento deste trabalho, a seção a seguir apresenta a estrutura de um sistema de VC.

1.3.1 Estrutura de um Sistema de Visão Computacional

Os Sistemas que trabalham com VC são criados para realização de atividades ou problemas específicos, que geralmente são baseados em sistemas especialistas, que também foram desenvolvidos para um problema específico. Dessa forma, Milano e Honorato (2014, p.2) acreditam que não há um padrão específico para basear a arquitetura de um sistema de visão computacional.

Jungong *et al.* (2013) afirmam que um problema recorrente para todo Sistema de VC é o mapeamento de superfícies, que é o mapeamento de objetos reais, gerando um modelo de representação visual dos mesmos. Assim, Milano e Honorato (2014, p.2) concluíram que os sistemas seguem um protocolo que descrevem algumas etapas em comum, que são: Aquisição de Imagem, Pré-Processamento, Extração de Características, Detecção de áreas de interesse e Processamento de Alto Nível, apresentados na Figura 3.

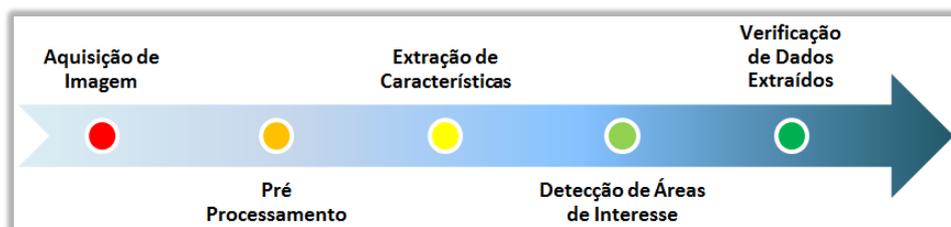


Figura 3 - Protocolo de um sistema de visão computacional

A aquisição de imagem é o processo de aquisição de uma imagem ou de um conjunto de imagens a partir de sensores de câmeras, no qual os pixels de cada imagem indicam coordenadas de luz e propriedades físicas. Dentro desta etapa deve-se destacar o *Microsoft Kinect Sensor*, que é um *Scanner 3D* que tem sido utilizado nos Sistemas de VC, não por ser o melhor existente no mercado, mas sim por ser uma alternativa financeiramente viável em meio aos outros sensores (JUNGONG *et al.*, 2013; YAMADA *et al.*, 2013; COSTA, 2011).

A etapa de pré-processamento é o processo realizado antes de obter informações de uma imagem, de forma a aplicar métodos específicos que facilitem a identificação de um objeto, tais como, destaque de contornos, bordas e figuras geométricas. Com a imagem pré-processada, é possível a extração de características matemáticas, como textura, bordas, formato, tamanho e movimento.

A detecção e segmentação é o processo realizado para destacar regiões relevantes da imagem, segmentando-as para processamento de alto nível. Esta etapa desempenha um papel importante, pois além de retirar partes desnecessárias ao processamento, minimiza quantidade de recursos computacionais necessários para o processamento de alto nível.

Por fim, o Processamento de alto nível inclui a precisão e validação dos dados objetivos, o que permite estimar os parâmetros sobre a imagem e classificação dos objetos contidos em diferentes categorias.

No que tange às ferramentas para digitalização, entende-se que o Kinect, apresentado na seção a seguir, é um Sistema de VC que foi desenvolvido com objetivo de permitir a interação natural do homem com a máquina. Para que isto se tornasse possível, foi necessário o uso de diversas tecnologias, dentre elas, redes neurais. Assim, os Sistemas de VC são desenvolvidos para uma finalidade específica e, de acordo com cada domínio, as etapas do processo têm suas adaptações.

1.3.2 Microsoft Kinect Sensor

No dia 4 de outubro de 2010, nos Estados Unidos foi lançado pela Microsoft o Kinect, acessório do console Xbox 360 que, através de sistemas de áudio com quatro microfones captam a voz do usuário, permitindo reconhecimento de fala e comando, possuindo também um sistema de imagem e profundidade com câmeras RGB comum com resolução 640x480 com emissores de feixes infravermelhos e outra infravermelha que capta os feixes proporcionando o reconhecimento de profundidade e um sistema motor que possibilita a rotação vertical para que possa acompanhar o posicionamento e deslocamento do jogador, reconhece movimentos dentro de um determinado espaço, interpretando e transformando esses movimentos em comandos para que haja interação entre o console e o usuário através de interface natural, que dispensa uso do controle (ROCHA, 2012; CARDOSO, 2013; MICROSOFT, 2014).

Rocha (2012) apresentou também a possibilidade de utilizar o Kinect como ferramenta no atendimento fisioterapêutico de pacientes neurológicos, vindo que os jogos que são utilizados no Kinect podem interferir nas capacidades biomotoras. Já Nunes (2011) propôs um estudo onde via a possibilidade do Kinect como ferramenta de extração de características afetivas do usuário, estudando como os computadores poderiam reconhecer e responder as emoções humanas e dessa forma o pudesse também despertar emoções em humanos. A proposta do pesquisador era utilizar o Kinect em um estudo de caso, extrair linguagens corporais do usuário, distinguindo sua personalidade, sobretudo envolvendo questões de motivação para o jogador.

Atualmente, o Kinect permite o reconhecimento e a interação simultânea de até seis usuários (MICROSOFT, 2014). Um usuário detectado pelo Kinect é mapeado em articulações, que constituem um esqueleto segmentado, onde cada final de um segmento é conhecido por *Joint*. Assim, o corpo humano é dividido em vinte *Joints*, sendo elas: cabeça, ombro central (pescoço), ombros direito e esquerdo, cotovelos direito e esquerdo, pulsos direito e esquerdo, mãos direita e esquerda, espinha, lados direito, centro e esquerdo do quadril, joelhos direito e esquerdo, calcanhares direito e esquerdo e pés direito e esquerdo (MICROSOFT, 2014). A Figura 7 apresenta o mapa das *Joints* ao longo do corpo.

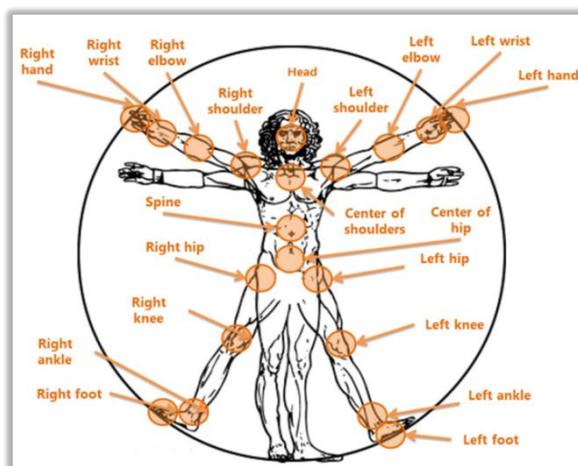


Figura 4 - Joints ao longo do corpo, mapeadas nativamente pelo Kinect.

Cada uma das *Joints* apresentadas na Figura 7 tem sua coordenada tridimensional no espaço visualizado pelo Kinect. Assim é possível saber a localização de alguma *Joint* específica, se a mesma estiver no campo de visão do Kinect, apresentado na Figura 5.

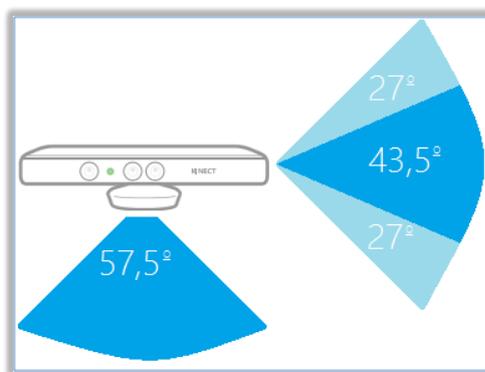


Figura 5 - Ângulo de visão do Kinect (CARDOSO, 2013, p.24).

Na Figura 5, percebe-se que campo de visão do Kinect dá-se de forma piramidal a partir do acessório, limitado a 57,5° graus na horizontal e 43,5° na vertical (CARDOSO, 2013; COSTA, 2013; MICROSOFT, 2014). O eixo vertical é motorizado e permite ao Kinect ser regulado com mais ou menos 27°.

O campo de visão do Kinect é diretamente ligado com a profundidade com que o alvo a ser identificado se encontra. Isto é, o ângulo vertical de 43,5° faz com que o campo de visão se estenda conforme a profundidade. Contudo, a distância horizontal máxima a ser considerada para que os dados sejam capturados com maior precisão é de 4 metros, que garante que indivíduos de diversas alturas sejam identificados (MICROSOFT, 2014).

1.3.3 Sensor de Profundidade do Kinect

Como já dito na seção anterior, o Kinect possui sensores que permitem a identificação da dimensão de profundidade. Seu funcionamento é possível através de um emissor e uma câmera de luz infravermelha, que possibilita ao Kinect obter imagens tridimensionais independente da condição de luminosidade do ambiente. A luz emitida pelo projetor é passada por uma grade de difração, fazendo com que a luz emitida se transforme em pequenos pontos, que são captados pela câmera. Assim, a profundidade é medida pela distância entre o sensor e o objeto (COSTA, 2013).

A utilização do sensor de profundidade do Kinect em conjunto com a câmera *RGB* contribui para solução do problema recorrente em um sistema de VC, que é o mapeamento de superfícies e reconhecimento de objetos no ambiente visualizado.

Assim, a etapa de Aquisição de Imagem do protocolo de um sistema de VC tem se tornado uma tarefa menos dispendiosa, tanto em esforços de desenvolvimento quanto recursos computacionais, já que o Kinect oferece uma maneira eficiente para Aquisição de Imagem.

1.3.4 Sensor de Áudio

O sensoriamento de áudio do Kinect se baseia em um controlador com quatro microfones distribuídos unidimensionalmente dentro de si. Através destes é possível gravar áudio, bem como encontrar a localização da fonte de som e a direção da onda de áudio, que permitem identificar qual a posição do indivíduo que lançou o comando de voz. O ângulo de escuta do Kinect é de 100 graus horizontais, conforme apresentado na Figura 6.

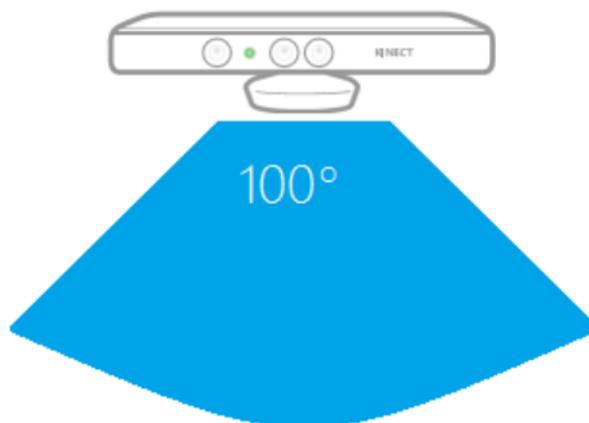


Figura 6 - Ângulo de escuta do Kinect (CARDOSO, 2013, p.144).

O SDK foi programado para fazer com que o Kinect se adapte nativamente ao ambiente, através do cancelamento de eco acústico e supressão de ruídos. Isto permite que eventuais ruídos não venham atrapalhar na qualidade do som captado, melhorando a precisão na interpretação dos comandos de voz e, conseqüentemente, a experiência para com o usuário.

Em suma, o Kinect permite que seus sensores trabalhem todos em conjunto, ou seja, é possível utilizar o sensor de profundidade, RGB e áudio juntos, permitindo que as aplicações desenvolvidas se tornem mais naturais ao usuário, proporcionando além de movimentos, comandos de voz que ajudam na dispensação eventuais controles remotos, tornando a utilização da aplicação mais intuitiva e atraente.

2 MATERIAIS E MÉTODOS

Nesta seção serão apresentados o local e o período previsto para o desenvolvimento da aplicação e suas funcionalidades, bem como os recursos de hardware e software que foram utilizados.

2.1 Local e Período

Para o desenvolvimento deste trabalho, foram utilizados o Laboratório de Multimídia e Computação Gráfica (LABMIDIA) em conjunto com o Laboratório de Tecnologia em Saúde (LTS), ambos do Centro Universitário Luterano de Palmas (CEULP/ULBRA), vinculados ao grupo de Pesquisa, Tecnologia, Saúde e Qualidade de Vida, no qual se objetivou desenvolver uma aplicação que automatize o IMP, proposto por Askevold em 1975, utilizando o Kinect.

O presente trabalho foi desenvolvido como requisito parcial para a disciplina de “Trabalho de Conclusão de Curso em Sistemas de Informação II (TCC II)” no decorrer do segundo semestre de 2014.

2.2 Materiais

Compreendidos como recursos de *hardware* e *software*, os materiais utilizados para elaboração do trabalho são de propriedade particular ou disponibilizados pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA) por meio de seus laboratórios. Os outros recursos foram obtidos gratuitamente na Internet.

2.3 Hardware

Os recursos de *hardware* que foram utilizados para implementação do trabalho são um notebook, um Sensor *Kinect* do Xbox e um adaptador do sensor Kinect para computador ou notebook. O notebook e o *Kinect* são de propriedade particular, sendo que estes são interligados através do adaptador disponibilizado pela Ulbra.

2.4 Softwares

Os *softwares* que foram utilizados para realização deste trabalho são listados a seguir:

- *Microsoft Windows 7*: é um sistema operacional presente em boa parte dos computadores pessoais. Sua utilização é imprescindível, já que é a base para instalação de ferramentas de desenvolvimento e drivers do Sensor Kinect.
- *Microsoft Kinect SDK 1.8*: disponibilizado pela própria Microsoft, é um pacote de ferramentas de desenvolvimento de aplicativos para o *Kinect*. A SDK

permite a utilização de C++, C# ou Visual Basic para criar aplicativos e experiências que suportam gestos e reconhecimento de voz utilizando o Kinect no Windows. Atualmente, a última versão é a 2.0, a qual suporta apenas a segunda geração do dispositivo Kinect (MICROSOFT, 2014).

- *Visual Studio 2012*: trata-se de um pacote de programas da Microsoft para desenvolvimento de aplicações com base na plataforma .NET Framework (MICROSOFT, 2014).

2.5 Métodos

O presente trabalho é uma pesquisa aplicada a fins práticos, no qual se objetiva desenvolver uma aplicação que automatize o IMP proposto por Askevold em 1975 utilizando o *Microsoft Kinect Sensor*.

Para que isto fosse possível, a primeira etapa deste trabalho foi realizar um procedimento metodológico baseado em pesquisa bibliográfica, disponível em livros, artigos de revista, trabalhos de conclusão de curso, dissertações de mestrado e teses de doutorado para escrita do referencial teórico, acerca dos conceitos sobre o procedimento de marcação IMP, EC, VC.

Concluídas as pesquisas acima citadas, foi feito um levantamento bibliográfico acerca dos conceitos computacionais que possibilitariam o desenvolvimento da aplicação. Esses conceitos envolvem desde os recursos que permitem a aplicação se comunique com o Kinect –SDK –, bem como a IDE de desenvolvimento, linguagem de programação e componentes que facilitam o desenvolvimento, também conhecidos como bibliotecas.

As pesquisas realizadas foram norteadas através da colaboração de um profissional fisioterapeuta que, além de explicar conceitos detalhados sobre a aplicação do procedimento, o mesmo colaborou indicando os trabalhos mais relevantes, bem como dicas e sugestões para desenvolvimento do trabalho.

Ao final deste estudo, foi elaborada a revisão de literatura na qual foram apresentadas as definições de conceitos relacionados ao IMP, EC, VC, bem como o detalhamento das Tecnologias Computacionais que seriam necessárias para o desenvolvimento do trabalho.

A partir dos estudos citados, foi possível iniciar o desenvolvimento gradativo da aplicação, isto é, as funcionalidades foram incrementadas parcialmente, permitindo que correções e melhorias fossem realizadas ainda na etapa de desenvolvimento. O início do

desenvolvimento se deu a partir de uma biblioteca inicial contendo os métodos indispensáveis para funcionamento do Kinect acoplado ao computador, que contemplam funções básicas, tal como configuração do *driver*, comunicação com o computador, movimentação do eixo motorizado, ativação dos sensores *RGB* e *Depth*, reconhecimento do indivíduo, dentre outras.

Com as funcionalidades básicas implementadas, foi utilizado um recurso nativo da SDK de desenvolvimento do Kinect, o mapeamento dos pontos do corpo humano, que dispensa esforços de desenvolvimento para reconhecer partes do corpo. Os pontos oferecidos nativamente foram analisados e utilizados como referência para marcação da IC no que tange a IR, papel anteriormente cumprido pelo profissional avaliador.

Contudo, constatou-se que os pontos utilizados não forneciam subsídios suficientes para que o profissional avaliador pudesse realizar a avaliação com a devida precisão, fornecendo dados incorretos, já que os pontos não eram os mesmos utilizados em um procedimento de avaliação tradicional utilizando o IMP.

Então, foi definida outra metodologia para captura e análise dos pontos para composição da IR, na qual foi desabonada a utilização dos pontos oferecidos nativamente pelo Kinect. Neste caso, passou-se a utilizar apenas os pontos marcados por meio das mãos do avaliador e do paciente, que forneciam respectivamente a IR e a IP.

Posteriormente foi desenvolvida a interface para interação do avaliador com a aplicação, a fim de possibilitar sua utilização em caráter de teste. A interface permitiu que a aplicação pudesse ser utilizada para testes iniciais.

Assim, para que o procedimento automatizado pudesse ser utilizado pelo avaliador, foi definido o protocolo de avaliação automatizada, que é um roteiro de procedimento de avaliação que contempla as regras e especificações a serem seguidas pelo avaliador e paciente na aplicação do procedimento, objetivando a correta utilização da aplicação para obtenção dos resultados.

A seção a seguir apresentará os resultados obtidos com o desenvolvimento deste trabalho.

3 RESULTADOS E DISCUSSÃO

A presente seção trata dos resultados obtidos na realização deste trabalho. Inicialmente será apresentada uma visão geral da aplicação de acordo com sua arquitetura e funcionamento. Posteriormente, serão apresentados detalhes acerca da codificação da aplicação, tangenciando as funcionalidades essenciais. Por fim, será apresentado o protocolo automatizado para utilização da aplicação desenvolvida para análise do esquema corporal, que contém as regras e o passo a passo de utilização, objetivando a obtenção de resultados mais satisfatórios.

3.1 Visão geral da Aplicação

Para melhor entendimento, esta seção foi subdividida em Arquitetura da Aplicação e a Aplicação propriamente dita, denominada de AutoIMP. A subseção de Arquitetura sistematizará o funcionamento da aplicação de uma forma geral, enquanto a subseção da aplicação apresentará a aplicação, suas funcionalidades e interfaces.

3.1.1 *Arquitetura*

O Kinect permite a visualização do ambiente e a identificação dos usuários, que interagem com a aplicação através de movimentos e comandos de voz. A arquitetura geral é descrita na Figura 7, no qual o Item 3 é o núcleo do sistema, que permite a interação dos usuários com o a aplicação que, por sua vez, é baseada em tecnologias e regras que permitem processar as entradas dos usuários e apresentar o resultado.

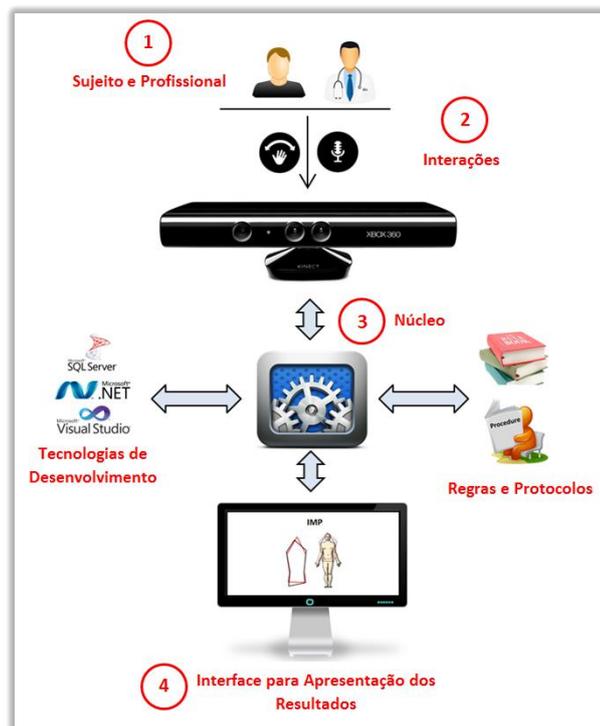


Figura 7 - Arquitetura da Aplicação

No Item 1 da Figura 7, percebe-se que a aplicação permite que o sujeito a ser avaliado e o profissional sejam identificados, uma vez que estes permanecem em posição previamente estabelecida, obedecendo os limites do campo de visão do Kinect.

No Item 2 são apresentadas as interações dos indivíduos com o Kinect através de movimentos e comandos de voz. Além do reconhecimento do ambiente e identificação dos indivíduos, o Kinect identifica a posição das mãos do profissional e do sujeito. À mão do profissional, quando tocada em um ponto específico, dá-se o nome de Ponto referente à Imagem Real, análogo à marcação real através procedimento tradicional.

Quando tocado pelo profissional, o sujeito a ser avaliado projeta em direção ao Kinect, com sua mão, a posição percebida, a qual corresponde à marcação manual da imagem percebida através do procedimento tradicional. O ponto pode ser marcado ou desmarcado a qualquer momento, através de comandos de voz.

O Item 3 é o núcleo do sistema, que é responsável por todas as funcionalidades da aplicação. Ele permite que os requisitos sejam atendidos de forma gradativa, ou seja, correções e melhorias podem ser realizadas a qualquer momento, enquanto estiver na etapa de desenvolvimento. Além disto, com as imagens percebida e real, a aplicação deve calcular o IPC e apresentá-lo ao profissional.

No Item 4, percebe-se a Interface de Apresentação dos Resultados, através da qual toda e qualquer informação do usuário será passada. Assim, a sobreposição das Imagens real e percebida e o IPC são exemplos de Informações que serão apresentadas através desta interface. Neste contexto, as informações permitirão ao profissional evidenciar, a olho nu, a distorção entre as imagens. Esta etapa é análoga à etapa de resultado do procedimento tradicional, no qual também é possível avaliar os resultados, geralmente através de impressos.

3.1.2 *AutoIMP*

No decorrer deste trabalho foi desenvolvida a aplicação, tal como se apresenta na Figura 8. Para possibilitar um melhor entendimento, a interface foi enumerada em regiões de 1 a 10, que serão explicadas a seguir.

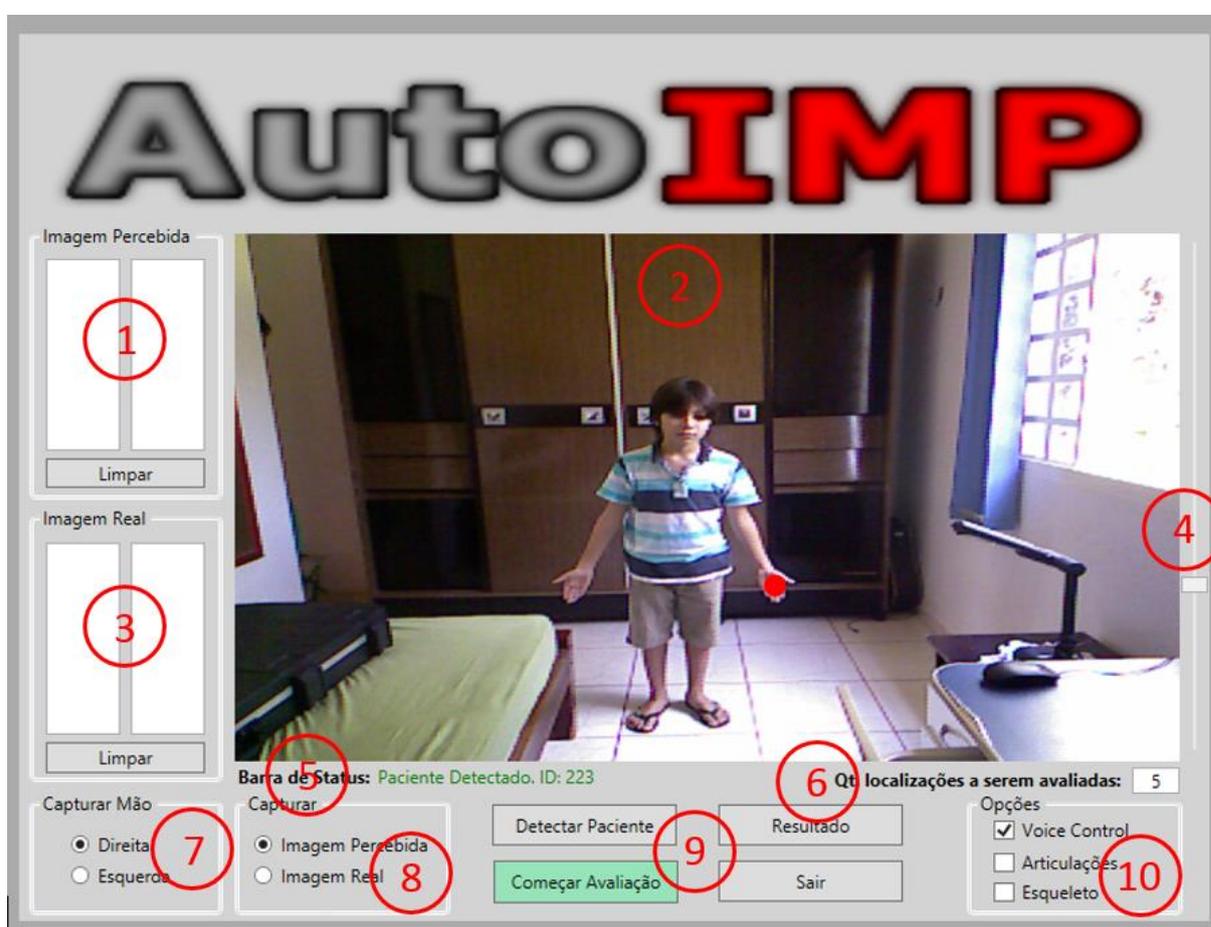


Figura 8 - Interface Inicial da Aplicação.

Na Figura 8, as regiões 1 e 3 são referentes às imagens Percebida e Real. Em ambos é possível notar duas colunas, representando às coordenadas X e Y, respectivamente; e o botão “Limpar”, que permite cancelar os pontos armazenados de cada Imagem.

A região 2 se refere ao fluxo de imagens *RGB*, que está sendo capturado pelo sensor RGB do Kinect e mostrado na referida área da aplicação. Nesta, é possível perceber o paciente com um círculo vermelho em sua mão direita. O ponto em questão é representado por coordenadas bidimensionais, sendo as coordenadas as localizações que serão armazenadas nos componentes de Imagem Percebida na região 1 e Imagem Real na região 3.

A região 4 contempla exclusivamente a rotação do eixo vertical do Kinect, que permite ajustar o ângulo para melhor visualização do paciente no campo visual do Kinect. Os ajustes de posicionamento do Kinect, tanto verticalmente quanto horizontalmente, devem ser feitos antes que se inicie a avaliação, pois as coordenadas sofrem alterações em relação ao posicionamento do Kinect.

Na região 5 é apresentada a Barra de Status, que apresenta informações úteis ao Avaliador. No exemplo mostrado na Figura 8, a informação apresentada se refere à identificação do esqueleto do paciente, que é fornecido pelo Kinect. O paciente pode ser definido através do acionamento da funcionalidade “Detectar Paciente”, que será apresentada a posteriori. Em suma, a identificação do esqueleto é importante por permitir que terceiros possam adentrar o campo visual do Kinect e não atrapalhar no rastreamento dos pontos. Assim, o avaliador pode se locomover dentro campo visual do Kinect, que não o confundirá com o paciente.

A região 6 se refere à quantidade de localizações ao longo do corpo do paciente que serão analisadas pelo Avaliador. Estas localizações são análogas às que foram apresentadas na Figura 1 e no Quadro 1, da seção 1.2 deste trabalho. Assim, no exemplo da Figura 8, foram definidas 5 localizações corporais para serem avaliadas. Isto fará com que a aplicação trabalhe sobre cinco localizações corporais.

A funcionalidade de Capturar Mão, mostrada na região 7, permite alterar entre qual mão do paciente será utilizada como referência para marcação dos pontos, em coordenadas X e Y. Cabe ao avaliador determinar qual mão do paciente será utilizada para composição dos pontos referentes à imagem percebida e real. Caso o avaliador acione esta funcionalidade, a aplicação alternará em tempo real o círculo vermelho na mão do paciente apresentado na Figura 8. Deve ser ressaltado ainda que, caso a opção *Voice Control* do ponto 10, que será apresentada adiante, esteja ativada, em qualquer momento o avaliador pode alterar entre a mão avaliada com o comando de voz “*Hand*”.

A captura dos pontos referentes à imagem real e percebida, mostradas na região 8, pode ser definida pelo avaliador a qualquer momento, selecionando com o mouse a captura que deseja, ou, caso a opção *Voice Control* esteja ativada, pode ser alternada através do comando de voz “*Change*”.

Os botões apresentados na região 9 são responsáveis por permitir ao Avaliador detectar ou começar a avaliação de um paciente. Neste sentido, o botão “Detectar Paciente” permite que a aplicação identifique o paciente, principalmente, no caso de haver mais de uma pessoa no campo visual do Kinect.

Para facilitar na identificação, foi definido na aplicação que o paciente é a pessoa que estiver com as duas mãos acima da cabeça, no momento em que o botão é acionado. Neste caso, o avaliador solicitará verbalmente ao paciente para que o mesmo posicione as duas mãos acima da cabeça. Na sequência, o avaliador acionará a funcionalidade e a aplicação determinará que o referido indivíduo é o paciente.

Detectado o paciente, será apresentado na Barra de Status da região 5 a informação referente à identificação do paciente. Em seguida, será projetado um círculo vermelho indicando qual mão do paciente será utilizada como referência para captura dos pontos.

Desta forma, pode o avaliador selecionar a qualquer momento quem é o paciente, sendo uma funcionalidade importante no caso de inconsistências do Kinect, já que este pode perder as referências do indivíduo que está definido como paciente. Deve ser ressaltado que esta funcionalidade também pode ser acionada verbalmente através do comando “*Detect*”, bem como o avaliador pode estar no campo visual do Kinect no momento em que desejar acionar a funcionalidade, já que a aplicação sabe que o paciente será o indivíduo que estiver com as duas mãos acima da cabeça.

O botão “Começar Avaliação”, destacado em verde, da região 9 da Figura 8, é acionado para que se comece a captura dos pontos. Ao clicar no botão, o mesmo ficará na cor vermelha com título “Parar Avaliação”, indicando que a funcionalidade está ativada e que a captura está em andamento. Tal funcionalidade é importante por permitir que o avaliador converse naturalmente com o paciente sem receio de que algum comando de voz seja interpretado pelo Kinect e algum ponto seja capturado indevidamente. O avaliador pode começar ou parar a avaliação através dos comandos de voz “*Start*” e “*Stop*”. Maiores detalhes sobre os comandos de voz estão na seção 3.3.2 deste trabalho.

Ainda na região 9, dentre os botões destacados, destaca-se o botão “Resultado”, que é responsável por apresentar uma nova janela no sistema contendo o resultado da avaliação ao avaliador. A funcionalidade deste botão será apresentada ainda nesta seção. O botão “Sair” é responsável por finalizar a aplicação a qualquer momento. Esta funcionalidade também pode ser acionada através do comando verbal “Exit”. Uma vez acionada, será perguntado ao usuário se o mesmo realmente deseja fechar a aplicação.

A Figura 8 apresenta ainda um grupo de funcionalidades que podem ser utilizadas pelo avaliador, apresentadas na região 10. As funcionalidades são: *Voice Control*, Articulações e Esqueleto. Tais funcionalidades podem ser ativadas a qualquer momento, sendo que seu impacto será expresso na imagem que está sendo capturada pelo sensor RGB e apresenta ao Avaliador e Paciente, na região 2.

A funcionalidade *Voice Control* é responsável por ativar os comandos de voz do Kinect, que pode ser acionada com o clique ou através do comando de voz “*Voice Control*”. Esta funcionalidade também permite que qualquer comando de voz indesejado não seja interpretado pelo Kinect, já que existe a necessidade do avaliador estar se comunicando com o paciente durante a avaliação.

A funcionalidade de Articulações é responsável por mapear as vinte Juntas reconhecidas pelo Kinect ao longo do corpo do paciente e mostra-las no campo de visualização de imagem na região 2. De maneira análoga, a funcionalidade de esqueleto faz a ligação entre os pontos e apresenta o esqueleto na região 2.

O supracitado botão “Resultado” apresentado na região 9 da Figura 8, como dito anteriormente, apresenta uma janela de resultado ao avaliador, tal como apresentado na Figura 9.

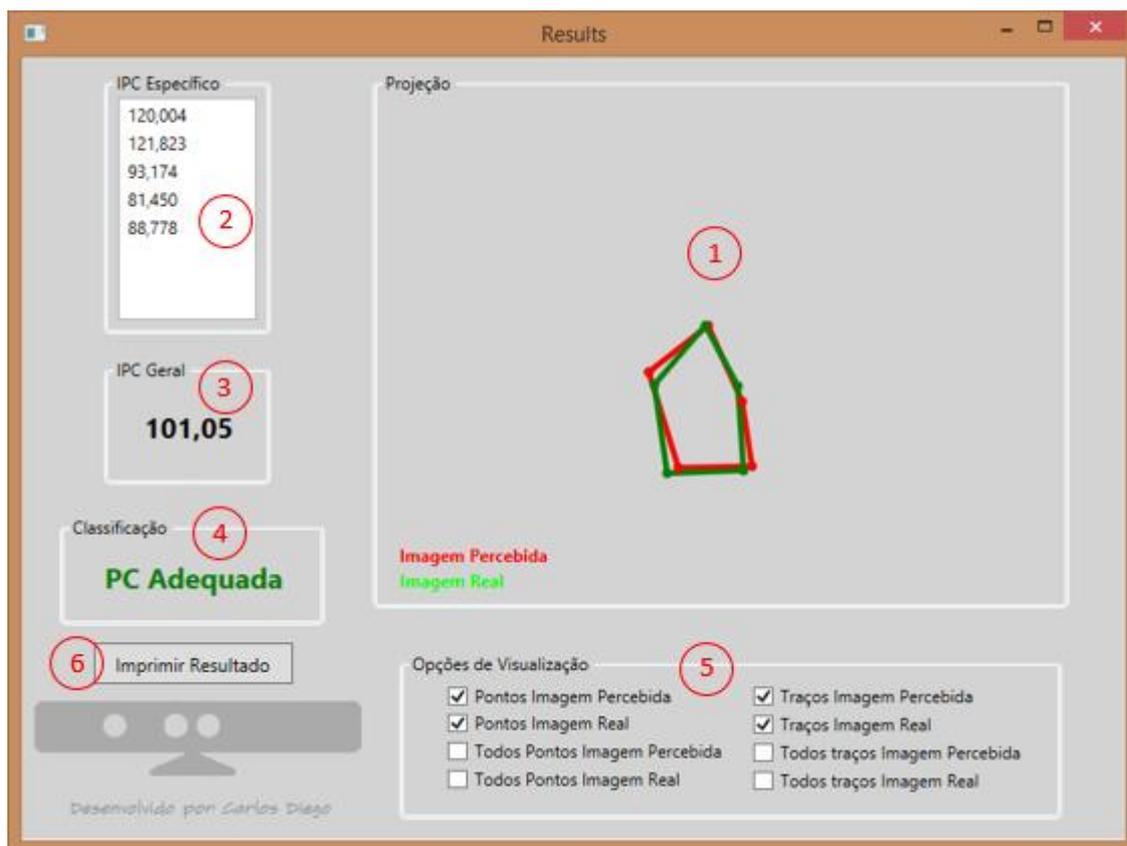


Figura 9 - Interface de Resultados da Avaliação

Na Figura 9 é possível perceber grupos de funcionalidades, que também foram enumerados para melhor entendimento, que serão descritas a seguir.

Na região 1, é apresentada a sobreposição dos desenhos da imagem percebida e imagem real. Em vermelho, apresenta-se o desenho da Imagem Percebida, e, em verde, o da Imagem Real. O desenho é apresentado conforme as opções de visualização que estão marcadas na região 5. A cada alteração nas opções de visualização, a imagem é atualizada instantaneamente.

Na região 2, a aplicação apresenta o Índice de Percepção Corporal para cada localização do corpo na ordem em que foram capturadas pelo avaliador, enquanto a região 3 apresenta a média geral do IPC envolvendo cada localização avaliada. Com isso, foi possível determinar o IPC Geral, conforme especificações definidas na seção 1.2.3 deste trabalho.

Definido o IPC, é possível classificar o indivíduo em hipoesquemático, normal ou hiperesquemático. O resultado desta classificação é apresentado na região 4.

Por fim, a região 5 apresenta as diversas opções de visualização dos desenhos referentes à imagem percebida e à imagem real. A aplicação permite que o Avaliador avalie as imagens referente a todos pontos capturados, uma vez que a Imagem Percebida ou Imagem Real é mostrada através da média dos pontos capturados. Dentre as opções disponíveis, destacam-se pontos da imagem percebida e pontos da imagem real, bem como os traços da imagem percebida e os traços da imagem real.

Ao ativar os “pontos da imagem percebida” ou “pontos da imagem real”, a aplicação desenhará um ponto em vermelho ou verde, na localização média da quantidade de pontos analisados para cada localização corporal. Isto é, o avaliador pode capturar quantos pontos achar necessário para cada localização do corpo humano, mas a aplicação desenhará a média da localização das mesmas. Da mesma forma, ao ativar “traços da imagem percebida” ou “traços da imagem real”, o sistema desenhará em vermelho ou verde o traço referente à média dos pontos da imagem percebida ou pontos da imagem real.

As demais funcionalidades funcionam da mesma forma, contudo serão apresentados todos os pontos capturados para cada segmento do corpo analisado. Isto é, não será calculada a média dos pontos. Isto permite ao avaliador visualizar possíveis discrepâncias referentes a erros de captura dos pontos.

Por fim, o ponto 6 trata da pré-disposição do sistema para impressão do resultado da avaliação. Contudo, esta funcionalidade ainda não está completamente implementada, sendo necessário a criação de métodos específicos para impressão, que não fazem parte do escopo deste trabalho.

As seções a seguir apresentarão como as funcionalidades foram desenvolvidas.

3.2 Utilizando o Kinect para Automatizar o IMP

O protocolo para avaliação do esquema corporal através do IMP estabelece que devem ser analisados pontos específicos, que podem variar de acordo com especificações do profissional avaliador. Tradicionalmente, têm-se utilizado os pontos apresentados no Quadro 1 da seção 1.2 e as respectivas inferências do indivíduo que está sendo submetido ao procedimento, como é do procedimento aplicado por Ferreira (2012).

Por padrão, o Kinect mapeia o corpo humano em vinte *Joints* (pontos) e fornece a localização tridimensional de cada ponto ao desenvolvedor. Percebe-se que os pontos fornecidos pelo Kinect são diferentes dos pontos tradicionalmente analisados através do IMP.

Por exemplo, o ápice da cabeça é um dos pontos utilizados no estudo de Pereira (2010), mas o Kinect fornece o centro da cabeça, conforme observado na Figura 10. Entretanto, foi definido, a princípio, que os pontos a serem utilizados para compor a IR estariam dentre os fornecidos nativamente pelo Kinect, apresentados na Figura 10, sendo eles: cabeça, ombros direito e esquerdo, quadril esquerdo e direito.

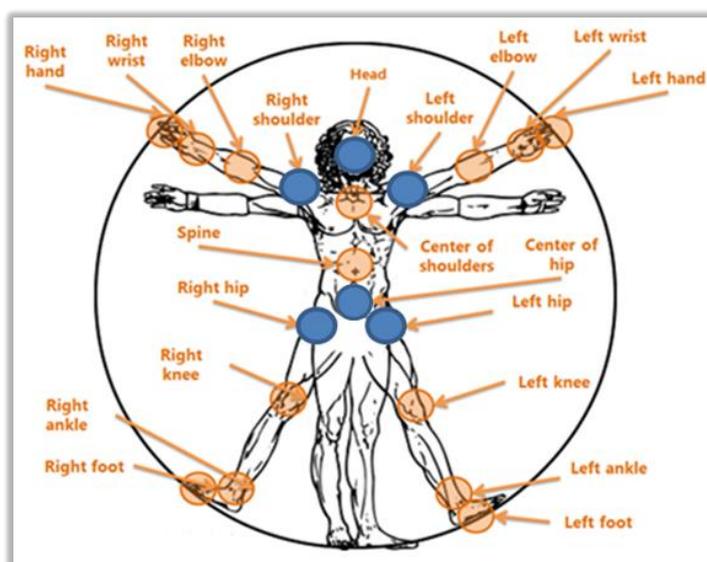


Figura 10 – As 20 Joints do esqueleto (MICROSOFT, 2014).

A Figura 10 apresenta os pontos que são disponibilizados pelo Kinect ao desenvolvedor, através da API, que poupa esforço de desenvolvimento por parte dos desenvolvedores. Como dito, os pontos destacados em azul foram selecionados para compor a IR do indivíduo, de forma análoga ao procedimento tradicional, onde o profissional avaliador determina os pontos reais a serem analisados que serão percebidos pelo paciente. Contudo, percebeu-se que, apesar da localização dos pontos reconhecidos automaticamente se aproximarem dos reconhecidos tradicionalmente, os pontos reconhecidos automaticamente variavam de acordo com os padrões antropométricos de cada paciente, bem como não conferiam com os pontos que deveriam ser analisados pelo profissional avaliador.

Tal motivo se explica pelo fato que o algoritmo para reconhecimento dos pontos, nativo do Kinect, trabalha de forma a representar visualmente o indivíduo de forma mais exata. Isto é, irregulares na postura, por exemplo, são tratadas pelo Kinect, para que as aplicações que sejam desenvolvidas reconheçam melhor o indivíduo.

Neste sentido, os pontos determinados na Figura 10 não forneciam a precisão necessária para que a aplicação pudesse fornecer dados suficientemente corretos, já que os

pontos se adaptavam em localizações diferentes do corpo, variando de acordo com cada indivíduo, que na maioria das vezes, não podiam ser tocadas pelo avaliador, por exemplo, centro da cabeça.

Neste contexto, foi ressaltado ainda que o fato dos pontos não serem os mesmos utilizados pelos profissionais na aplicação tradicional do procedimento, também comprometiam a eficácia da aplicação, já que limitava a capacidade do profissional avaliador no que tange a análise de outros pontos ao longo do corpo do paciente, caso fosse necessário.

A Figura 11 apresenta a disparidade entre os pontos utilizados no procedimento tradicional e os selecionados para comporem a automatização do procedimento.

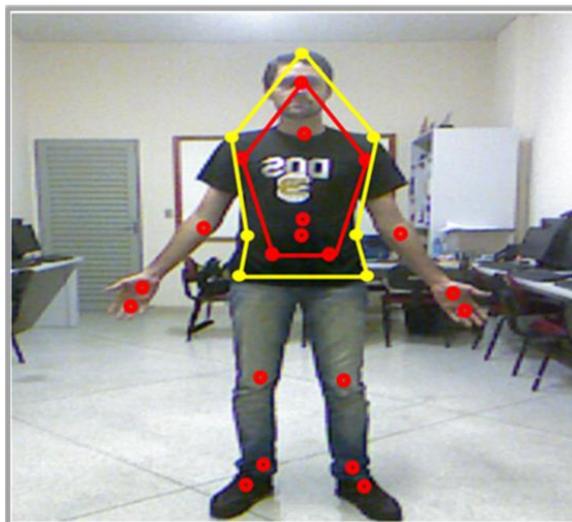


Figura 11 - Marcação dos pontos que compõem a Imagem Real através do procedimento tradicional (amarelo) e automatizado (vermelho).

Na Figura 11, é evidente a diferença entre a marcação tradicional e a automatizada, que pode chegar até seis centímetros de diferença, variando de acordo com o tamanho corporal de cada paciente e as condições de ambiente em que o Kinect está sendo utilizado. Percebe-se que a marcação automatizada fornecia apenas cinco pontos como referência, enquanto a marcação tradicional da IR se baseava em pelo menos sete pontos.

Diante dos resultados prévios obtidos, foi constatado que a aplicação estava inviável para uso, pois fornecia dados inconsistentes e limitava a quantidade de pontos que o profissional avaliador desejava utilizar.

Tal fato permitiu perceber que os parâmetros que estavam sendo utilizados não eram suficientes para fornecer os dados referentes à imagem real que o profissional fisioterapeuta precisava. Portanto, decidiu-se que seria utilizada outra abordagem para captura dos pontos para composição da Imagem Real.

A nova abordagem foi caracterizada por não utilizar parte dos pontos fornecidos nativamente pelo Kinect, utilizando apenas os pontos referentes às mãos do paciente e do profissional avaliador, que não participava do escopo da aplicação. Ou seja, anteriormente o avaliador apenas tocava o paciente no ponto que desejava ser avaliado, sendo que a aplicação não capturava nenhum dado referente à pessoa do avaliador, já que a posição real de um ponto era imposta pelo Kinect.

Por sua vez, o paciente, de olhos vendados, posicionava diante de si sua mão na localização onde achava que havia sido tocado, como se estivesse tocando em um espelho, procedimento semelhante à avaliação tradicional. Desta forma, a aplicação capturava a localização bidimensional de sua mão para compor a IP.

De acordo com a nova abordagem, o profissional avaliador também passa a ser um usuário participante da aplicação, isto é, no momento em que este toca o paciente, a aplicação passa a capturar também a posição de sua mão, que passa a corresponder à posição real de um ponto, compondo então a IR.

Diante do exposto, as seções a seguir explanarão acerca do funcionamento da aplicação com base na realidade apresentada.

3.3 Implementação

Como destacado na seção 2 deste trabalho, a aplicação foi desenvolvida no ambiente do Microsoft Visual Studio 2012, através da linguagem de programação C#, nativa da plataforma .NET, enquanto o kit de desenvolvimento do Kinect foi utilizado na versão 1.8, sendo a mais recente disponibilizada pela fabricante para esta versão do Kinect.

A princípio, pensou-se na aplicação detectando apenas o paciente, fornecendo dados para composição da imagem percebida, enquanto os dados dos pontos reais eram fornecidos pelos pontos nativos do Kinect. Contudo, esta primeira abordagem restringia o papel do avaliador e não fornecia dados suficientes para atender o procedimento.

Por este motivo, foi estabelecido que o avaliador passasse a fazer parte do campo visual do Kinect, afim de tocar os pontos que desejasse que o paciente representasse. Assim podia o avaliador determinar qualquer localidade ao longo do corpo do paciente para ser avaliado.

Desta forma, apesar do avaliador estar presente no campo visual do Kinect correspondente a avaliação, o Kinect foi programado para capturar dados apenas do paciente, que passou a fornecer dados percebidos e dados reais dos pontos para composição da imagem percebida e imagem real.

As subseções a seguir explanarão acerca da codificação da aplicação, começando a partir da inicialização do Kinect e declaração de variáveis de ambiente, até a explanação dos métodos mais complexos.

3.3.1 Inicialização do Sensor e declaração das variáveis necessárias

Após a instalação dos drivers e do SDK, foi possível começar o desenvolvimento da aplicação. O código apresentado na Figura 12 mostra as declarações iniciais das variáveis para o funcionamento da aplicação.

```
30 public partial class MainWindow : Window
31 {
32     public bool MaoEsquerdaAcimaCabeca;
33     public bool MaoDireitaAcimaDaCabeca;
34     public bool mostrarEsqueleto=false;
35     public bool mostrarArticulacoes=false;
36     public bool voiceControl = false;
37     public bool comecar_avaliacao = false;
38     public bool detectouPaciente = false;
39     public int idPaciente = -1;
40     public bool clicou_para_detectar_paciente=false;
41     bool capturar = false;
42
43     public SpeechRecognitionEngine speechEngine;
44     KinectSensor kinect;
45
46     List<SkeletonPoint> pontosImagemPercebida;
47     List<SkeletonPoint> pontosImagemReal;
```

Figura 12 - Declaração das variáveis iniciais.

Na Figura 12, é apresentado a declaração da classe parcial da aplicação, denominada MainWindow, que é classe principal da aplicação desenvolvida. Da linha 32 à 41 são declaradas variáveis de tipos primitivos, responsáveis por possibilitar que determinadas ações, que serão explicadas nas seções seguintes, sejam tomadas pela aplicação.

Na linha 43 é declarado o objeto speechEngine, do tipo SpeechRecognitionEngine, que é responsável por instanciar o motor de

reconhecimento de voz da Microsoft, que será utilizado para o reconhecimento dos comandos de voz pela Aplicação.

A linha 44 apresenta a instanciação do objeto `kinect`, do tipo `KinectSensor`, do próprio SDK de desenvolvimento, que possui todos os métodos e variáveis relacionadas ao Kinect. Qualquer ação que se deseje realizar na aplicação, no que tange a algum recurso do Kinect, deve ser tomada a partir deste objeto, tais como, acesso as câmeras e microfones, até regulação do eixo vertical.

Declaradas as variáveis e objetos chave, foi possível começar o desenvolvimento das funcionalidades do sistema, que se apresentam nas seções seguintes.

3.3.2 Comandos de Voz

O AutoIMP foi desenvolvido com base nas interfaces naturais, objetivando eliminar o máximo possível de controles remotos que possam existir entre a aplicação e o avaliador. Neste sentido, foi utilizado o recurso de captura de áudio do Kinect, citado na seção 1.3.4 deste trabalho.

Para que os comandos de voz pudessem ter sido implementados, primeiramente foi necessário adicionar o *Assembly speech* da Microsoft, que é um pacote de comandos de áudio já pré compilados, que facilita o desenvolvimento de aplicações de áudio. Com o Assembly devidamente incluso no Projeto, e a declaração da variável `speechEngine`, apresentada na seção 3.3.1, foi possível criar o método para reconhecimento de voz através do Kinect, conforme apresentado na Figura 13.

```
49 private static RecognizerInfo GetKinectRecognizer()
50 {
51     foreach (RecognizerInfo recognizer in SpeechRecognitionEngine.InstalledRecognizers())
52     {
53         string value;
54         recognizer.AdditionalInfo.TryGetValue("Kinect", out value);
55         if ("True".Equals(value, StringComparison.OrdinalIgnoreCase)
56             && "en-US".Equals(recognizer.Culture.Name, StringComparison.OrdinalIgnoreCase))
57         {
58             return recognizer;
59         }
60     }
61     return null;
62 }
63 }
```

Figura 13 - Declaração do Método de inicialização dos comandos de voz para o Kinect.

Na Figura 13, o método `GetKinectRecognizer` é responsável reconhecer os dispositivos de captura de áudio que são do tipo Kinect, conforme se observa no laço de repetição da linha 51. No caso da aplicação desenvolvida, foi utilizado apenas um sensor.

Portanto, logo após identificá-lo, o mesmo é retornado para quem chamar este método, tal como mostrado na linha 58.

Com o método `GetKinectRecognizer` criado, torna-se possível a criação da gramática que será utilizada na aplicação. A gramática contém todos comandos de voz que serão reconhecidos, bem como as respectivas frases que serão repassadas para aplicação. Deve ser ressaltado que a gramática é nativamente criada no idioma Inglês, visto que a Microsoft não oferece suporte para criação de comandos de voz no idioma português.

O método de inicialização da gramática se dá conforme apresentado na Figura 14.

```
64 public void inicializarGramatica()
65 {
66     RecognizerInfo ri = GetKinectRecognizer();
67
68     if (ri != null)
69     {
70         this.speechEngine = new SpeechRecognitionEngine(ri.Id);
71         var comandos_de_voz = new Choices();
72         //Comando capturado --> comando entendido que sera trabalhado
73         comandos_de_voz.Add(new SemanticResultValue("voice control", "kinect"));
74         comandos_de_voz.Add(new SemanticResultValue("detect", "detect"));
75         comandos_de_voz.Add(new SemanticResultValue("okay", "okay"));
76         comandos_de_voz.Add(new SemanticResultValue("ok", "okay"));
77         comandos_de_voz.Add(new SemanticResultValue("now", "okay"));
78         comandos_de_voz.Add(new SemanticResultValue("cancel", "cancel"));
79         comandos_de_voz.Add(new SemanticResultValue("start", "start"));
80         comandos_de_voz.Add(new SemanticResultValue("stop", "stop"));
81         comandos_de_voz.Add(new SemanticResultValue("change", "change"));
82         comandos_de_voz.Add(new SemanticResultValue("exit", "exit"));
83         comandos_de_voz.Add(new SemanticResultValue("yes", "yes"));
84         comandos_de_voz.Add(new SemanticResultValue("print", "print"));
85         comandos_de_voz.Add(new SemanticResultValue("hand", "hand"));
86         var gb = new GrammarBuilder { Culture = ri.Culture };
87         gb.Append(comandos_de_voz);
88         var g = new Grammar(gb);
89         speechEngine.LoadGrammar(g);
90         speechEngine.SpeechRecognized += SpeechRecognized;
91         speechEngine.SetInputToAudioStream(kinect.AudioSource.Start(), new
92             SpeechAudioFormatInfo(EncodingFormat.Pcm, 16000, 16, 1, 32000, 2, null));
93         speechEngine.RecognizeAsync(RecognizeMode.Multiple);
94     }
95 }
```

Figura 14 - Método responsável por inicializar a gramática da aplicação.

Na linha 66 da Figura 14 é possível notar o objeto `ri`, do tipo `RecognizerInfo`, que recebe como valor o resultado do método `GetKinectReconizer()`, apresentado anteriormente, que contém a identificação de qual sensor Kinect será utilizado como fonte de áudio para captura dos comandos de voz.

Com a identificação do sensor que está conectado no computador, o método retorna a instanciação do mesmo, que será adicionado para o objeto `speechEngine`, conforme a linha 70. Deve ser ressaltado que este objeto é considerado chave, por ser crucial para que o `AutoIMP` detecte comandos de voz, conforme abordado na seção 3.3.1 deste trabalho.

Na linha 71, a variável `comandos_de_voz` é do tipo `Choices()`, e é responsável por armazenar todos possíveis comandos de voz que são interpretados pela aplicação. Das linhas 73 a 85 são declarados e inseridos na variável `comandos_de_voz` todos os comandos de voz utilizados pela aplicação. A Tabela 1 apresenta maiores detalhes sobre os comandos de voz e suas ações no sistema.

Tabela 1 - Comandos de Voz

Comando	Ação na Aplicação
“Voice Control”	Comando responsável por ativar ou desativar o reconhecimento dos comandos de voz por parte da aplicação.
“Detect”	Comando responsável por detectar o usuário que estiver com as duas mãos acima da cabeça para ser o paciente. Maiores detalhes estão escritos na seção 3.3.4 deste trabalho.
“Start”	Comando responsável por dar início à captura dos pontos percebidos ou reais.
“Okay”, “Ok” ou “Now”	Comando que realiza a captura de um ponto no momento em que o comando é verbalizado.
“Cancel”	Comando cancela o último ponto capturado, da imagem percebida ou imagem real.
“Stop”	Comando que finaliza o procedimento de captura dos pontos percebidos ou reais.
“Change”	Comando que altera entre captura dos pontos para compor Imagem Percebida ou Imagem Real
“Exit”	Comando que finaliza a aplicação.
“Yes”	Comando análogo ao comando <i>Enter</i> .
“Print”	Comando responsável por tirar uma <i>Screenshot</i> da tela.
“Hand”	Comando responsável por alterar entre a mão esquerda ou direita, para captura dos pontos da imagem percebida ou real.

Declarados os comandos de voz, o método `SpeechRecognized`, apresentado na Figura 15, é responsável por permitir a detecção dos comandos em tempo real.

```

120 //LOOP INFINITO PARA RECONHECIMENTO DE VOZ
121 private void SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
122 {
123     const double indiceDeCerteza = 0.3;
124     if (e.Result.Confidence >= indiceDeCerteza)
125     {
126         switch (e.Result.Semantics.Value.ToString())
127         {
128             case "kinect":
129                 this.VerificarVoiceControl();
130                 break;
131             case "okay":
132                 if (voiceControl)
133                     if (comecar_avaliacao == true)
134                         capturar = true;
135                 break;

```

Figura 15 - Método para reconhecimento em tempo real dos comandos de voz.

Na linha 123 da Figura 15, a variável `indiceDeCerteza`, a qual está atribuída o valor 0.3, é considerada determinante para o reconhecimento dos comandos de voz, pois os reconhecimentos de voz são baseados em aproximação. Assim, é possível que a aplicação compreenda um comando mesmo que este não seja pronunciado totalmente correto. A estrutura de controle `If`, da linha 124, é responsável por permitir que todas expressões que tiverem o índice de certeza maior que 0.3 sejam toleradas pela aplicação. Isto ajuda a evitar que comandos não desejados sejam interpretados como um comando reconhecido, bem como facilita na identificação dos comandos verbalizados pelo avaliador à aplicação.

Para cada comando de voz interpretado, haverá uma decisão a ser tomada pela aplicação. Logo, para cada comando de voz da Tabela 1, haverá sua respectiva ação na aplicação. Neste sentido, a partir da linha 128, são especificadas as ações a serem realizadas de acordo com cada comando identificado.

Por exemplo, ainda na linha 128, caso seja identificado o comando “Kinect”, a aplicação alternará entre a ativação ou desativação do *VoiceControl*, funcionalidade responsável por ativar ou desativar os comandos de voz do sistema.

Com o funcionamento dos comandos de áudio, o posicionamento da mão do paciente é incipiente na aplicação, pois fornece os dados referentes as localizações que são analisadas. Portanto, a seção a seguir apresenta os conceitos e implementações que foram utilizadas para realizar a detecção da mão do paciente.

3.3.3 Detecção da Mão

A detecção das mãos do paciente é fundamental no que tange à precisão dos resultados fornecidos pela aplicação, já que as mesmas definem a IP e a IR. Assim, o posicionamento do paciente em relação ao Kinect implica significativamente na precisão dos resultados. Por isso, é imprescindível que durante a realização do teste, o paciente não se locomova, bem como é recomendável pelo menos três capturas de pontos para cada localização corporal que o avaliador deseja avaliar no paciente.

As articulações também são conhecidas por *Joints* ou pontos ao longo do corpo. Assim, para detectar a posição de alguma articulação ao longo do corpo, foi desenvolvido o método `capturarPosicaoArticulacao`, conforme mostrado na Figura 16.

```
307 private SkeletonPoint capturarPosicaoArticulacao(int idSkeleto, JointType junta, SkeletonFrame frameAtual)
308 {
309     SkeletonPoint posicao = new SkeletonPoint();
310     if (frameAtual == null) return posicao;
311     using (frameAtual)
312     {
313         Skeleton[] esqueletos = new Skeleton[frameAtual.SkeletonArrayLength];
314         frameAtual.CopySkeletonDataTo(esqueletos);
315         IEnumerable<Skeleton> esqueletosRastreados =
316             esqueletos.Where(esqueleto => esqueleto.TrackingState ==
317                 SkeletonTrackingState.Tracked && esqueleto.TrackingId == idSkeleto);
318         Skeleton usuario = null;
319         if (esqueletosRastreados.Count() > 0)
320             usuario = esqueletosRastreados.First();
321         if (usuario != null)
322         {
323             return usuario.Joints[junta].Position;
324         }
325     }
326     return posicao;

```

Figura 16 - Método responsável por capturar a posição de alguma articulação.

Na Figura 16, o supracitado método recebe três valores como parâmetro, sendo o primeiro o `idSkeleto` do tipo `int`, responsável por guardar a identificação do esqueleto em que se deseja capturar a posição da articulação, podendo ser, por exemplo, a identificação do paciente ou avaliador. O segundo parâmetro é do tipo `JointType`, que é próprio do SDK do Kinect, responsável por determinar qual junta se deseja capturar ao longo do corpo. Por fim, o terceiro parâmetro recebe um objeto do tipo `SkeletonFrame`, que é um quadro de imagem onde será detectado a junta.

Determinado qual esqueleto se deseja capturar a posição de alguma junta, bem como qual junta se deseja capturar, a linha 315 é responsável por identificar o esqueleto se, e somente se, existir algum com o `id` informado. Caso não exista, o método retornará um `SkeletonPoint` vazio para quem o chamou.

Existindo o esqueleto em questão, a linha 323 retornará a posição tridimensional do ponto solicitado para o método ou variável que o chamou.

Na Figura 17, o método `DesenharArticulacao` é responsável por desenhar o círculo na mão do indivíduo apresentado na Figura 8.

```
22 public void DesenharArticulacao(Joint articulacao, Canvas canvasParaDesenhar)
23 {
24     int diametroArticulacao = articulacao.JointType == JointType.HandRight ? 50 : 10;
25     int larguraDesenho = 4;
26     Brush corDesenho = articulacao.JointType == JointType.HandRight ? Brushes.Red : Brushes.Yellow;
27     Ellipse objetoArticulacao = CriarComponenteVisualArticulacao(diametroArticulacao, larguraDesenho, corDesenho);
28     ColorImagePoint posicaoArticulacao = ConverterCoordenadasArticulacao(articulacao, canvasParaDesenhar.ActualWidth,
29                                                                     canvasParaDesenhar.ActualHeight);
30     double deslocamentoHorizontal = posicaoArticulacao.X - objetoArticulacao.Width / 2;
31     double deslocamentoVertical = (posicaoArticulacao.Y - objetoArticulacao.Height / 2);
32
33     if (deslocamentoVertical >= 0 && deslocamentoVertical < canvasParaDesenhar.ActualHeight
34         && deslocamentoHorizontal >= 0 && deslocamentoHorizontal < canvasParaDesenhar.ActualWidth)
35     {
36         Canvas.SetLeft(objetoArticulacao, deslocamentoHorizontal);
37         Canvas.SetTop(objetoArticulacao, deslocamentoVertical);
38         Canvas.SetZIndex(objetoArticulacao, 100);
39
40         canvasParaDesenhar.Children.Add(objetoArticulacao);
41     }
42 }
```

Figura 17 - Trecho de código responsável por desenhar os pontos e o círculo na mão dos indivíduos.

Como dito, o método `DesenharArticulacao` é responsável por o círculo vermelho em volta da mão do indivíduo apresentado na Figura 8. Na linha 24, é apresentada uma expressão com operador ternário, que objetiva definir o tamanho do círculo em cada articulação, variando apenas no caso da articulação ser a mão direita.

Em seguida, na linha 26 o operador ternário objetiva determinar a cor do círculo para cada articulação, variando apenas no caso da mão em questão ser a direita, onde a cor será vermelha, enquanto nos demais casos, amarela.

O método `PosicaoDaMao` no trecho de código da Figura 18 é responsável por capturar a posição das mãos do paciente e do avaliador e retornar ao método que o chamou, um vetor contendo duas posições de `SkeletonPoint`.

```

14 public static SkeletonPoint[] PosicaoDaMao (this SkeletonFrame quadro, KinectSensor kinectSensor, Canvas canvasParaDesenhar)
15 {
16     //Verificando para ver se o Kinect não está bugado.
17     if (kinectSensor == null) throw new ArgumentNullException("Erro não tratado com a instanciação do KinectSensor");
18     if (canvasParaDesenhar == null) throw new ArgumentNullException("Erro no canvasParaDesenhar");
19     //Declarando o vetor que retornará a posição da mão dos 2 esqueletos.
20     SkeletonPoint[] posicaoDaMao = new SkeletonPoint[2];
21     //Para controle do numero de esqueletos.
22     int contador = 0;
23     //Identificando os esqueletos para renderizá-los na tela.
24     Skeleton e = null;
25     IEnumerable<Skeleton> esqueletos = ObterEsqueletoUsuario(quadro);
26     if (esqueletos.Count() > 0)
27         e = esqueletos.First();
28     foreach (Skeleton esqueleto in esqueletos)
29     {
30         if (esqueleto != null)
31         {
32             EsqueletoUsuarioAuxiliar esqueletoUsuarioAuxiliar = new EsqueletoUsuarioAuxiliar(kinectSensor);
33             posicaoDaMao[contador++] = esqueleto.Joints[JointType.HandRight].Position;
34         }
35     }
36     //Retorna a posição da mão de A e B.
37     return posicaoDaMao;
38 }

```

Figura 18 - Método que captura a posição da mão direita do Avaliador e do Paciente.

Na Figura 18, o método `PosicaoDaMao` retorna um vetor de `SkeletonPoint`, que é uma classe disponibilizada pela API de desenvolvimento do Kinect que contem a posição X, Y e Z de um determinado ponto ao longo do corpo humano. Neste sentido, o objetivo deste método é reconhecer dois indivíduos (A e B) e retornar a posição da mão direita dos mesmos.

Para que isto fosse possível, o código descrito na linha 20 instancia um vetor de `SkeletonPoint` de tamanho 2, para posteriormente preenchê-los com as Juntas referentes à mão de A e B, que são identificadas entre as linhas 25 e 35, onde o código se responsabiliza por verificar se tem e quantos indivíduos estão presentes no campo visual do Kinect.

Por fim, a linha 37 retorna o vetor de duas posições do tipo `SkeletonPoint` contendo a posição das mãos do profissional avaliador e do paciente, que torna possível calcular a distância entre ambas, permitindo identificar a diferença em termos de distância entre as posições referentes às imagens real e percebida, que serão trabalhados na seção a seguir.

3.3.4 Detecção do Paciente

O paciente é detectado através do comando verbal “*Detect*”. Ao ser pronunciado, a aplicação procurará no campo visual do sensor de infra vermelho a existência de algum indivíduo que esteja com as duas mãos acima da cabeça. Caso seja detectado algum indivíduo, o mesmo será definido para ser o paciente.

Desta forma, torna-se difícil que a aplicação confunda entre o paciente e o avaliador, ambos presentes no campo visual do Kinect. A Figura está definindo que o indivíduo no

campo visual do Kinect que estiverem com as duas mãos acima da cabeça será o paciente. A Figura 18 apresenta como se dá a detecção do Paciente.

```
216 | if (clicou_para_detectar_paciente)
217 | {
218 |     //A pessoa que estiver com as duas mãos acima da cabeça.
219 |     idPaciente = detectarPaciente(e.OpenSkeletonFrame());
220 |     if (idPaciente != -1)
221 |     {
222 |         detectouPaciente = true;
223 |         botao_startAvaliacao.IsEnabled = true;
224 |     }
225 |     else
226 |     {
227 |         detectouPaciente = false;
228 |     }
229 |     clicou_para_detectar_paciente = false;
230 | }
```

Figura 19 - Método que lança ação para aplicação detectar paciente.

Na Figura 19, a variável `clicou_para_detectar_paciente`, que foi apresentada na seção 3.3.1, é do tipo booleana. Sempre que for identificado o comando de voz “Detect”, esta variável receberá o valor *True*, que satisfará a condição da estrutura de controle *If*, na linha 216.

Deste modo, na linha 219, a aplicação chamará o método `detectarPaciente`, passando como parâmetro um quadro de imagens que está sendo capturado em tempo real. A Figura 20 apresenta maiores detalhes sobre o método.

```
265 | private int detectarPaciente(SkeletonFrame quadro)
266 | {
267 |     if (quadro == null) return -1;
268 |     using (quadro)
269 |     {
270 |         return quadro.DetectarIdPaciente();
271 |     }
272 | }
```

Figura 20 - Método para detectar paciente.

Na linha 270, percebe-se o retorno deste método, que é o ID do usuário detectado com as duas mãos acima da cabeça. Caso não seja detectado nenhum usuário, a aplicação continuará sem informação do Paciente, de forma que o Avaliador terá que realizar o procedimento de detecção do paciente novamente. A Figura 21 apresenta como o paciente, com as duas mãos acima da cabeça, é detectado.

```

105 public static int DetectarIdPaciente(this SkeletonFrame quadro)
106 {
107     //Verificando para ver se o Kinect não está bugado.
108     if (quadro == null)
109         throw new ArgumentNullException("Erro não identificado. Tente novamente.");
110     IEnumerable<Skeleton> esqueletos = ObterEsqueletoUsuario(quadro);
111     if (esqueletos.Count() > 0)
112     {
113         foreach (Skeleton esqueleto in esqueletos)
114         {
115             if (esqueleto != null && esqueleto.TrackingState==SkeletonTrackingState.Tracked)
116                 if (esqueleto.Joints[JointType.HandRight].Position.Y > esqueleto.Joints[JointType.Head].Position.Y)
117                     if (esqueleto.Joints[JointType.HandLeft].Position.Y > esqueleto.Joints[JointType.Head].Position.Y)
118                         return esqueleto.TrackingId;
119         }
120     }
121     return -1;
122 }

```

Figura 21 - Método responsável por detectar o ID do usuário que estiver com as duas mãos acima da cabeça.

Na Figura 21, o primeiro passo a ser realizado para detecção do Paciente é a verificação do quadro de imagem que foi passado, objetivando evitar inconsistências na aplicação. Caso o quadro seja nulo, a aplicação lançará uma exceção, evitando que a mesma pare de funcionar inesperadamente, tal como apresentado nas linhas 108 e 109.

Posteriormente, na linha 110, o método `ObterEsqueletoUsuario` é responsável por detectar todos os esqueletos que estão no frame e enumerá-los na variável `esqueleto`. Maiores detalhes sobre este método serão apresentados após a explicação ainda nesta seção.

Após detectar todos os esqueletos que estão em um quadro de imagem, as linhas de 113 a 120 são responsáveis por analisar, caso existam indivíduos detectados, conforme a linha 111, se algum indivíduo está com as duas mãos acima da cabeça.

Deve ser ressaltado que a aplicação determinará como paciente se, e somente se, algum usuário estiver, de fato, com as mãos acima da cabeça no campo visual do Kinect, pois é possível que, se um indivíduo estiver fora do campo visual, a aplicação tente adivinhar o posicionamento das juntas deste indivíduo.

Desta forma, percebeu-se que tal funcionalidade poderia ser boa ou ruim, e foi desativada, pois em alguns casos detectaria o indivíduo quando não fosse necessário, ou poderia determinar um indivíduo que não estivesse com as duas mãos acima da cabeça.

Caso seja encontrado algum indivíduo que esteja com as duas mãos acima da cabeça, o método retornará o ID do indivíduo, para compor a identificação do paciente. Caso contrário, o método retornará o valor -1, indicando que o paciente não foi detectado.

A Figura 22 apresenta maiores detalhes sobre o método `ObterEsqueletoUsuario`, que foi utilizado no método `DetectarIdPaciente`, da Figura 20.

```
92 public static IEnumerable<Skeleton> ObterEsqueletoUsuario(this SkeletonFrame quadro)
93 {
94     Skeleton[] esqueletos = new Skeleton[quadro.SkeletonArrayLength];
95     quadro.CopySkeletonDataTo(esqueletos);
96     IEnumerable<Skeleton> esqueletosRastreados = esqueletos.Where(esqueleto =>
97         esqueleto.TrackingState == SkeletonTrackingState.Tracked);
98     return esqueletosRastreados;
99 }
```

Figura 22 - Método para obter esqueletos contidos em um quadro de imagem.

A Figura 22 apresenta o método responsável por detectar em um quadro de imagem, a existência de algum esqueleto. Na linha 94 é declarado um vetor de esqueletos, do tamanho da quantidade de esqueletos que foi detectada no quadro de imagem.

Após a declaração do vetor de esqueletos, o método `CopySkeletonDataTo` do objeto `quadro` é responsável por copiar os dados referentes a posicionamento, ID e status de rastreamento de cada esqueleto para o vetor de esqueletos que fora criado.

Na linha 96 foi declarada a variável `esqueletosRastreados`, que receberá exclusivamente os esqueletos contidos na variável `esqueletos` que possuem status rastreado, visto que podem existir esqueletos que não estão totalmente rastreados, os quais tem suas posições de articulações preditas pela própria API do Kinect. Por fim, na linha 98, o método retorna a quem o chamou, o vetor de esqueletos rastreados.

Detectado o paciente e o posicionamento de suas mãos, foi possível realizar a implementação da captura dos pontos para composição da imagem percebida e imagem real. A seção 3.3.5 apresentará detalhes acerca da implementação dos referidos métodos.

3.3.5 Captura dos Pontos da Imagem Percebida e Real

O comando para captura de um ponto é verbal, sendo neste caso o comando “Ok”. Quando pronunciado, este comando captura a posição da mão do paciente. Portanto, é necessário que o paciente esteja detectado, caso contrário será lançada uma exceção. A Figura 22 apresenta o trecho de código responsável pela captura dos pontos da imagem percebida ou real.

```

236 if (capturar==true)
237 {
238     SkeletonPoint position;
239     if(botao_radio_mao_direita.IsChecked==true)
240         position = capturarPosicaoArticulacao(idPaciente, JointType.HandRight, e.OpenSkeletonFrame());
241     else
242         position = capturarPosicaoArticulacao(idPaciente, JointType.HandLeft, e.OpenSkeletonFrame());
243     if (rb_percebida.IsChecked == true)
244     {
245         if (pontosImagemPercebida == null)
246             pontosImagemPercebida = new List<SkeletonPoint>();
247         pontosImagemPercebida.Add(position);
248         listBox_percebido_x.Items.Add(string.Format("{0:n3}", position.X));
249         listBox_percebido_y.Items.Add(string.Format("{0:n3}", position.Y));
250     }
251     else
252     {
253         if (pontosImagemReal == null)
254             pontosImagemReal = new List<SkeletonPoint>();
255         pontosImagemReal.Add(position);
256         listBox_real_x.Items.Add(string.Format("{0:n3}", position.X));
257         listBox_real_y.Items.Add(string.Format("{0:n3}", position.Y));
258     }
259     capturar = false;
260 }

```

Figura 23 - Método para capturar os pontos

Na Figura 23, para que se satisfaça a condição do método, é necessário que a variável `capturar` da linha 236 seja verdadeira. Para que esta variável esteja verdadeira, faz-se necessário que o avaliador verbalize o comando “Ok”. Imediatamente a aplicação irá procurar saber qual mão está sendo utilizada como referência para marcação dos pontos, para assim pegar sua posição, tal como mostrado das linhas 238 a 242.

Posteriormente, é necessário verificar se estão sendo capturados os pontos para imagem percebida ou imagem real, como apresentado das linhas 243 a 258. Neste intervalo de código, também é realizada a inserção de cada ponto capturado em sua respectiva coluna de valores da imagem real ou percebida, tal como apresentado no ponto 1 e 3 da Figura 8.

Por fim, a linha 259 é responsável por atribuir o valor `false` à variável `capturar`, indicando que o ponto já foi capturado, evitando que a aplicação entre em um loop infinito de captura de pontos.

3.3.6 Cálculo da Média dos Pontos Capturados

É evidente que a precisão dos dados capturados está diretamente relacionada com a quantidade de referências capturadas para cada localização do corpo humano. Em sua pesquisa, Salzano (2012) afirma que são capturadas três referências para cada localização corporal para composição da imagem percebida, sendo que a média das três localizações fornecem uma maior precisão da real localização percebida pelo paciente.

Neste sentido, a aplicação foi desenvolvida para permitir que o avaliador capture quantas referências desejar para cada localização corporal. Contudo, a aplicação restringe a um número igual de pontos para cada localização que se deseje analisar.

Como a captura dos pontos referentes a imagem real também são aferidos a partir da mão do paciente, embora de olhos abertos e visualizando sua imagem à tela, é viável que se capture mais de três localizações para cada localização corporal que se deseje analisar.

Visto que a média dos pontos capturados fornecem uma localização mais precisa no que tange a localização que está sendo analisada, foi necessário criar um método para o cálculo da média de tais pontos. Para que isso fosse possível, também foi necessário saber exatamente quantas localizações foram avaliadas ao longo do corpo do paciente. A Figura 24 apresenta o método `calcularMediaPontos`.

```
124 public List<SkeletonPoint> calcularMediaPontos(List<SkeletonPoint> pontos)
125 {
126     List<SkeletonPoint> lista = new List<SkeletonPoint>();
127     float soma_x = 0;
128     float soma_y = 0;
129     float soma_z = 0;
130     int contador = 0;
131     int pontos_por_ponto = pontos.Count() / this.qtPontos;
132     foreach (SkeletonPoint ponto in pontos)
133     {
134         soma_x += ponto.X;
135         soma_y += ponto.Y;
136         soma_z += ponto.Z;
137         contador++;
138         if (contador == pontos_por_ponto)
139         {
140             SkeletonPoint novo = new SkeletonPoint();
141             novo.X = soma_x / pontos_por_ponto;
142             novo.Y = soma_y / pontos_por_ponto;
143             novo.Z = soma_z / pontos_por_ponto;
144             lista.Add(novo);
145             soma_x = soma_y = soma_z = contador = 0;
146         }
147     }
148     return lista;
```

Figura 24 - Método responsável por realizar o cálculo da média dos pontos.

Na Figura 24, pode ser observado que o supracitado método recebe como parâmetro uma lista de `SkeletonPoint`. Não obstante, os pontos desta lista são referentes aos pontos da imagem percebida ou imagem real.

Como a notação cartesiana utilizada pelo Kinect é tridimensional, das linhas 127 a 129 foram declaradas três variáveis do tipo `float` para armazenamento da soma de cada dimensão da quantidade total de pontos capturados para imagem percebida ou imagem real.

Na linha 131 da Figura 24, a variável `pontos_por_ponto` recebe o valor total de localizações dividido pela quantidade de pontos analisados. Neste caso, ela armazena quantos pontos foram capturados para cada localização do corpo.

Em seguida, das linhas 137 a 145, é realizado a soma das quantidades de pontos para cada dimensão. Posteriormente, o valor da soma é dividido pela variável `pontos_por_ponto`, gerando então a média dos pontos capturados.

Imediatamente, a média dos pontos capturados de uma determinada localização é inserida em uma nova lista de pontos. Este processo se repete enquanto houverem localizações para terem a média calculada. Por fim, o método retorna a nova lista contendo apenas os valores da média de cada localização corporal analisada, conforme a linha 148.

Em suma, a captura dos pontos é um fator determinante para o funcionamento da aplicação. Sua precisão é um fator preponderante para a análise sistematizada dos resultados, bem como cálculo do IPC e a classificação geral do paciente. Assim, a seção 3.3.7 apresentará o passo a passo para criação dos métodos para cálculo do IPC e classificação do paciente.

3.3.7 Cálculo do IPC e classificação do Paciente

Para que o cálculo do IPC pudesse ser realizado, foi necessário realizar o cálculo da média dos pontos analisados da imagem percebida e imagem real, através do método explicado na seção 3.3.6 deste trabalho.

Como explicado na seção 1.2.3, a fórmula para cálculo do IPC é representada pela média das três marcações do tamanho percebido pelo indivíduo dividido pelo tamanho real multiplicado por 100. Contudo, não precisam ser exatamente três aferências capturadas, tampouco, média calculadas apenas para o tamanho percebido.

Neste sentido, podem ser capturadas quantas aferências o avaliador julgar necessário, tanto para imagem real quanto percebida, desde que respeite a mesma quantidade de capturas para cada localização corporal.

Assim, a aplicação realiza automaticamente o cálculo da média da posição percebida e real. Em seguida, divide o valor obtido referente a imagem percebida pelo valor obtido

referente a imagem real e multiplica por 100. A Figura 25 apresenta a implementação do método para cálculo do IPC.

```
151 public void calcularIPC(List<SkeletonPoint> pontosPercebidos, List<SkeletonPoint> pontosReais)
152 {
153     //Como a média já está calculada, resta-nos realizar os cálculos.
154     int count = pontosPercebidos.Count();
155     float total = 0;
156     for (int i = 0; i < count; i++)
157     {
158         float valor = (((pontosPercebidos[i].X / pontosReais[i].X * 100) +
159             (pontosPercebidos[i].Y / pontosReais[i].Y * 100)) / 2);
160         lb_ipc.Items.Add(string.Format("{0:n3}", valor));
161         total += valor;
162     }
163     label_ipc_geral.Text = string.Format("{0:n2}", total/count);
164     if (total / count < 99.4)
165     {
166         label_classificacao_ipc.Text = "Hipoesquemático";
167         label_classificacao_ipc.Foreground = Brushes.Red;
168     }
169     else
170     {
171         if (total / count < 112.3)
172         {
173             label_classificacao_ipc.Text = "PC Adequada";
174             label_classificacao_ipc.Foreground = Brushes.Green;
175         }
176         else
177         {
178             if (total / count > 112.3)
179             {
180                 label_classificacao_ipc.Text = "Hiperesquemático";
181                 label_classificacao_ipc.Foreground = Brushes.Red;
182             }
183         }
184     }
185 }
```

Figura 25 - Método que realiza o cálculo do IPC

Na Figura 25, será calculado o IPC para cada segmento corporal analisado. As linhas 156 a 162 são responsáveis por realizar o cálculo da fórmula do IPC e adicionar o resultado obtido na coluna de resultados do IPC, no campo 2 da Figura 9. Este procedimento se repete para todos os pontos ao longo do corpo do paciente que foram capturados durante o teste.

Posteriormente, entre as linhas 164 e 183, é realizada a classificação do indivíduo, sugerida por Segheto *et al.* (2010), onde os sujeitos que se perceberam abaixo de 99,4% foram classificados Hipoesquemático, tal como mostrado na linha 166. Se a média geral do IPC ficar entre 99,4% e 112,3%, o indivíduo será classificado com a PC adequada, como mostrado na linha 173. Por fim, se a média geral do IPC for superior a 112,3%, o será classificado com hipoesquematia, como consta na linha 180.

Por fim, a classificação é atualizada e mostrada na janela de resultado, no ponto 4 da Figura 9.

3.3.8 Projeção dos pontos na tela

Considerando que no teste tradicional os pontos são projetados e ligados, para obter o desenho da percepção corporal, foi necessário que a aplicação também realizasse a projeção dos pontos e apresentasse o resultado ao avaliador.

Embora o Kinect forneça posições tridimensionais, foram consideradas apenas as posições X e Y, para posterior renderização em um objeto do tipo Canvas, nativo nas aplicações WPF do Visual Studio, utilizado quando se deseja trabalhar com imagens. Neste sentido, a Figura 26 apresenta o método `Desenhar()`, responsável pelo desenho no objeto Canvas.

```

40 public void Desenhar()
41 {
42     if (startou)
43     {
44         canvas.Children.Clear();
45         if (cb_pontos_imagem_percebida.IsChecked == true)
46             desenharPontos(calcularMediaPontos(pontosPercebidos), Brushes.Red);
47         if (cb_pontos_imagem_real.IsChecked == true)
48             desenharPontos(calcularMediaPontos(pontosReais), Brushes.Green);
49         if (cb_tracos_imagem_percebida.IsChecked == true)
50             desenharOssos(calcularMediaPontos(pontosPercebidos), Brushes.Red);
51         if (cb_tracos_imagem_real.IsChecked == true)
52             desenharOssos(calcularMediaPontos(pontosReais), Brushes.Green);
53         if (cb_todos_pontos_ip.IsChecked == true)
54             foreach (SkeletonPoint a in pontosPercebidos)
55                 desenharPontos(pontosPercebidos, Brushes.Red);
56         if (cb_todos_pontos_ir.IsChecked == true)
57             foreach (SkeletonPoint a in pontosReais)
58                 desenharPontos(pontosReais, Brushes.Green);
59         if (cb_todos_tracos_ip.IsChecked == true)
60         {
61             int total = pontosPercebidos.Count();
62             int pontos_por_ponto = total / qtPontos;
63             List<SkeletonPoint> temporaria;
64             for(int i=0; i<pontos_por_ponto;i++)
65             {
66                 temporaria = new List<SkeletonPoint>();
67                 for (int j = i; j < total; j = j + pontos_por_ponto)
68                     temporaria.Add(pontosPercebidos[j]);
69                 desenharOssos(temporaria, Brushes.Red);
70             }
71         }
72         if (cb_todos_tracos_ir.IsChecked == true)
73         {
74             int total = pontosReais.Count();
75             int pontos_por_ponto = total / qtPontos;
76             List<SkeletonPoint> temporaria;
77             for (int i = 0; i < pontos_por_ponto; i++)
78             {
79                 temporaria = new List<SkeletonPoint>();
80                 for (int j = i; j < total; j = j + pontos_por_ponto)
81                     temporaria.Add(pontosReais[j]);
82                 desenharOssos(temporaria, Brushes.Red);
83             }
84         }
85     }
86 }

```

Figura 26 - Desenho dos pontos e imagens da tela.

Na Figura 26, o método `Desenhar()` atende as solicitações do usuário através das opções de visualização mostradas no ponto 5 da Figura 9. Neste sentido, para cada opção marcada pelo usuário, a aplicação irá desenhar na tela a solicitação correspondente.

Desta forma, das linhas 45 a 83, todas as estruturas de controle `if` verificam se a opção correspondente de visualização foi marcada pelo usuário. Caso positivo, será realizada uma chamada ao método `desenharPontos` ou `desenharOssos`, ambos recebendo dois parâmetros: uma lista contendo pontos a serem desenhados, do tipo `SkeletonPoint`, e outro contendo a cor a ser utilizada para desenho, do tipo `Brushes`, nativo de aplicações `c#`.

O método `desenharPontos` é responsável por projetar no Canvas, os pontos contidos na lista, de acordo com a cor solicitada. A Figura 27 apresenta o Método para desenhar pontos no Canvas.

```
184 public void desenharPontos(List<SkeletonPoint> pontos, Brush corDesenho)
185 {
186     int diametroArticulacao = 7;
187     int larguraDesenho = 7;
188
189     foreach (SkeletonPoint ponto in pontos)
190     {
191         Ellipse objetoPonto = CriarComponenteVisualArticulacao(diametroArticulacao, larguraDesenho, corDesenho);
192         ColorImagePoint posicaoPonto = ConverterCoordenadasArticulacao(ponto);
193         double deslocamentoHorizontal = posicaoPonto.X - objetoPonto.Width / 2;
194         double deslocamentoVertical = (posicaoPonto.Y - objetoPonto.Height / 2);
195         Canvas.SetLeft(objetoPonto, deslocamentoHorizontal);
196         Canvas.SetTop(objetoPonto, deslocamentoVertical);
197         Canvas.SetZIndex(objetoPonto, 100);
198         canvas.Children.Add(objetoPonto);
199     }
200 }
```

Figura 27 - Método para desenhar pontos no Canvas

Na Figura 27, inicialmente foi determinado o diâmetro e a largura do ponto que será desenhado, conforme linhas 186 e 187. Em seguida, das linhas 189 a 199, e realizado um laço de repetição entre todos os pontos que foram passados como parâmetro para o método, onde a coordenada de cada ponto será transformada em coordenadas que serão desenhadas na imagem.

O método `desenharOssos` é responsável por realizar a ligação poligonal entre os pontos. A Figura 28 apresenta maiores detalhes sobre este método.

```
202 public void desenharOssos(List<SkeletonPoint> pontos, Brush corDesenho)
203 {
204     int larguraDesenho = 4;
205     Line objetoOsso;
206     ColorImagePoint posicaoArticulacaoOrigem;
207     ColorImagePoint posicaoArticulacaoDestino;
208     for (int i = 0; i < pontos.Count(); i++)
209     {
210         posicaoArticulacaoOrigem = ConverterCoordenadasArticulacao(pontos[i]);
211         if(pontos[i]==pontos.Last())
212             posicaoArticulacaoDestino = ConverterCoordenadasArticulacao(pontos[0]);
213         else
214             posicaoArticulacaoDestino = ConverterCoordenadasArticulacao(pontos[i+1]);
215         objetoOsso = CriarComponenteVisualOsso(larguraDesenho, corDesenho, posicaoArticulacaoOrigem.X,
216             posicaoArticulacaoOrigem.Y, posicaoArticulacaoDestino.X, posicaoArticulacaoDestino.Y);
217         canvas.Children.Add(objetoOsso);
218     }
219 }
```

Figura 28 - Método para realizar a ligação entre os pontos

Na Figura 28, inicialmente é determinado a largura e a cor do desenho, conforme linhas 204 e 205. Em seguida, nas linhas 206 e 207 será criado dois pontos, A e B, que serão

respectivamente o primeiro ponto e o segundo ponto da lista de pontos que foi passada por parâmetro.

Das linhas 208 a 218, os códigos se responsabilizam desenhar o traço, ou osso, entre o ponto A e B, bem como percorrer a lista de pontos de forma que A e B sejam sempre novos ossos a serem desenhados na tela, enquanto existirem pontos.

O resultado deste método é a projeção de linhas verdes ou vermelhas sobre os pontos na tela, de forma a desenhar um polígono. O ponto 1 da Figura 9 apresenta o resultado da utilização deste método tanto para pontos da imagem percebida, quanto pontos da imagem real.

3.4 Protocolo para Avaliação através do Procedimento Automatizado

Primeiramente, o avaliador deverá conversar com o paciente, deixando claro que se trata de uma avaliação automatizada, expondo as regras que este paciente deverá seguir.

O segundo passo a ser realizado é a verificação do Kinect, onde o avaliador deverá certificar que o Kinect está devidamente acoplado ao computador e conectado a uma fonte de energia elétrica. Em seguida, o mesmo deverá iniciar a aplicação, que fará automaticamente as devidas verificações do Kinect e, caso algo não esteja normal, o avaliador será advertido através de alertas.

Em seguida, o avaliador vendará os olhos do paciente e o guiará até a posição centralizada do campo de visão do Kinect. O avaliador deve certificar que o paciente esteja visível e centralizado no campo de visão do Kinect, respeitando a distância máxima de 3,5 metros de distância do dispositivo.

Depois, o avaliador verifica na aplicação se os pontos que deseja marcar estão visíveis no corpo do Paciente. Caso os pontos não estejam visíveis, o avaliador deverá reposicionar o paciente de forma que os pontos sejam visíveis. Caso os pontos estejam visíveis, o avaliador informará ao paciente que o mesmo deverá levantar as duas mãos acima da cabeça e acionará a funcionalidade de detectar paciente, clicando no botão “Detectar Paciente” ou comandando verbalmente através do comando “*Detect*”, caso a opção *Voice Control* esteja ativada. Se o avaliador desejar ativar esta opção sem se locomover até o computador, o mesmo pode acioná-la ou desativá-la através da instrução verbal “*Hello Kinect*”.

Se tudo ocorrer bem, o paciente será detectado e a barra de status apresentará o seu ID. Caso o mesmo não seja detectado, o avaliador deverá repetir o procedimento de detectar o paciente.

Com o paciente detectado, o avaliador deverá se posicionar atrás do paciente, de forma que seja possível tocar os pontos que deseja. É recomendado que o avaliador seja o menos visível quanto possível no campo de visão do Kinect, a fim de evitar que o Kinect perca precisão ao confundir paciente e avaliador. É altamente recomendável que o avaliador não atravesse à frente do paciente, pois isto acarretará na possível perda de referência do paciente. Isto acarretará na não identificação do paciente, a qual pode ser verificada pelo avaliador ao perceber se o círculo vermelho está na mão do paciente. Caso a referência seja perdida, o avaliador deverá repetir o passo de Detecção do Paciente.

Detectado o paciente, o avaliador deverá verbalizar ao paciente será tocado em algumas partes de seu corpo, as quais deverão ser projetadas com sua mão à frente, com o braço esticado, simbolizando a localização X e Y do ponto onde foi tocado.

Antes de começar a captura dos pontos, o avaliador deve se certificar de quantas regiões ao longo do corpo do paciente serão avaliadas. O número de regiões deve ser informado na aplicação no campo “Quantidade de Localizações a serem avaliadas”. Certificado de que a quantidade esteja correta, o avaliador poderá começar a captura dos pontos clicando no botão começar avaliação ou, caso o *Voice Control* esteja ativado, verbalizar o comando “*Start*”.

Neste momento, o avaliador começará a capturar os pontos referentes à imagem percebida do paciente. Poderão ser capturados quantos pontos o avaliador julgar necessário para cada localização no corpo do paciente. Contudo, a quantidade de pontos capturados deve ser igual para todas as localizações. Por exemplo, se forem capturados três pontos para cada posição e o avaliador havia definido cinco localizações ao longo do corpo do paciente, obviamente totalizará em quinze pontos capturados.

Para capturar um ponto, o avaliador deverá certificar que o *Voice Control* esteja ativado, pois somente será possível capturar pontos através da instrução verbal “Ok”. Ao ser pronunciado, o avaliador perceberá imediatamente que as coordenadas do ponto avaliado foram inseridas nas colunas da Imagem Percebida. Caso um ponto seja capturado

erroneamente, o avaliador poderá verbalizar a instrução “*Delete*”, que apagará o último ponto capturado.

Assim que o avaliador conseguir capturar todos os pontos que deseja, deverá parar a captura dos pontos através do comando “*Stop*”, bem como desativar o Voice Control através do comando “*Hello Kinect*”. Após isso, o avaliador retirará a venda do paciente e o informará que o procedimento será repetido, com a diferença que o paciente estará se vendo na tela à sua frente, e que deverá projetar fielmente os pontos que estão sendo tocados.

O avaliador não precisará capturar a mesma quantidade de pontos que foi capturado para cada localização corporal na composição da Imagem Percebida. Contudo, poderá capturar quantos pontos julgar necessário, respeitando a mesma quantidade de pontos para cada localização corporal.

Com isso, o avaliador ativará novamente o *Voice Control* através do comando verbal “*Hello Kinect*” e antes de começar a avaliação dos pontos reais, deverá selecionar a opção de captura de pontos reais, ou basta verbalizar a instrução verbal “*Change*” e verificar se a opção foi marcada automaticamente.

Selecionada a opção de marcar pontos referentes à imagem real, o avaliador poderá começar a captura dos pontos clicando no botão começar ou verbalizando o comando “*Start*”, tal como feito anteriormente. Assim, o avaliador capturará os pontos que achar necessário da mesma forma que o fez para Imagem Percebida, sendo que ao finalizar, deverá parar a captura dos pontos clicando no botão “Parar captura”, em vermelho, ou através do comando “*Stop*”, bem como desativar o *Voice Control*.

Neste momento, a avaliação estará finalizada e o avaliador pode dispensar o paciente do campo de visão do Kinect, bem como analisar os resultados clicando no botão “Resultados”.

4 CONSIDERAÇÕES FINAIS

Foi objetivo deste trabalho o desenvolvimento de uma aplicação que automatize o *Image Marking Procedure* para análise do esquema corporal utilizando o Microsoft Kinect. O estudo do procedimento tradicional foi essencial para o desenvolvimento da aplicação, denominada AutoIMP. Sem a concreta compreensão do funcionamento e do protocolo do procedimento tradicional, não seria possível alcançar os objetivos deste trabalho.

Além dos estudos relacionados ao procedimento de avaliação, foi necessária a contextualização acerca do funcionamento do Kinect e de sua SDK, que permitiram que as fases iniciais do desenvolvimento da aplicação pudessem ser iniciadas. Desse modo, o acoplamento do Kinect ao computador, instalação de drivers, configuração da SDK e implementação das funcionalidades básicas, como as de instanciação do dispositivo e ligação dos fluxos das câmeras, foram as primeiras atividades realizadas no que tange a etapa de desenvolvimento.

Com as funcionalidades básicas implementadas, foi utilizado um recurso nativo da SDK de desenvolvimento do Kinect, o mapeamento dos pontos do corpo humano, que dispensa esforços de desenvolvimento para reconhecer partes do corpo. Os pontos oferecidos nativamente foram analisados e, inicialmente, utilizados como referência para marcação da IC no que tange a IR, papel anteriormente cumprido pelo profissional avaliador.

Contudo, constatou-se que os pontos nativos não forneciam subsídios suficientes para que o profissional avaliador pudesse realizar a avaliação com a devida precisão, fornecendo dados incorretos, já que os pontos não eram os mesmos utilizados em um procedimento de avaliação tradicional utilizando o IMP.

Então, foi definida outra lógica para captura e análise dos pontos para composição da IR, na qual foi desabonada a utilização dos pontos oferecidos nativamente pelo Kinect, passando-se a utilizar apenas os pontos referentes às mãos do paciente, que forneciam respectivamente dados para composição da imagem real e imagem percebida.

Definida a nova abordagem, a aplicação foi desenvolvida objetivando a valorização da utilização de interfaces naturais, proporcionando maior liberdade e experiência ao avaliador e ao paciente. Para trabalhar com os fluxos de áudio, foi utilizada a classe `AudioStream`, que é responsável por possibilitar o acesso da aplicação aos microfones do Kinect. O fluxo de imagem, definido na classe `ImageStream`, possibilitou que a aplicação mostrasse em tempo

real o que estava sendo visto pela câmera RGB do Kinect, sendo importante por possibilitar que o paciente se veja na imagem apresentada à sua frente e marque os pontos reais que estão sendo sugeridos pelo avaliador. Não obstante, a classe `DepthStream` foi de vital importância neste trabalho, pois através dela foi possível capturar as informações relacionadas às distâncias entre os pontos, bem como possibilitar o rastreamento das mãos do paciente.

Com isso, foi possível capturar os pontos referentes a imagem percebida e imagem real, que passaram a ser fornecidos pelo próprio paciente. Deve ser ressaltado que a captura de vários pontos para cada localização corporal se mostrou eficiente no que tange a precisão dos resultados. Assim, a aplicação deixa a critério do avaliador a quantidade de pontos que o mesmo deseja avaliar no paciente.

Com os pontos capturados, foi possível calcular o IPC e classificar o indivíduo em hipoesquemático, hiperesquemático ou normal. Além disto, através da utilização da classe `Canvas`, foi possível desenhar os pontos na tela e realizar a ligação poligonal entre os mesmos, além de proporcionar diferentes tipos visualizações destas imagens, que proporciona uma análise sistematizada por parte do avaliador.

Por fim, foi criado o protocolo para avaliação automatizada utilizando a aplicação desenvolvida, que contém as regras e os passos a serem seguidos pelo avaliador para se obter um resultado satisfatório através da aplicação.

Conclui-se, então, que a automatização do procedimento através da aplicação pode ser utilizada para análise do esquema corporal dos indivíduos, além de proporcionar ao avaliador uma análise sistemática dos resultados do teste. Entretanto, por se tratar de unidades de medidas diferentes, não é possível realizar a comparação entre as medidas obtidas através do procedimento automatizado com as medidas obtidas através do procedimento manual, embora os resultados se aproximem.

Em suma, foi possível compreender que o Kinect é uma ferramenta poderosa, embora ainda sofra com algumas limitações de hardware, que são aceitáveis considerando seu baixo custo. Contudo, é objeto de trabalhos futuros, a utilização do novo Kinect, visando obter resultados mais precisos, bem como uma análise estatística em um grupo de indivíduos, através do procedimento automatizado e manual, que permitirá aferir, estatisticamente, a real precisão da aplicação e a viabilidade de sua utilização no dia a dia dos profissionais relacionados.

REFERÊNCIAS

- ASKEVOLD, F. **Measuring body image**. *Psychother Psychosom* 1975;26:71-7.
- BALLARD, D. H.; BROWN, C. M. **Computer Vision**. New Jersey: Prentice-Hall, Inc., 1982.
- BARBOSA, D. S. **Autopercepção da imagem corporal por idosos integrantes do grupo fisioterapêutico**: “Escola de posturas da 3ª idade” de João Pessoa – PB. Monografia (Bacharelado em Fisioterapia) – Universidade Federal da Paraíba, João Pessoa, 2004.
- CARDOSO, Gabriel Schade. **Microsoft Kinect: criando aplicações interativas com o Microsoft Kinect**. São Paulo: Casa do Código, 2013. 167 p.
- COSTA, Diogo; CAVALCANTI, José Carlos; COSTA, Dickson. A Cambrian Explosion of Robotic Life. **Management Science and Engineering**. Quebec . p. 98-105. 2011. Disponível em <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1751523>. Acesso em 18 de Abril de 2014.
- FERREIRA, Cíntia de Cássia *et al.* Análise da imagem e esquema corporal em indivíduos com diagnóstico de síndrome de pós-poliomielite. **Revista Neurociência**, São Paulo, n. 20, p.50-57, 2012.
- FONSECA, Cristiane Costa *et al.* Benefícios da estimulação perceptual corporal no esquema corporal de idosos. **Revista Brasileira de Geriatria e Gerontologia: Estimulação perceptual em idosos**, Rio de Janeiro, v. 2, n. 15, p.353-364, 2012.
- JUNGONG, H.; LING, S.; DONG, X.; JAMIE, S. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. **IEEE Transactions on Cybernetics**, v.43, n. 5, p. 1318 – 1334, out. 2013. Disponível em <<http://dx.doi.org/10.1109/TCYB.2013.2265378>>. Acessado em 26 de Maio de 2014.
- MATTOS, E. L. **Utilização de Biofeedback cinemático com sensor Kinect em sessões de fisioterapia**. 2012. 106 f. TCC (Graduação) – Curso de Ciência da Computação, Departamento de Centro de Ciências Tecnológicas da Terra e do Mar, Universidade Vale do Itajaí, Itajaí, 2012.
- MICROSOFT. **Visual Studio**. 2014. Disponível em: <<http://www.visualstudio.com/pt-br/visual-studio-homepage-vs.aspx>>. Acesso em 27 de Maio de 2014.

MICROSOFT. **JointType Enumeration**. 2014. Disponível em: <<http://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>>. Acesso em: 15 de maio de 2014.

MICROSOFT. **Microsoft Kinect: Kinect for Windows**. 2014. Disponível em: <<http://www.microsoft.com/en-us/kinectforwindows/>>. Acesso em: 20 de junho de 2014.

MILANO, D.; HONORATO, L. B. **Visão Computacional**. 2014. Disponível em: [http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNICAMP_visao Computacional.pdf](http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNICAMP_visao%20Computacional.pdf). Acesso em 26 de Maio de 2014.

MOURA, R. M. B. *et al.* A Importância da Imagem Corporal na Assimetria da Postura em Pacientes Hemiplégicos e Hemiparéticos. **Scire Salutis**, Aquidabã, v. 2, n. 2, p.16-28, 2012.

NEVES, L. A. P. *et al.* **Avanços em Visão Computacional**. Curitiba, PR: Ompipax Editora Ltda, 2012. 418p.

NOGUEIRA, Renatta Simões. **Dança de salão: percepção corporal, equilíbrio e qualidade de vida em idosos**. 97 f. Dissertação (Mestrado) - Curso de Ciências e Envelhecimento, Universidade São Judas Tadeu, São Paulo, 2012.

SHILDER, P. **A imagem do corpo**. São Paulo: Martins Fontes, 1994.

PEREIRA, Emy Suelen *et al.* Comportamento do esquema corporal do adolescente ao idoso. **Revista Brasileira de Ciências do Envelhecimento Humano**, Passo Fundo, v. 7, supl. 1, p. 20-28, 2010.

ROCHA, P. R.; DEFAVARI, A. H.; BRANDÃO, P. S. Estudo da viabilidade da utilização do Kinect como ferramenta no atendimento fisioterapêutico de pacientes neurológicos. In: Simpósio Brasileiro de Games e Entretenimento Digital, 11, 2012, Brasília: Sbc, 2012. p. 16 - 22. Disponível em: <<http://sbgames.org/sbgames2012/proceedings/papers/gamesforchange/g4c-04.pdf>>. Acesso em: 07 nov. 2012.

SEGHETO, Wellington *et al.* Proposta de categorização para análise da percepção corporal. In: I Simpósio Internacional de Imagem Corporal e o I Congresso Brasileiro de Imagem Corporal, 1, 2010, Campinas: Centro de Convenções da Unicamp, 2010. P. 1 - 6. **Anais**. Disponível em:

<<http://fefnet172.fef.unicamp.br/hotsites/imagemcorporal2010/cd/anais/trabalhos/portugues/Area4/IC4-05.pdf>>. Acesso em: 02 de Março de 2014.

THURM, Bianca Elisabeth. **Efeitos da dor crônica em atletas de alto rendimento em relação ao Esquema Corporal, Agilidade Psicomotora e estados de Humor.** 82 f. Dissertação (Mestrado) - Curso de Educação Física, Universidade São Judas Tadeu, São Paulo, 2007.

SALZANO, Fabio Tapia. **Revisão dos métodos empregados na avaliação da dimensão corporal em pacientes com transtornos alimentares.** 2011. 6 f. Dissertação (Mestrado) - Curso de Educação Física, Universidade São Judas Tadeu, São Paulo, 2011.

THURM, Bianca Elisabeth. **Perfil da Percepção Corporal e a insatisfação corporal em mulheres com transtornos alimentares – Uma proposta de intervenção corporal.** 2012. 57 f. Tese (Doutorado) – Curso de Educação Física, Universidade São Judas Tadeu, São Paulo, 2012.

YAMADA, F. A. A. *et al.* **Reconstrução de Objetos 3D utilizando Estruturas de Indexação Espacial com o Microsoft Kinect.** 2013. Disponível em:

<<http://www.ufjf.br/getcomp/files/2013/03/Reconstru%C3%A7%C3%A3o-de-Objetos-3D-utilizando-Estruturas-de-Indexa%C3%A7%C3%A3o-Espacial-com-o-Microsoft-Kinect.pdf>>. Acesso em 26 de Maio de 2014.