

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCUS ALMEIDA DA SILVEIRA

**Técnica de Navegação em Documentos
Utilizando Microsoft Kinect**

Trabalho de Graduação.

Prof. Dra. Carla Maria Dal Sasso Freitas
Orientadora

Porto Alegre, Dezembro de 2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Gostaria de agradecer minha professora orientadora, Prof. Dra. Carla Maria dal Sasso Freitas, pela oportunidade e experiência obtidas neste trabalho, me ajudando na implementação e revisão deste. Gostaria de agradecer minha família e a todos que estiveram comigo durante este processo, especialmente à minha namorada, Manuela, cujo apoio foi fundamental.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	5
LISTA DE FIGURAS.....	6
RESUMO.....	7
ABSTRACT	8
1 INTRODUÇÃO	9
1.1 Motivação	9
1.2 Objetivo do trabalho	9
2 INTERAÇÃO POR GESTOS	10
2.1 introdução	10
2.2 Kinect.....	11
2.3 SDKs	11
2.3.1 OpenNI/NITE	11
2.3.2 Kinect SDK	12
3 TRABALHOS RELACIONADOS	13
3.1 Kinect SDK Dynamic Time Warping Gesture Recognition	13
3.2 FAAST	13
3.3 Gesture Recognition With a Wii Controller	14
3.4 Recognizing Hand Gestures with Microsoft's Kinect	14
4 TÉCNICA PROPOSTA	15
4.1 Descrição da técnica e metodologia utilizada.....	15
5 AVALIAÇÃO	18
5.1 Sujeitos e Procedimento	18
5.1.1 Sujeitos	18
5.1.2 Procedimento de teste	18
5.2 Resultados e Discussão	23
5.2.1 Caracterização dos usuários de teste	23
5.2.2 Resultados do experimento.....	25
6 CONCLUSÃO.....	31
REFERÊNCIAS	32

LISTA DE ABREVIATURAS E SIGLAS

SDK	Software Development Kit
PC	Personal Computer
E3	Electronic Entertainment Expo
NES	Nintendo Entertainment System
PS	Playstation
DTW	Dynamic Time Warping
FAAST	Flexible Action and Articulated Skeleton Toolkit
KIPS	Kinect Presentation System

LISTA DE FIGURAS

Figura 2.1: Joints do esqueleto.....	15
Figura 4.1: Regiões definidas para detecção.....	15
Figura 5.1: Usuário em teste.....	19
Figura 5.2: Questionário de avaliação.....	20
Figura 5.3: Usuários Destros / Canhotos.....	23
Figura 5.4: Usuários que já usou Kinect ou outro dispositivo semelhante.....	23
Figura 5.5: Frequência com que usuários apresentam slides.....	24
Figura 5.6: Como usuários costumam apresentar seus slides.....	24
Figura 5.7: Gráfico de autoclassificação do nível de usabilidade computacional dos usuários.....	24
Figura 5.8: Uso estimado do método desenvolvido.....	25
Figura 5.9: Facilidade de uso.....	25
Figura 5.10: Ajuda técnica para uso.....	26
Figura 5.11: Integração das funções do sistema.....	26
Figura 5.12: Inconsistências no sistema.....	27
Figura 5.13: Comandos reconhecidos.....	27
Figura 5.14: Tempo adequado para execução dos comandos.....	27
Figura 5.15: Confiança no sistema.....	28
Figura 5.16: Necessidade de conhecimento prévio.....	28
Figura 5.17: Facilidade de uso estimada.....	28
Figura 5.18: Sistema apresenta vantagens em relação ao modelo convencional.....	29
Figura 5.19: Mais tempo de uso aumentaria a facilidade de uso.....	29
Figura 5.20: Vale a pena investir no método desenvolvido.....	29
Figura 5.21: Conjunto total das respostas.....	30

RESUMO

Com a popularização de novos dispositivos de interação, temos disponíveis cada vez mais métodos para interagir com computadores utilizando interfaces mais naturais com usuários. Este trabalho propõe o desenvolvimento de uma técnica de interação entre usuários e computador, empregando um desses novos dispositivos, o Kinect. A técnica utiliza a posição dos braços do usuário para reconhecer gestos pré-definidos e transmiti-los a uma aplicação em background como se fossem comandos do teclado, os quais são interpretados e executados na aplicação.

O trabalho mostra o uso da técnica, com uma aplicação de apresentação de slides, onde o método não convencional de interação resulta em maior facilidade de uso.

Palavras-Chave: Kinect, Interação homem computador, Apresentação de slides.

Documents Browsing using Microsoft Kinect Technique

ABSTRACT

With the popularization of new interaction devices, we have at our disposal increasingly more interaction methods for building natural user interfaces. This work proposes the development of a human-computer interaction technique, using one of these new devices, Microsoft Kinect. The technique takes the position of the user's arms to recognize predefined gestures, and sends them to a background application as keystrokes. The application interprets and executes the commands.

This work demonstrates the technique in a slide presentation task. We show that such non-conventional interaction technique is easier to use in the chosen experimental task.

Keywords: Kinect, Human-computer interaction, slide presentation.

1 INTRODUÇÃO

1.1 Motivação

Desde os anos 80 computadores são largamente utilizados através de teclado e mouse, e vemos filmes de ficção científica nos quais os computadores são usados como se fosse por magia, seja entendendo qualquer comando dado por voz, ou gestos em uma tela holográfica. O Microsoft Kinect[5] é um passo para tornar interfaces naturais uma realidade no dia a dia do uso computacional.

O trabalho foi motivado pelo interesse em investigar técnicas de interação usando o Microsoft Kinect e implementar seu uso em uma aplicação comum, como a de apresentar slides.

1.2 Objetivo do trabalho

Este trabalho tem como objetivo investigar o poder de detecção de movimentos do Microsoft Kinect e avaliar a experiência de usuários através de uma técnica simples de interação desenvolvida. Com este método de interação, diversas atividades realizadas pelo computador, poderão ser feitas livremente sem a necessidade da proximidade com o aparelho ou da utilização de acessórios extras como mouse sem fio, deixando o usuário com as mãos livres para fazer o que quiser interagindo com o computador ao mesmo tempo.

A técnica foi implementada para uso com uma aplicação de apresentação de slides. Assim poderemos apresentar slides utilizando o computador, sem apertar nenhuma tecla ou botão, usando apenas os nossos braços para avançar ou retroceder a apresentação.

2 INTERAÇÃO POR GESTOS

2.1 Introdução

A interação entre homem e computador sempre foi um dos quesitos mais importantes quando se pensa em como facilitar o uso dos computadores para os usuários existentes e os que nunca tocaram em um computador. Os dispositivos de entrada tradicionais muitas vezes dificultam o ingresso de usuários menos experientes neste meio digital, sejam pessoas com dificuldades motoras para controlar um mouse, ou pessoas que se perdem no meio de tantas teclas em um teclado. Por esses motivos sempre houve pesquisas para desenvolver essa área, a fim de facilitar atividades executadas no computador.

Um dos grandes agentes de avanço nessa área atualmente é a indústria de jogos eletrônicos, que busca sempre inovar a experiência fornecida por seus consoles de videogames. Dentre as empresas dessa indústria, a Nintendo foi sempre a mais inovadora nesse quesito, criando consoles portáteis como Game & Watch[3], o Game Boy[2] e o Virtual Boy[18], lançando diversos dispositivos de entrada diferentes dos tradicionais joysticks, como a pistola Zapper[19] em 1984 e Power Glove[10] em 1989, ambas para o NES [7].

Em 2006, foi lançado o Nintendo Wii [8], alvo de muitas críticas sobre seus gráficos inferiores aos concorrentes Playstation 3 e Xbox 360 e líder de mercado durante muitos anos após seu lançamento. Com a popularização do Wii e seu revolucionário Wiimote, as pessoas começaram a utilizar o controle em aplicações além dos jogos de Wii, desenvolvendo diversos projetos de aplicações controladas por movimentos do dispositivo.

Em 2009, a Microsoft e a Sony anunciaram seus produtos concorrentes para diminuir a liderança da Nintendo devido ao seu controle por movimentos. A Sony anunciou o PS Move[11], controle com as mesmas características do Wiimote, mas com melhor sensibilidade, e a Microsoft anunciou o Project Natal, futuramente conhecido por Microsoft Kinect.

2.2 Kinect

O Microsoft Kinect é um dispositivo de entrada desenvolvido pela Microsoft para seu console Xbox 360. Ele possui 3 câmeras, uma RGB convencional e duas usadas para detectar a distância do usuário, e um array de microfones utilizado para captar apenas a fala do usuário, removendo o som emitido pela TV.

O Kinect foi lançado pela divisão jogos da Microsoft para combater o sucesso surpreendente do Nintendo Wii e seu controle Wiimote com seus jogos mais casuais que só utilizavam movimentos e estava arrecadando cada vez mais usuários, muitos dos quais nunca haviam tocado em um videogame, e conquistando sozinho uma fatia enorme do mercado. Na feira de games E3 de 2009 o Kinect foi mostrado ao público com algumas demonstrações de conceito feita pela Microsoft e outros estúdios parceiros. Nesta mesma E3 a Sony também apresentou o seu concorrente do Nintendo Wii, o PS Move, que era basicamente igual em funções ao Wiimote, mas com uma melhor sensibilidade.

Com o lançamento do Microsoft Kinect, o sucesso foi tão grande que a Microsoft resolveu investir pesado em outros usos para seu novo dispositivo, a fim de mantê-lo em constante uso e não cair no esquecimento como mais um “dispositivo da moda”. A Microsoft já anunciou a intenção de uma integração total do Kinect com o próximo sistema operacional, Windows 8, prometendo revolucionar a maneira de como o novo OS será utilizado.

Por limitações do dispositivo, não é possível detectar movimentos próximos ao Kinect, o que impede o uso deste para tarefas mais cotidianas com aquelas realizadas quando sentados na frente do computador. Mas já foi anunciada uma nova versão do Kinect, que promete melhorar a sua precisão, aumentando seu ‘campo de visão’, possibilitando detectar gestos faciais a uma distância de 50cm. Eles prometem ainda melhor seu array de microfones sendo possível reconhecer estados emocionais dos usuários com base no tom da voz e na expressão facial, de modo que o Kinect possa ser usado no dia a dia, para tarefas habituais.

2.3 SDKs

2.3.1 OpenNI/NITE

A primeira SDK a ser lançada foi a OpenNI da empresa israelense PrimeSense responsável pela tecnologia utilizada no sensor 3D do Microsoft Kinect [9]. Essa SDK foi lançada no final de 2010 e utilizada largamente como forma de “hackear” o Microsoft Kinect para ser utilizado por outra máquina (computador) além do Xbox 360, que era seu principal e único alvo originalmente.

Atualmente, ela ainda possui algumas vantagens em relação à SDK oficial, que está na sua versão beta 2, como um framework para reconhecimento de gestos com mãos e é multissistemas operacionais, funcionando em Linux, Windows (XP, Vista e 7) e Mac OS X.

Possui, entretanto, algumas desvantagens, como fazer uso de uma série de drivers que precisam ser instalados manual e individualmente, o que torna seu uso extremamente improvável para usuários finais. Ela também não tem suporte ao

reconhecimento de áudio do Kinect. A SDK requer uma pose de calibragem para o reconhecimento do usuário no início das aplicações.

Foi esta SDK que mostrou à Microsoft que seu dispositivo era mais que apenas um concorrente do Nintendo Wii e convenceu a Microsoft de que o Kinect poderia ser usado para muitas outras coisas.

2.3.2 Kinect SDK

A Kinect SDK é a SDK oficial da Microsoft, lançada em junho de 2011 (versão beta) [6]. Essa versão traz alguns códigos de amostra feitos pela Microsoft, cada um demonstrando uma função específica da SDK, como reconhecimento de áudio, amostragem das diferentes câmeras, detecção do esqueleto e colisão do esqueleto com objetos. A SDK possui classes prontas para essas funções, tornando simples a utilização de cada uma das funções.

A SDK permite até 6 usuários detectados ao mesmo tempo, possibilitando o uso multiusuário ou a restrição de apenas um dos usuários em frente à câmera ser considerado pela aplicação.

O esqueleto detectado pela SDK é armazenado numa estrutura que contém diferentes segmentos do corpo. Cada um desses segmentos é definido pelas coordenadas dos seus extremos, as *joints*. A estrutura possui 21 *joints*: cabeça, ombro central (pescoço), ombros direito e esquerdo, cotovelos direito e esquerdo, pulsos direito e esquerdo, mãos direita e esquerda, espinha, lados direito, esquerdo e centro do quadril, joelhos direito e esquerdo, calcanhares direito e esquerdo e pés direito e esquerdo. A figura 2.1 abaixo mostra cada uma dessas *joints*.

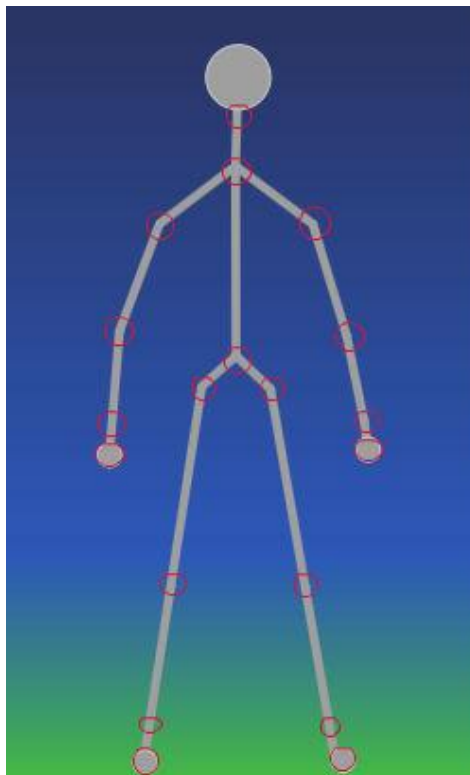


Figura 2.1: Joints do esqueleto.

3 TRABALHOS RELACIONADOS

Devido ao Kinect ser um dispositivo relativamente barato e a recente disponibilidade de uma SDK oficial da Microsoft, estão surgindo cada vez mais projetos utilizando este dispositivo [16][17]. A seguir estão alguns destes projetos.

3.1 Kinect SDK Dynamic Time Warping Gesture Recognition

O Kinect SDK Dynamic Time Warping (DTW) Gesture Recognition [4] é um projeto utilizando a SDK oficial da Microsoft, lançado em julho de 2011, em que o Kinect é utilizado para gravar gestos feitos por um usuário, armazená-los em um arquivo de texto e interpretar as ações do usuário com base nestes gestos armazenados. Para gravar os gestos, o usuário escolhe em uma lista o nome do gesto, ativa o modo de gravação e executa o gesto em 32 frames (um pouco mais de um segundo). O arquivo de texto contém as coordenadas X e Y de 12 pontos do modelo do usuário em 32 frames diferentes. Esse modelo é composto por pontos que representam a parte superior do corpo: cabeça, centro dos ombros, ombro direito, ombro esquerdo, coluna, centro do quadril, cotovelo direito, cotovelo esquerdo, pulso direito, pulso esquerdo, mão direita e mão esquerda.

O reconhecimento dos gestos gravados é feito utilizando um algoritmo baseado em vetores e vizinhos próximos dos pontos superiores do corpo do usuário. Isso quer dizer que o usuário pode realizar os gestos na velocidade que desejar, não importando a velocidade em que eles foram gravados. Mas isso também significa que todos os pontos superiores são considerados para o reconhecimento do gesto. Sendo assim, se um gesto de movimentação da mão direita é gravado enquanto a mão esquerda está abaixada, ele só será reconhecido quando a mão esquerda estiver abaixada.

Como o arquivo gravado possui apenas as coordenadas X e Y, não é possível reconhecer movimentos em 3D, como a mão se aproximando da câmera.

3.2 FAAST

O Flexible Action and Articulated Skeleton Toolkit (FAAST) [1] é um middleware desenvolvido com base nas SDKs não oficiais OpenNI e NITE. Ele foi criado por um grupo de pessoas da USC (University of Southern California) para integrar o Kinect à softwares já existentes que não possuem integração nativa ao dispositivo.

3.3 Gesture Recognition with a Wii Controller

O Gesture Recognition with a Wii Controller é um projeto realizado na Universidade de Oldenburg – Alemanha, utilizando o controle de Nintendo Wii, o Wiimote, para realizar o reconhecimento de gestos [14]. O projeto utiliza o acelerômetro para capturar as entradas do controle, permitindo o treinamento de gestos arbitrários dos usuários para reconhecimento destes posteriormente em sistemas, como a visualização de fotos em uma televisão.

3.4 Recognizing Hand Gestures with Microsoft's Kinect

O Recognizing Hand Gestures with Microsoft's Kinect propõe uma técnica para reconhecimento de gestos com a mão utilizando o Kinect [15]. Este trabalho utilizou as SDKs OpenNI e NITE para sua implementação.

Foi utilizado um trabalho anterior de integrantes do Departamento de Ciência da Computação da Universidade de Stanford para identificar a posição da mão do usuário e capturar uma sequência de imagens de 64x64, em que a mão deve estar com base na posição detectada. Com estas imagens da mão, é analisada a imagem capturada e atribuída a cada pixel uma probabilidade de este pixel fazer parte da mão. Após realizada toda a análise e identificar a posição estimada da mão, é analisada uma sequência de posições da mão para obter-se um gesto.

Esses gestos são reconhecidos com base em posições da mão no início e fim da sequência de imagens, como, por exemplo, uma mão aberta no início e fechada no fim da sequência representa o gesto de pegar algo.

4 TÉCNICA PROPOSTA

4.1 Descrição da técnica e metodologia utilizada

Para a realização dos objetivos deste trabalho, foi utilizada a versão beta da SDK oficial da Microsoft, para facilitar o uso em máquinas com sistema operacional Windows 7 ou mais atual. Posteriormente, já que a Microsoft está investindo fortemente nessa interação do Kinect com PC, não será necessária a instalação manual de diversos drivers.

A técnica desenvolvida utiliza os *streams* de vídeo e profundidade gerados. Com o *stream* de vídeo, detectam-se as posições do esqueleto, seus segmentos e *joints*. Calcula-se o ângulo formado pelo segmento do antebraço (figura 4.1), e com esse valor, executa-se a ação definida ou não. As ações definidas para testar a técnica foram executar “Page Down” com o braço direito e “Page Up” com o braço esquerdo.

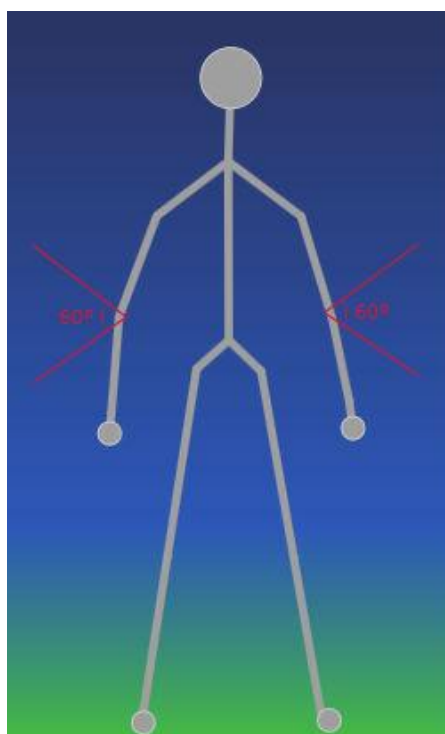


Figura 4.1: Regiões definidas para detecção.

Para a elaboração da técnica, foi usado uma das amostras de código presentes na Kinect SDK, o ShapeGame em C#. Este código possui algumas funções já prontas para a atualização da estrutura do esqueleto 2D com base nos *streams* de vídeo e uma janela de visualização do esqueleto. A estrutura do esqueleto possui as coordenadas X e Y dos *joints* relativas ao canto superior esquerdo da janela.

Para facilitar o uso das coordenadas, elas foram normalizadas com valores de 0 a 1, e a variável Y foi invertida para considerar o canto inferior esquerdo como ponto de origem do sistema de coordenadas. Foi criada uma função para calcular o ângulo formado por dois pontos, dadas as suas coordenadas, e retorna um valor de 0 a 359, e outra função para verificar se um ângulo está entre dois outros ângulos dados, que retorna um valor booleano.

Foram definidos dois padrões de reconhecimento, antebraço direito formando um ângulo de 90° com o braço direito (180° no sistema de coordenadas do universo) e antebraço esquerdo formando um ângulo de 90° com o braço esquerdo (0° no sistema de coordenadas do universo). Foi considerada para esses gestos uma região de 60° para determinar quando o usuário deseja executar a ação, como é visto na figura 4.1.

Para evitar possíveis execuções indesejadas, foi definido um intervalo mínimo de tempo (0,75s), no qual o antebraço deve permanecer dentro da região definida, e também um intervalo de tempo (1,5s) para a ação ser repetida, em que o usuário pode permanecer com o antebraço na região e a ação será repetida a cada intervalo de tempo.

Também foi definida uma altura máxima do cotovelo, permitindo que o usuário possa gesticular mais livremente e apontar detalhes na tela. Esta altura foi calculada com base nos *joints* da cabeça e do centro do quadril, reconhecendo os comandos apenas quando o cotovelo estiver a menos de 20% desta altura.

Com esses valores definidos, a cada frame capturado pelo Kinect, varre-se a estrutura do esqueleto em busca das coordenadas que contêm os segmentos dos *joints* dos pulsos e cotovelos direito e esquerdo. Ao identificar o segmento que contenha um dos dois lados, verifica-se se o ângulo formado está dentro da região definida, utilizando a função criada para verificar se o ângulo formado pelos pontos do pulso e cotovelo está entre dois outros ângulos limites. Em caso positivo, ativa-se uma *flag* indicando que o braço entrou na região de detecção. Caso o braço permaneça na região, a ação definida para ele é executada a cada 1,5 segundos. Em caso negativo, verifica-se se a *flag* está ligada para detectar que o braço acabou de sair da região de detecção. Caso o braço saia da região de detecção em menos de 1,5 segundos e mais de 0,75 segundos, a ação é executada uma vez; caso ele fique menos de 0,75 segundos dentro da região, a ação não é executada. Ao sair da região de detecção, desativa-se a *flag*.

A implementação dessa técnica pode ser vista a seguir:

```
private void RecognizeGesture(Dictionary<Bone, BoneData> segments, JointID joint)
{
    double minAngle, maxAngle;
    double normX1, normY1, normX2, normY2;
    string comando;
    Point3d head, hip, joint1, joint2;
    head.X = 0; head.Y = 0;
    hip.X = 0; hip.Y = 0;
    joint1.X = 0; joint1.Y = 0;
    joint2.X = 0; joint2.Y = 0;
    switch (joint)
    {
```



```

case JointID.WristRight:
    minAngle = 145.0;
    maxAngle = 205.0;
    gesture = gestureNext;
    acao = nextPage;
    comando = "{PGDN}";
    break;
case JointID.WristLeft:
    minAngle = 325.0;
    maxAngle = 35.0;
    gesture = gesturePrevious;
    acao = previousPage;
    comando = "{PGUP}";
    break;
default:
    minAngle = 0.0;
    maxAngle = 0.0;
    gesture = gestureNext;
    acao = nextPage;
    comando = "{Down}";
    break;
}
try
{
    foreach (var segmentPair in segments)
    {
        if (segmentPair.Key.joint1 == JointID.Head)
        {
            head.X = segmentPair.Value.seg.x1;
            head.Y = this.ActualHeight - segmentPair.Value.seg.y1;
        }
        if (segmentPair.Key.joint1 == JointID.HipCenter)
        {
            hip.X = segmentPair.Value.seg.x1;
            hip.Y = this.ActualHeight - segmentPair.Value.seg.y1;
        }
        if (segmentPair.Key.joint1 == joint)
        {
            joint1.X = segmentPair.Value.seg.x1;
            joint2.X = segmentPair.Value.seg.x2;
            joint1.Y = this.ActualHeight - segmentPair.Value.seg.y1;
            joint2.Y = this.ActualHeight - segmentPair.Value.seg.y2;
        }
        if (head.X != 0 && hip.X != 0 && joint1.X != 0)
        {
            normX1 = joint1.X / this.ActualWidth;
            normX2 = joint2.X / this.ActualWidth;
            normY1 = joint1.Y / this.ActualHeight;
            normY2 = joint2.Y / this.ActualHeight;
            //Entrou na região definida
            if (BetweenAngle(Angle(normX1, normY1, normX2, normY2),
minAngle, maxAngle) && ((joint2.Y - hip.Y) / (head.Y - hip.Y)) < 0.20))
            {
                if (!gesture)
                {
                    gesture = true;
                    gestureTime = DateTime.Now;
                    FlyingText.NewFlyingText(screenRect.Width / 30, new
Point(screenRect.Width / 2, screenRect.Height / 2), "gesture Detected");
                }
                else
                {
                    gestureDur = DateTime.Now - gestureTime;
                    //Se continua na região definida por tempo estendido
executa ação

```


detectar se os pulsos estão a frente dos cotovelos utilizando a coordenada Z acrescentada à estrutura.

Esta técnica funciona razoavelmente bem, considerando que o usuário se mantenha virado diretamente para o Kinect ou no sentido oposto. A técnica tem problemas quando o usuário começa a rotacionar o corpo, provocando identificações incorretas, tornando-a inaceitável nestas condições.

Com isso, caso o usuário fique virado de costas para o Kinect, as ações permanecem iguais para o Kinect: braço que está à esquerda do seu campo de visão executa “Page Down” e braço que está à direita do seu campo de visão executa “Page Up”.

A implementação desse reconhecimento pode ser vista a seguir:

```
private void RecognizeFacing(Dictionary<Bone, BoneData> segments)
{
    Point3d shoulderLeft, shoulderRight, hipCenter, wristLeft, wristRight,
    elbowLeft, elbowRight;
    shoulderLeft.X = 0;    shoulderLeft.Z = 0;
    shoulderRight.X = 0;   shoulderRight.Z = 0;
    hipCenter.X = 0;      hipCenter.Z = 0;
    wristLeft.X = 0;      wristLeft.Z = 0;
    wristRight.X = 0;     wristRight.Z = 0;
    elbowLeft.X = 0;      elbowLeft.Z = 0;
    elbowRight.X = 0;     elbowRight.Z = 0;
    try
    {
        foreach (var segmentPair in segments)
        {
            switch (segmentPair.Key.joint1)
            {
                case JointID.HipCenter:
                    hipCenter.X = (float)segmentPair.Value(seg.x1);
                    hipCenter.Z = (float)segmentPair.Value(seg.z1);
                    break;
                case JointID.ElbowRight:
                    elbowRight.X = (float)segmentPair.Value(seg.x1);
                    elbowRight.Z = (float)segmentPair.Value(seg.z1);
                    shoulderRight.X = (float)segmentPair.Value(seg.x2);
                    shoulderRight.Z = (float)segmentPair.Value(seg.z2);
                    break;
                case JointID.WristRight:
                    wristRight.X = (float)segmentPair.Value(seg.x1);
                    wristRight.Z = (float)segmentPair.Value(seg.z1);
                    elbowRight.X = (float)segmentPair.Value(seg.x2);
                    elbowRight.Z = (float)segmentPair.Value(seg.z2);
                    break;
                case JointID.ElbowLeft:
                    elbowLeft.X = (float)segmentPair.Value(seg.x1);
                    elbowLeft.Z = (float)segmentPair.Value(seg.z1);
                    shoulderLeft.X = (float)segmentPair.Value(seg.x2);
                    shoulderLeft.Z = (float)segmentPair.Value(seg.z2);
                    break;
                case JointID.WristLeft:
                    wristLeft.X = (float)segmentPair.Value(seg.x1);
                    wristLeft.Z = (float)segmentPair.Value(seg.z1);
                    elbowLeft.X = (float)segmentPair.Value(seg.x2);
                    elbowLeft.Z = (float)segmentPair.Value(seg.z2);
                    break;
                default:
                    break;
            }
        }
    }
}
```

```
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }

    if ((shoulderLeft.X != 0) && (shoulderRight.X != 0))
    {
        if (!BetweenAngle(Angle(shoulderLeft.X, shoulderLeft.Z,
shoulderRight.X, shoulderRight.Z), 340.0, 20.0))
        {
            facing = Facing.Side;
            facingLabel = "Side";
        }
        else
        {
            if ((wristRight.Z > elbowRight.Z) || (wristLeft.Z > elbowLeft.Z))
            {
                facing = Facing.Back;
                facingLabel = "Back";
            }
            else
            {
                facing = Facing.Front;
                facingLabel = "Front";
            }
        }
    }
}
```

5 AVALIAÇÃO

Com o objetivo de avaliar a usabilidade da técnica, realizou-se uma sessão de testes com pessoas de diversas áreas, faixas etárias e experiência computacional, usando a técnica desenvolvida para interação com PowerPoint. A seguir apresenta-se a descrição do procedimento de avaliação e os resultados obtidos com o experimento.

5.1 Sujeitos e procedimento

5.1.1 Sujeitos

A avaliação foi feita com uma amostra de nove usuários. Destes nove, quatro homens e cinco mulheres de diversas áreas, como estudantes, médicos, psicólogos, engenheiro e dona de casa. A faixa etária foi bastante variada, sendo um usuário com 16 anos, cinco entre 23 e 31 anos e três usuários com mais de 53 anos.

O modo de operação de slides habitualmente utilizado pelos usuários que apresentam slides ficou dividido entre mouse/teclado e os que solicitam auxílio de alguém para passar os slides.

5.1.2 Procedimento de teste.

O teste consistiu em uma apresentação pelos usuários de um conjunto de slides no PowerPoint utilizando o Microsoft Kinect para avançar e retroceder os slides.

Para a realização dos testes de avaliação, preparou-se uma sala com um notebook conectado a um projetor e o Microsoft Kinect (Figura 5.1). Cada usuário foi instruído a respeito dos comandos pré-definidos para o teste. Foi criada uma aplicação com a visualização do esqueleto do usuário, que utiliza a técnica descrita anteriormente e executa os comandos “Page Up” e “Page Down”, como também descrito anteriormente. Esta aplicação mantém-se minimizada e é executado o Microsoft PowerPoint em modo “Apresentação de Slides”.

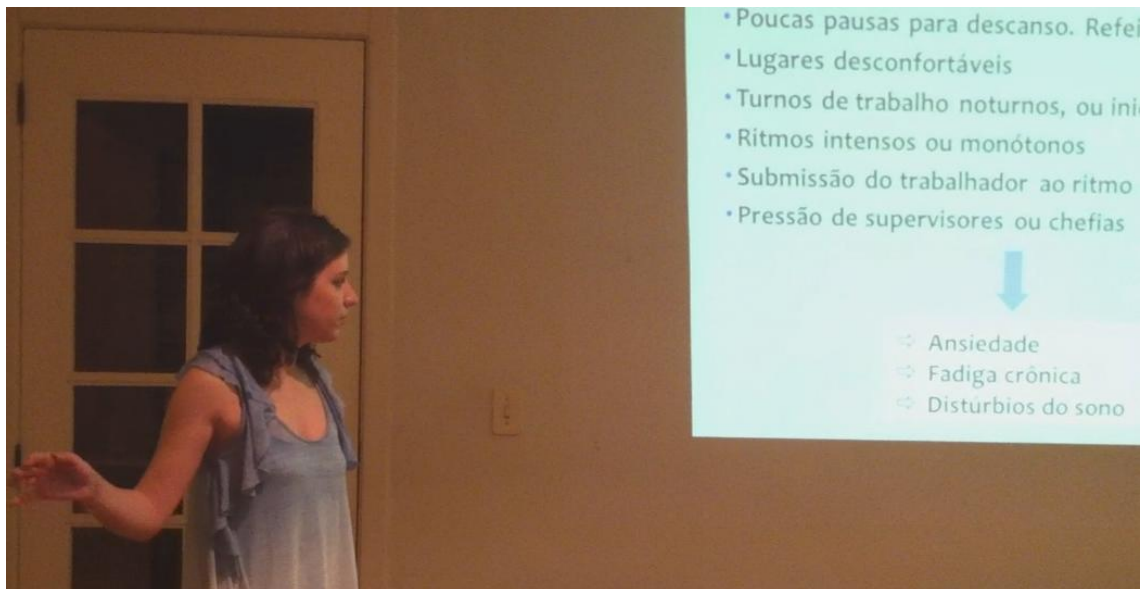


Figura 5.1: Usuário em teste

Após o experimento, cada usuário respondeu perguntas em questionário *online* com duas sessões, uma com questões pessoais e outra com perguntas relacionadas ao experimento. As perguntas relacionadas ao experimento foram feitas utilizando a escala Likert, para obter posteriormente uma visão geral sobre a média das respostas.

O questionário pode ser observado na figura 5.2 [12].

Questionário Kinect Presentation System

Responda este questionário honestamente após realizar o experimento com o KIPS.

* Required

Informações pessoais

Nome *

Idade *

Profissão *

Destro / Canhoto *

 ▾

Você já usou o Kinect ou algum dispositivo semelhante de detecção de movimentos? *

- Sim
 Não

Com que frequência você costuma apresentar slides usando computador? *

 ▾

Como você costuma apresentar slides *

 ▾

Como você classifica seu grau de familiaridade com computadores?

familiaridade = facilidade de uso do computador

1 2 3 4 5

Muito Ruim Muito Bom

Figura 5.2: Questionário de avaliação

Informações sobre o experimento

Eu usaria este sistema com frequência *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Achei o sistema fácil de usar *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Não acho que eu precisaria da ajuda de um técnico para usar o sistema *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Achei que as funções desse sistema estavam bem integradas *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Não achei que haviam muitas inconsistências nesse sistema *

1 2 3 4 5

Discordo fortemente Concordo fortemente

O sistema reconheceu todos comandos executados *

1 2 3 4 5

Discordo fortemente Concordo fortemente

O tempo para execução dos comandos está adequado *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Figura 5.2 (continuação): Questionário de avaliação

Me senti confiante usando este sistema *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Não precisei aprender muitas coisas antes de conseguir usar este sistema *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Imagino que a maioria das pessoas aprenderiam a usar este sistema rapidamente *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Este sistema apresenta vantagens em relação ao modelo convencional *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Acredito que com mais tempo de uso, poderei utilizar este método com mais facilidade *

1 2 3 4 5

Discordo fortemente Concordo fortemente

Acredito que vale a pena investir neste método para substituir o método convencional *

método convencional = teclado / mouse / controle remoto

1 2 3 4 5

Discordo fortemente Concordo fortemente

Comentários gerais

Sugestões / críticas / relato da experiência

Submit

Figura 5. 2 (continuação): Questionário de avaliação

5.2 Resultados e Discussão

5.2.1 Caracterização dos usuários de teste

Através das respostas do questionário, pode-se caracterizar os usuários da seguinte forma (Figuras 5.3 a 5.7) [13]:

1. Apenas um dos usuários era canhoto.
2. Três usuários já usaram dispositivos de detecção de movimentos anteriormente (Nintendo Wii), mas apenas como diversão. Um destes três também já havia utilizado o Kinect com o Xbox 360 para jogar.
3. Dois dos usuários nunca apresentaram slides em computadores. Dois deles apresentam slides frequentemente em palestras e aulas. Os outros 5 apresentam slides em apresentações de trabalhos, em média 3 por semestre.
4. O modo de operação de slides habitualmente utilizado pelos usuários que apresentam slides ficou dividido entre mouse/teclado conectados diretamente por cabo ao computador e os que solicitam auxílio de alguém para passar os slides.
5. Apenas um dos usuários considerou sua familiaridade com computadores muito ruim. Os outros ficaram bem distribuídos entre razoável e muito boa familiaridade.

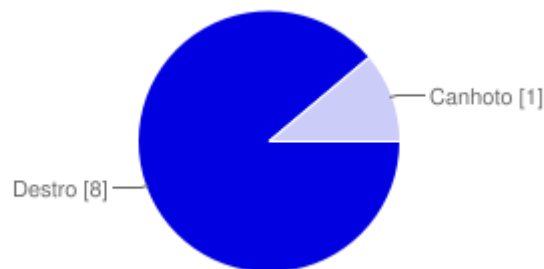


Figura 5.3: Usuários Destros / Canhotos: Destro / Canhoto

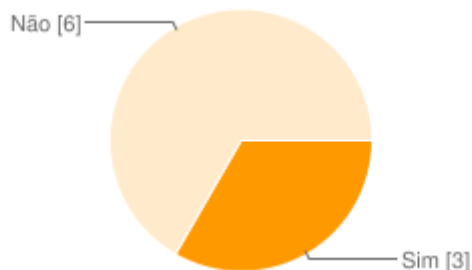


Figura 5.4: Usuários que já usou Kinect ou outro dispositivo semelhante: Você já usou o Kinect ou algum dispositivo semelhante de detecção de movimentos? (Sim / Não)

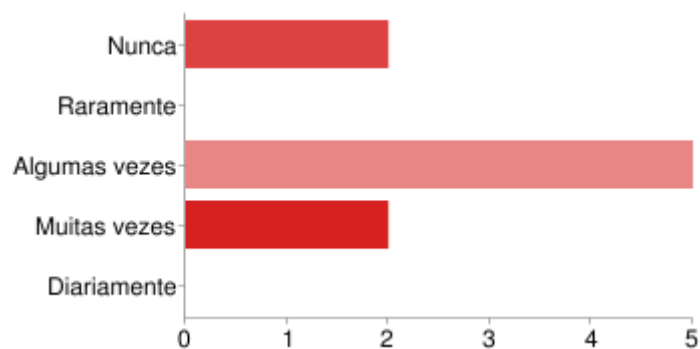


Figura 5.5: Frequência com que usuários apresentam slides: Com que frequência você costuma apresentar slides usando computador? (Nunca / Raramente / Algumas vezes / Muitas vezes / Diariamente)

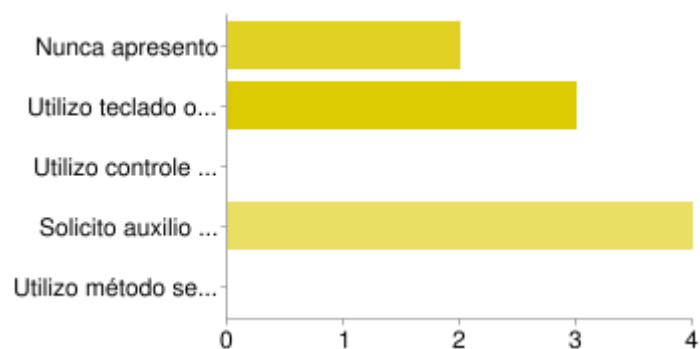


Figura 5.6: Como usuários costumam apresentar seus slides: Como você costuma apresentar slides? (Nunca apresento / Utilizo teclado ou mouse / Utilizo controle remoto ou mouse sem fio / Solicito auxílio a alguém para passar slides / Utilizo método semelhante ao do experimento)

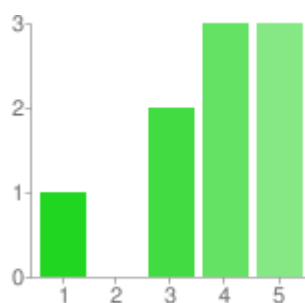


Figura 5.7: Gráfico de autoclassificação do nível de familiaridade computacional dos usuários: Como você classifica seu grau de familiaridade com computadores? (1: Muito ruim, 5: Muito bom)

5.2.2 Resultados do experimento

A seguir temos os resultados das questões relacionadas à sessão de testes realizada, precedidos por conclusões e comentários.

A maioria dos usuários usaria este sistema com frequência, caso ele estivesse disponível. Apenas um usuário não usaria o sistema, pois nunca apresenta slides e considera-se ‘muito ruim’ com computadores, levando à conclusão de que ele não usaria o sistema, pois não apresentaria slides.

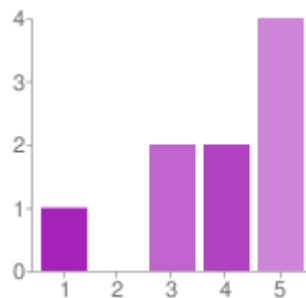


Figura 5.8: Uso estimado do método desenvolvido: Eu usaria este sistema com frequência (1=Discordo fortemente – 5=Concordo fortemente)

Um dos usuários achou o sistema de difícil uso por considerar que melhorias ainda devem ser feitas no software ou até no hardware para obter-se uma detecção mais eficiente. Os outros usuários acharam fácil, apesar de alguns terem tido mais dificuldades na detecção dos comandos.

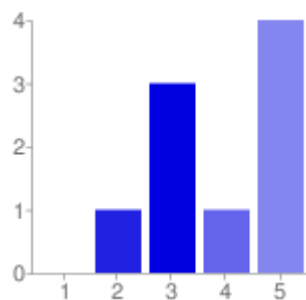


Figura 5.9: Facilidade de uso: Achei o sistema fácil de usar (1=Discordo fortemente – 5=Concordo fortemente)

Os usuários consideram que precisariam de ajuda técnica para utilizar o sistema, caso seja necessário montar o Kinect e colocá-lo no melhor local possível.

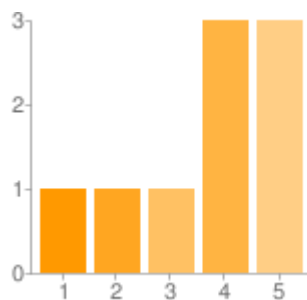


Figura 5.10: Ajuda técnica para uso: Não acho que eu precisaria da ajuda de um técnico para usar o sistema (1=Discordo fortemente – 5=Concordo fortemente)

Todos os usuários consideraram as funções definidas do sistema bem integradas. Alguns desejavam mais algumas funções, além de avançar e retroceder, como um 'laser' na tela, controlado pela mão.

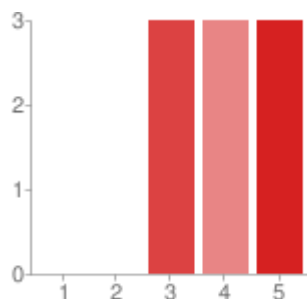


Figura 5.11: Integração das funções do sistema: Achei que as funções desse sistema estavam bem integradas (1=Discordo fortemente – 5=Concordo fortemente)

A maioria dos usuários considerou que o sistema ainda possui algumas inconsistências (Figura 5.12) apesar de a maioria considerar que ele reconheceu a maior parte dos movimentos executados (Figura 5.13). Alguns consideraram os intervalos de tempo para a execução dos comandos muito altos (Figura 5.14). Outros usuários consideraram ruim o comando de avançar a cada 1,5 segundos, desejando que avançasse apenas uma vez ao manter o braço dentro da região de detecção.

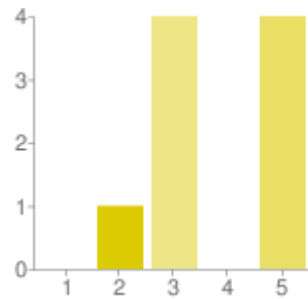


Figura 5.12: Inconsistências no sistema: Não achei que haviam muitas inconsistências nesse sistema (1=Discordo fortemente – 5=Concordo fortemente)

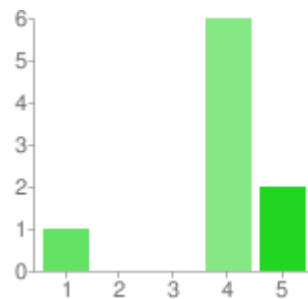


Figura 5.13: Comandos reconhecidos: O sistema reconheceu todos os comandos executados (1=Discordo fortemente – 5=Concordo fortemente)

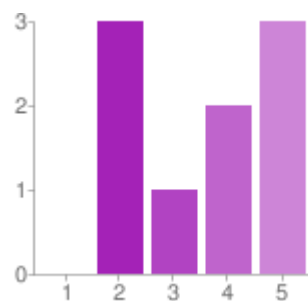


Figura 5.14: Tempo adequado para execução dos comandos: O tempo para execução dos comandos está adequado (1=Discordo fortemente – 5=Concordo fortemente)

Os usuários distribuíram-se crescentemente nas respostas relativas à confiança no sistema, mostrando que a maioria sentiu-se relativamente confiante, tendo apenas 2 usuários que não consideraram o sistema confiável o suficiente.

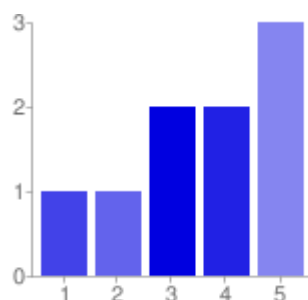


Figura 5.15: Confiança no sistema: Senti-me confiante usando este sistema (1=Discordo fortemente – 5=Concordo fortemente)

Um usuário considerou que o sistema necessitava de muitas instruções para funcionar perfeitamente (Figura 5.2.16) e por isso imagina que a maioria das pessoas não aprenderiam a usar o sistema rapidamente (Figura 5.2.17). A maioria considerou fácil e aceitável a quantidade de instruções para o uso do sistema.

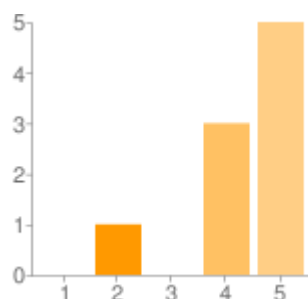


Figura 5.16: Necessidade de conhecimento prévio: Não precisei aprender muitas coisas antes de conseguir usar este sistema (1=Discordo fortemente – 5=Concordo fortemente)

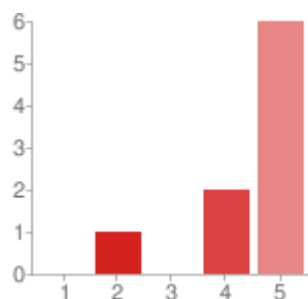


Figura 5.17: Facilidade de uso estimada: Imagino que a maioria das pessoas aprenderiam a usar este sistema rapidamente (1=Discordo fortemente – 5=Concordo fortemente)

Todos consideram que a técnica desenvolvida é mais vantajosa do que o modelo convencional (Figura 5.2.18). Eles acreditam que com um tempo de uso mais estendido podem utilizar o sistema perfeitamente (Figura 5.2.19) e que este método de interação deveria substituir o método convencional atual (Figura 5.2.20).

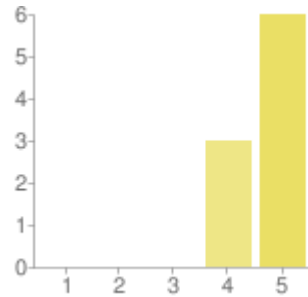


Figura 5.18: Sistema apresenta vantagens em relação ao modelo convencional: Este sistema apresenta vantagens em relação ao modelo convencional (1=Discordo fortemente – 5=Concordo fortemente)

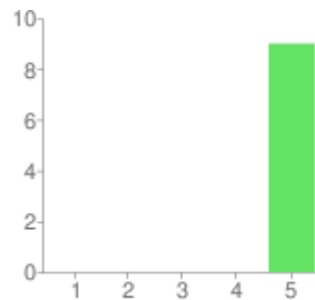


Figura 5.19: Mais tempo de uso aumentaria a facilidade de manuseio: Acredito que com mais tempo de uso, poderei utilizar este método com mais facilidade (1=Discordo fortemente – 5=Concordo fortemente)

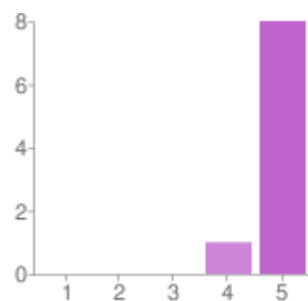


Figura 5.20: Vale a pena investir no método desenvolvido: Acredito que vale a pena investir neste método para substituir o método convencional (1=Discordo fortemente – 5=Concordo fortemente)

Observando os usuários durante a avaliação percebeu-se que a maioria dos usuários teve uma curva de aprendizado relativamente baixa. Em média, em dez minutos de uso da técnica, os usuários estavam dominando os comandos programados para passar e voltar os slides com bastante eficiência.

Pode-se observar também uma certa perda de desempenho no reconhecimento quando mais de uma pessoa estava na frente da câmera, tornando inviável o uso da técnica apresentada com mais de um usuário ao mesmo tempo. Também foi observado que para a técnica reconhecer perfeitamente os comandos, o usuário precisa ficar de frente para o Microsoft Kinect e o ambiente não pode estar completamente escuro.

Observa-se que uma quantidade considerável dos usuários considerou o tempo para execução inadequado, preferindo um tempo menor para a detecção do comando. A maioria dos usuários considerou o sistema consistente, mesmo não tendo reconhecido 100% dos comandos desejados por eles. Todos os usuários acreditam que um método como este desenvolvido deveria substituir o atualmente utilizado, tornando apresentações por slides em palestras e aulas mais dinâmicas e fáceis de serem realizadas.

Nos relatos apresentados pelos usuários pode-se perceber como este método de apresentação pode ser útil em ocasiões como palestras, em que o apresentador precisa segurar um microfone, uma caneta laser para apontar detalhes nos slides e um controle remoto para passar slides, ou em aulas, nas quais o professor ou pede a um aluno ficar passando os slides, tirando-lhe a atenção completa, ou fica voltando ao computador a fim de passar os slides, ou segura controle remoto, e várias canetas para escrever no quadro. Nos relatos também foi mencionada a ocorrência de acabarem as pilhas do controle remoto, interrompendo o andamento da aula ou palestra.

Como pode ser visto na contabilização de todas as respostas (Figura 5.21), os usuários tiveram uma experiência bem positiva utilizando a técnica desenvolvida.

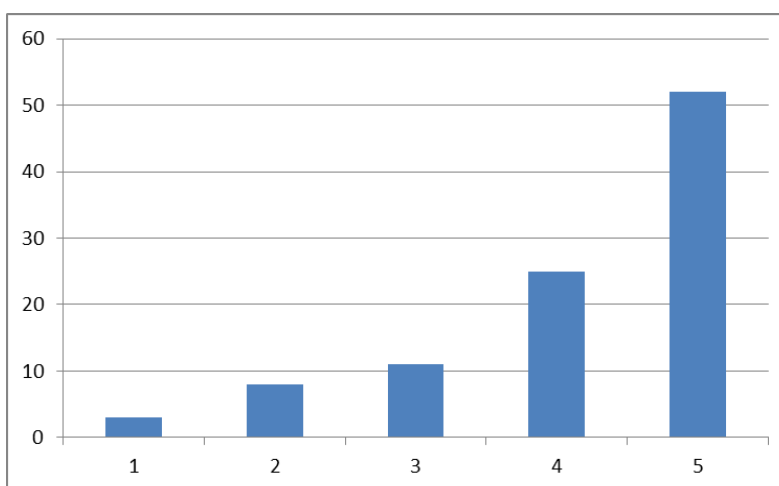


Figura 5.21: Conjunto total das respostas

6 CONCLUSÃO

Este trabalho propôs desenvolver uma técnica de interação baseada em gestos para controlar tarefas comandadas por teclado, como apresentação de slides, utilizando movimentos detectados pelo Microsoft Kinect. A técnica foi avaliada realizando sessões de testes com usuários reais.

O trabalho mostrou que, com uma técnica de detecção de gestos suficientemente sensível e precisa, é possível melhorar uma atividade como a de apresentação de slides, tornando-a mais interativa, interessante e fácil.

Com a implementação demonstrada, é fácil adicionar novos comandos e gestos à técnica, podendo ampliar seu uso para outras aplicações além da utilizada nos testes.

REFERÊNCIAS

- [1] **Flexible Action and Articulated Skeleton Toolkit**, [S.l.:s.n]. Disponível em: <[FAAST](#)> Acesso em: dez. 2011.
- [2] **Game Boy**, [S.l.:s.n]. Disponível em: <[Game Boy](#)>. Acesso em: dez. 2011.
- [3] **Game & Watch**, [S.l.:s.n]. Disponível em: <[Game & Watch](#)>. Acesso em: dez. 2011.
- [4] **Kinect SDK Dynamic Time Warping Gesture Recognition**, [S.l.:s.n]. Disponível em: <[KinectDTW](#)> Acesso em: dez. 2011.
- [5] **Kinect**, [S.l.:s.n]. Disponível em: <[Kinect](#)>. Acesso em: dez. 2011.
- [6] Microsoft Corporation. **Kinect SDK** e documentação, [S.l.:s.n]. Disponível em: <[kinectforwindows](#)>. Acesso em: dez. 2011.
- [7] **Nintendo Entertainment System**, [S.l.:s.n]. Disponível em: <[NES](#)>. Acesso em: dez. 2011.
- [8] **Nintendo Wii**, [S.l.:s.n]. Disponível em: <[Wii](#)>. Acesso em: dez. 2011.
- [9] **OpenNI e NITE SDK** e documentação, [S.l.:s.n]. Disponível em: <[openkinect](#)>. Acesso em: dez. 2011.
- [10] **Power Glove**, [S.l.:s.n]. Disponível em: <[Power Glove](#)>. Acesso em: dez. 2011.
- [11] **Playstation Move**, [S.l.:s.n]. Disponível em: <[PS Move](#)>. Acesso em: dez. 2011.
- [12] **Questionário**, [S.l.:s.n]. Disponível em: <[Questionário](#)>. Acesso em: dez. 2011.
- [13] **Questionário e respostas**, [S.l.:s.n]. Disponível em: <[Respostas](#)>. Acesso em: dez. 2011.
- [14] SCHLÖMER, T; POPPINGA, B; HENZE, N; BOLL, S. **Gesture Recognition with a Wii Controller**. Oldenburg: University of Oldenburg; OFFIS Institute for Information Technology: [s.n].
- [15] TANG, M. **Recognizing Hand Gesture with Microsoft's Kinect**. Palo Alto: Department of Electrical Engineering of Stanford University: [s.n], March 2011.
- [16] **Trabalhos relacionados utilizando Kinect**, [S.l.:s.n]. Disponível em: <[kinecthacks.com](#)>. Acesso em: dez. 2011.

[17] **Trabalhos relacionados utilizando Kinect**, [S.l.:s.n]. Disponível em: <kinecthacks.net>. Acesso em: dez. 2011.

[18] **Virtual Boy**, [S.l.:s.n]. Disponível em: <[Virtual boy](#)>. Acesso em: dez. 2011.

[19] **Zapper**, [S.l.:s.n]. Disponível em: <[Zapper](#)>. Acesso em: dez. 2011.