



UNIVERSIDADE PAULISTA

LUCAS OLIVEIRA DOS SANTOS

SOFTWARE DE ANÁLISE DO MOVIMENTO HUMANO

CAMPINAS

2017

LUCAS OLIVEIRA DOS SANTOS

SOFTWARE DE ANÁLISE DO MOVIMENTO HUMANO

Trabalho de conclusão de curso para
obtenção do título de graduação em
Engenharia Mecatrônica apresentado à
Universidade Paulista – UNIP.

Orientador: Prof. Doutor Luís Gustavo de Mello Parecêncio

**CAMPINAS
2017**

CIP - Catalogação na Publicação

dos Santos, Lucas Oliveira
SOFTWARE DE ANÁLISE DO MOVIMENTO HUMANO / Lucas Oliveira
dos Santos. - 2017.
4000 f.

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto
de Ciência Exatas e Tecnologia da Universidade Paulista, Campinas,
2017.

Área de Concentração: Acadêmica.

Orientador: Prof. Dr. Luís Gustavo de Mello Parecência.

1. Análise de movimentos humanos. 2. Microsoft Kinect. 3. Software
Labview. 4. Software Matlab. I. Parecência, Luís Gustavo de Mello
(orientador). II. Título.

LUCAS OLIVEIRA DOS SANTOS

SOFTWARE DE ANÁLISE DO MOVIMENTO HUMANO

Trabalho de conclusão de curso para
obtenção do título de graduação em
Engenharia Mecatrônica apresentado à
Universidade Paulista – UNIP.

Aprovado em:

BANCA EXAMINADORA

_____/_____/_____
Prof. Dr. Luís Gustavo de Mello Paracêncio

_____/_____/_____
Prof. Dr. Luís Fernando Lamas de Oliveira

_____/_____/_____
Prof. MSc. Roberto Canonico

_____/_____/_____
Prof. MSc. Roberto Canonico

**CAMPINAS
2017**

DEDICATÓRIA

Aos meus pais, irmão e minha noiva Thaise Xavier, e a toda minha família que, com muito carinho e apoio ,não mediram esforços para que eu chegasse até esta etapa de minha vida.

“Observador, analista de comportamento humano, estudo sua mente e capto seus futuros movimentos descobrindo seus pensamentos. Buscando ser expert em expressões faciais. Sempre estou de olho nas coisas e nos seres, analiso seus gestos, falas, postura e traço seu perfil. Busco enxergar o mundo entre as máscaras e mentiras. Dificilmente sou manipulado e tenho conhecimento do que você pensa, acha e quer.”

(Marcos Vinicius Trindade)

RESUMO

O uso de tecnologias para estudo do movimento humano vem crescendo exponencialmente a cada ano em diversas áreas, tais como: Fabricantes de chuteiras, tênis para corrida, lutadores, fisioterapeutas, na análise da evolução do tratamento, e muitas outras áreas.

O estudo principal deste trabalho tem como objetivo desenvolver um sistema de análise que fornecerá informações importantes sobre os movimentos do corpo humano, como aceleração de um membro, velocidade e deslocamento. Essas informações poderão ser interpretadas por médicos, esportistas, engenheiros e qualquer um que esteja realizando pesquisas com movimentos do corpo humano e precise de dados para estudos no desenvolvimento de teses, estudos na área acadêmica e de melhoria de produtos.

Este trabalho consistirá no uso de sensor de movimento Microsoft Kinect™ para obter dados dos membros, onde será programado em Labview™ como software de aquisição em tempo real com uma interface otimizada para o usuário. Todos os cálculos serão desenvolvidos em MatLab™.

Palavras-chave: Kinect, MatLab, Labview, Movimentos, Articulações

ABSTRACT

The use of technologies for motion studies is growing up each year in diferentes aeras , such as : Shoes Manufactores , Running shoes ,by fighters, by Physiotherapists in studies of treatment evolution and in much others areas.

Tha main study of this project is develop analytic system where will be able to supply important infomations about human motion , such as : Joint Aceleration , Speed and displacement. Informations that could be understood by medics , athletes , engineers, fighters and anyone that needs develop studies in academic area

This Project will consist in use of Microsoft Kinect sensor™ to get joints positions where will be programed in LabView™ as real time aquisition with an optimizes user interface.All calculs will be developed in MatLab™.

Keywords : Kinect,Matlab,Labview ,Joints,Limbs

SUMÁRIO

1. Introdução	1
2. Teoria.....	2
2.1 O que é o Kinect?	2
2.3 Modelo 3D das articulações detectadas.....	3
2.4 Labview	4
2.7 Acelerômetro de Smartphone	5
2.8 Geometria Analítica.....	6
2.9 Equações do movimento.....	7
3. Materiais e Métodos.....	8
3.1 Kinect	8
3.2 Smartphone.....	9
3.3 SDK para Kinect.....	10
4. Resultados	11
4.1 Apresentação do Software.....	11
4.2 Fluxo de dados.....	12
4.3 Controles do Kinect.....	13
4.4 Calibração com Acelerômetro Externo.....	14
4.5 Calibração por meio de um corpo Humano.....	15
4.6 Microsoft SDK V1.8	16
4.7 Anatomia	16
4.8 Análise de Caso	16
4.9 Análise do estudo de Caso	19
4.10 Custo final do Projeto	20
5. Discussão e Conclusão	21
6. Referência Bibliográfica	22
Apêndice A: Script Principal Matlab Coleta de dados do Kinect	23
Apêndice B : Script de tratamento dos dados Matlab.....	25
Apêndice C : Acesso	31

1. Introdução

O estudo do movimento humano vem crescendo proporcionalmente com o avanço da tecnologia de modo a nos permitir melhorar as técnicas de análise e seus dados. Neste estudo proposto como tema de trabalho de conclusão de curso, irei apresentar um software capaz de analisar em tempo real os movimentos do corpo humano que extrairá informações como: aceleração de um membro, velocidade e o deslocamento em todo o tempo que o usuário estiver aquisitando dados para projetar todos as informações em gráficos tridimensionais.

Neste projeto serão apresentados os métodos para aquisição de dados por meio de um sensor (Microsoft Kinect Sensor) e respectivas análises da interferência que o sensor possa sofrer.

O software chamado de *Body Analyser* fornecerá dados em espaciais de cada membro que poderão ser exportados para arquivos Excel.

2. Teoria

2.1 O que é o Kinect?

Kinect é um sensor desenvolvido pela Microsoft em 2010 para ser usado inicialmente apenas em consoles da linha de vídeo games *Xbox*. O hardware contém duas câmeras infravermelhas, uma RGB de 640x480 pixels, um motor de elevação que permite um ajuste de $\pm 30^\circ$, quatro microfones posicionados nas laterais e uma taxa de amostragem de 30 quadros por segundo.

O sensor trabalha em dois estágios: primeiro ele constrói um mapa de profundidade emitindo luzes infravermelhas e analisando os padrões de luz. Depois infere as posições com um banco de dados com mais de um milhão de exemplos. (MacCormik, 2017).

Ao realizar a inspeção de peças com câmeras de visão tomam-se cuidados com o ambiente e dentro do ambiente podemos destacar iluminação, pois qualquer variação da luz no objeto gera ruídos na imagem. Consequentemente o algoritmo de tratamento em grande parte não conseguirá ter um alto índice de repetibilidade. Com o uso de câmeras de profundidade esses problemas deixam de existir, pois tudo que é coletado se resume em apenas profundidades dos objetos no espaço.

(Borentein, 2012)



Figura 1 - Luz estruturada (MacCormik, 2017)

2.3 Modelo 3D das articulações detectadas

O algoritmo de detecção do sensor trabalha em duas configurações, são elas:

- Modo Stand: O algoritmo trabalha inferindo 20 posições do corpo humano e consegue no máximo detectar até 6 jogadores.
- Modo Sitted : O algoritmo para esse modo permite apenas inferir 10 posições do corpo humano e também detecta até 6 jogadores.

Centro do quadril = 1;	Direita do cotovelo = 10;
Espinha = 2;	Pulso Direito = 11;
Centro do Ombro = 3;	Mão Direita = 12;
Cabeça = 4;	Esquerda esquerda = 13;
Ombro esquerdo = 5;	Joelho Esquerdo = 14;
Cotovelo Esquerda = 6;	Tornozelo Esquerdo = 15;
Pulso Esquerdo = 7;	Pé esquerdo = 16;
Mão Esquerda = 8;	Quadril Direita = 17;
Direita de Ombro = 9;	Joelho Direito = 18;
	Direito do tornozelo = 19;
	Pé Direito = 20;

Tabela 1 – Posições do corpo possíveis de detectar pelo Kinect

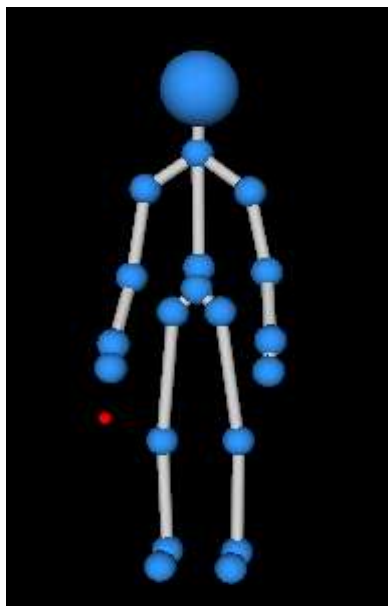


Figura 2 – Esqueleto tridimensional (Ahmed, Paul, & Gavrilova, 2017)

2.4 Labview

Labview é um ambiente de desenvolvimento integrado desenvolvido para engenheiros e cientistas. O software é considerado uma linguagem G otimizando a programação.

O *Body Analyzer* é programado em Labview pois necessitará trabalhar com o Windows, utilizando as API's da Microsoft.

O Labview conta com diversas bibliotecas para serem integradas em desenvolvimentos de softwares de pequeno, médio e grande porte. Muito utilizado na indústria de automação, para controle de máquinas pela fácil integração dos Hardwares da National Instruments e CLP's de mercado (Instruments, 2017).

2.7 Acelerômetro de Smartphone

Os smartphones modernos são equipados com inúmeros sensores que permitem um conforto e usabilidade aprimorada ao usuário. Um dos sensores que compõem os smartphones é o acelerômetro, usado por programadores de jogos e para uso no Android na coleta de giros de tela.

Os acelerômetros medem acelerações no intervalo de $\pm 2g$. A resolução pode chegar a 0,002g, na prática é limitada pela estrutura mecânica do celular. (Vieira & Aguiar, 2013)

O acelerômetro não mede propriamente a aceleração, mas sim a grandeza. Se colocarmos um smartphone com a tela para cima teremos a seguinte leitura:

$$a_x = a_y = 0, a_z = 9,82m/s^2$$



Figura 5 - Eixos do acelerômetro do Iphone (Vieira & Aguiar, 2013)

2.8 Geometria Analítica

Para calcular as distâncias dos membros serão usadas equações de geometria analítica, para analisarmos os deslocamentos e calcularmos posteriormente a velocidade e aceleração.

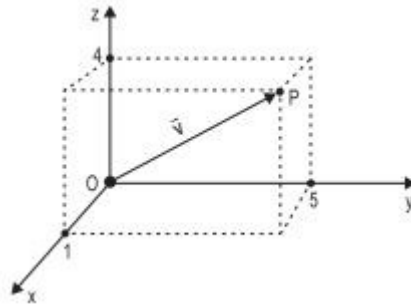


Figura 6 = Dados espaciais do vetor \vec{V} (Venturi, 2015)

$$\vec{V} = (1, 5, 4)$$

Como abuso de notação temos ainda que:

$$\vec{V} = (P - O) = P$$

Para obtermos as distâncias podemos usar:

$$|\vec{V}| = \sqrt{x^2 + y^2 + z^2}$$

Para calcularmos o ângulo entre dois membros podemos usar a seguinte equação:

$$\cos \theta = \frac{(u * v)}{(|u| * |v|)}$$

2.9 Equações do movimento

Para calcular a velocidade e aceleração dos membros serão usadas as seguintes equações do movimento uniformemente variado.

Para calcularmos a distância percorrida no espaço de tempo do mesmo membro usaremos:

$$S = S_0 + V_0 * t + \frac{a * t^2}{2}$$

Velocidade é a variação da posição no tempo, com isso podemos usar a equação:

$$V = V_0 + a * t$$

Aceleração é a variação da velocidade no tempo:

$$a = \frac{\Delta V}{\Delta T}$$

3. Materiais e Métodos

Neste capítulo serão descritos os equipamentos e os métodos usados para o desenvolvimento deste trabalho, são eles:

- Kinect 360 versão
- Smartphone Samsung Galaxy J3 2016

3.1 Kinect

O sensor escolhido foi o kinect V1.0 por seu hardware de baixo custo e com fácil integração ao Labview por meio do SDK V1.8 da Microsoft lançada para desenvolvedores.

Foram encontradas algumas limitações e para não influenciar no resultado final o ambiente de análise deve estar sob as seguintes condições:

- Para a maior performance na detecção dos membros não podem ser usados: calças com barras (ex: Calça jeans), blusas, toucas, luvas, calçados abertos ou fechados.

É indicado roupas:

Homens: Bermuda tactel e camisa.

Mulheres: Calça *legging*, camisa e cabelo preso estilo rabo de cavalo.

- Necessário tripé para Kinect com 70 cm de altura. Ver figura abaixo:



Figura 3 - Tripé para Kinect com 70 cm de altura.

3.2 Smartphone

O smartphone escolhido para o projeto foi o Samsung Galaxy J3 2016 equipado com acelerômetro BMA255 fabricado pela Bosch, com range de $\pm 2g / \pm 4g / \pm 8g / \pm 16g$ e resolução de 12bits. (Bosch, 2014)

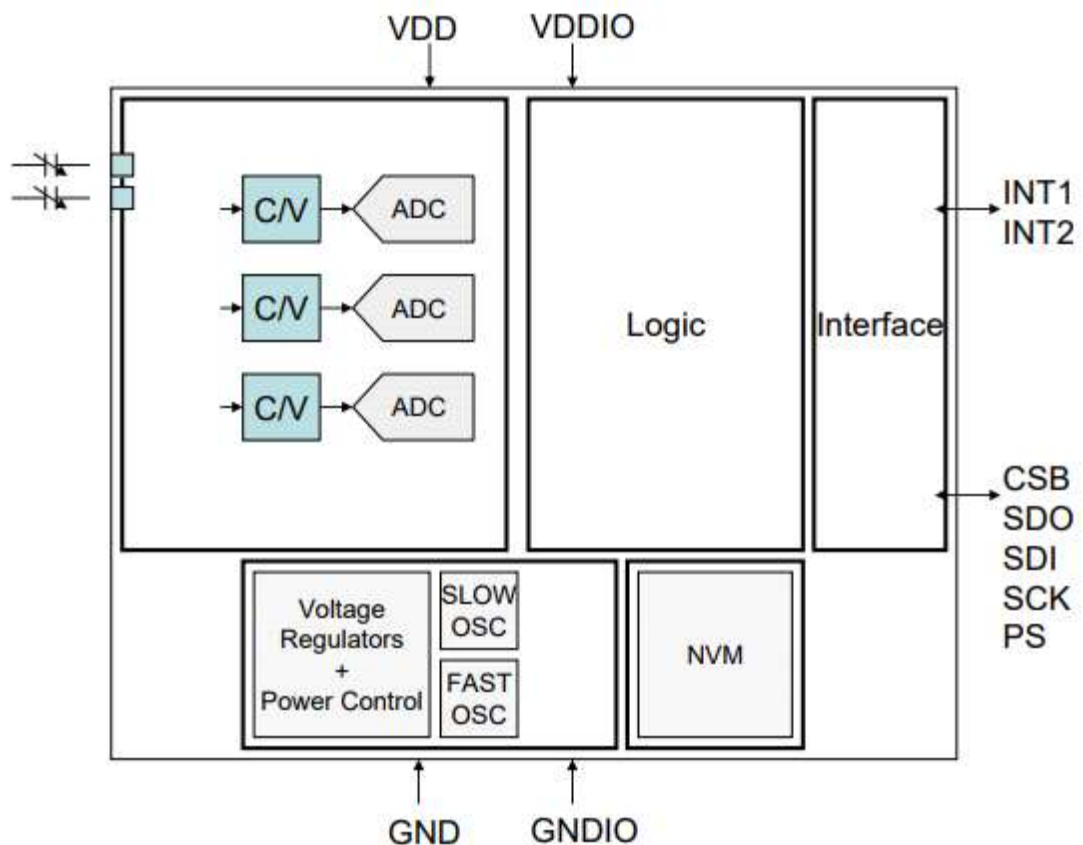


Figura 4 – Circuito do Acelerômetro BMA255

Por meio do aplicativo Amarino 2.0 serão enviados os valores medidos pelo acelerômetro via bluetooth para o software *Body Analyser*. (Amarino, 2017)

3.3 SDK para Kinect

Para realizar a integração com qualquer ambiente de programação Windows é necessário instalar o Kinect Windows SDK 1.8, que pode ser baixado gratuitamente dos repositórios da Microsoft. (Microsoft, 2017)

4. Resultados

4.1 Apresentação do Software

Body Analyser é um software desenvolvido em Labview, além de adquirir dados do sensor Kinect é possível obter em tempo real o esqueleto se movimentando e dados de aceleração, deslocamento e velocidade de um membro previamente selecionado.

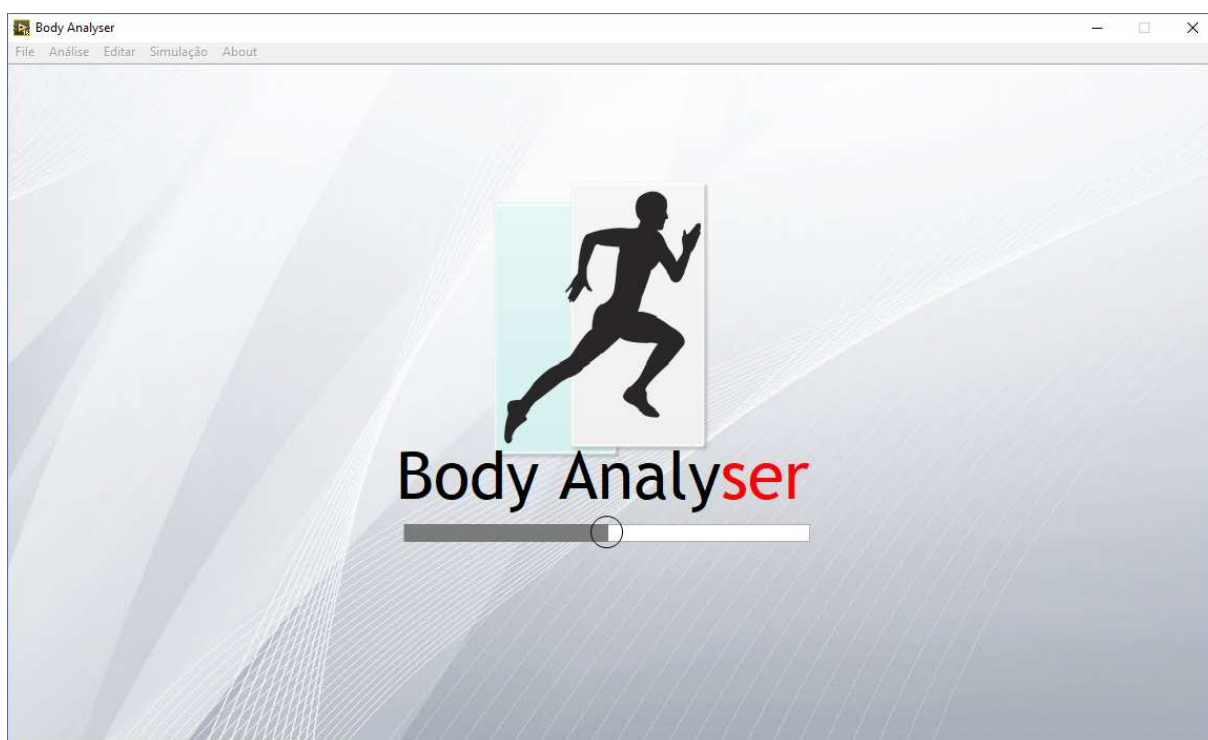


Figura 5 – Tela de inicialização do Body Analyser

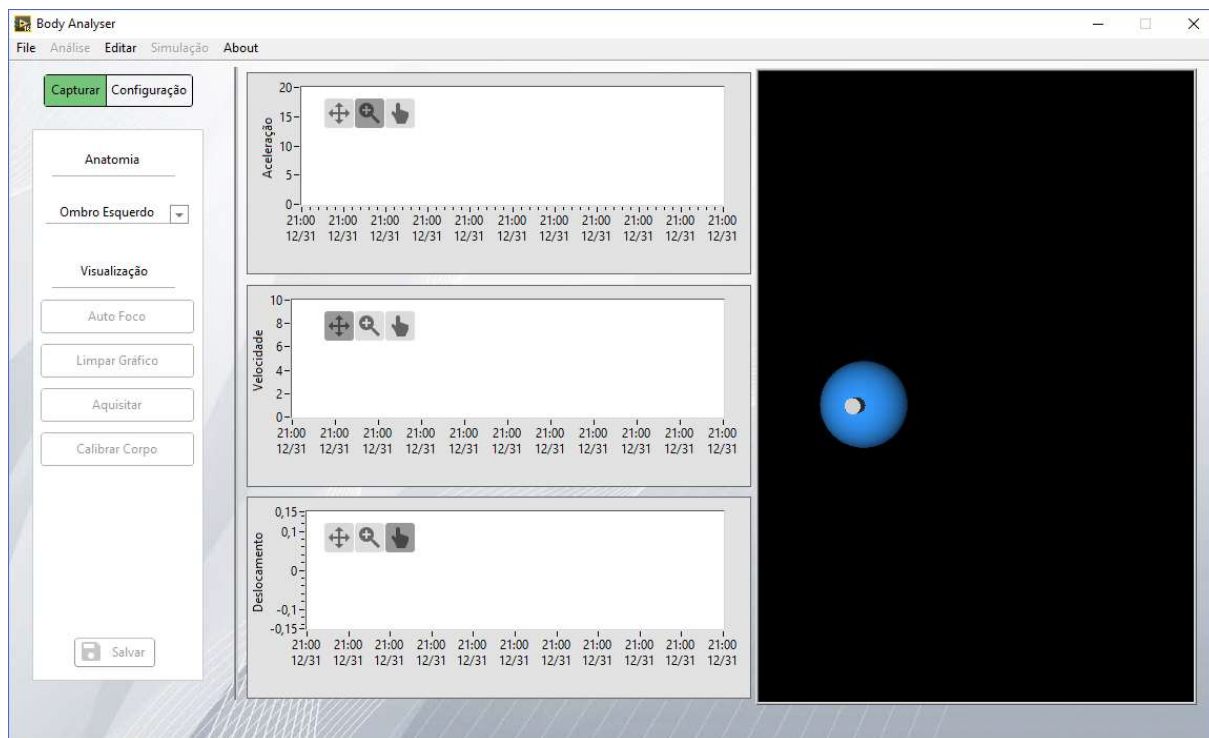


Figura 6 – Tela de Aquisição de dados

4.2 Fluxo de dados

A estrutura do software *Body Analyser* foi desenvolvida para otimizar recursos de processamento e preparada para aquisição de dados em tempo real. Onde dois While Loops revezam o trabalho entre cuidar dos eventos gerados pelo usuário e o segundo em realizar as ações, todo o fluxo é mantido por FIFO para que todos os frames sejam aquisitados sem perder nenhum dado e ao mesmo tempo fornecer ao usuário conforto em navegar na plataforma sem travar.

O Software é capaz de exportar os dados coletados para Excel e verificar se o hardware Kinect foi conectado por meio de chamada a DLL (*Dynamic Link Library*)

4.3 Controles do Kinect

Para realizar varredura de esqueleto humano é usado um sistema de elevação do hardware para variar o campo de visão. Em um sistema de malha fechada ao detectar que existe esqueleto o mesmo permanece na posição aquisitando dados, em contrapartida quando não existir ele avançará em degraus de 5° até +25° e irá até -25°. Veja a curva de varredura no gráfico abaixo:

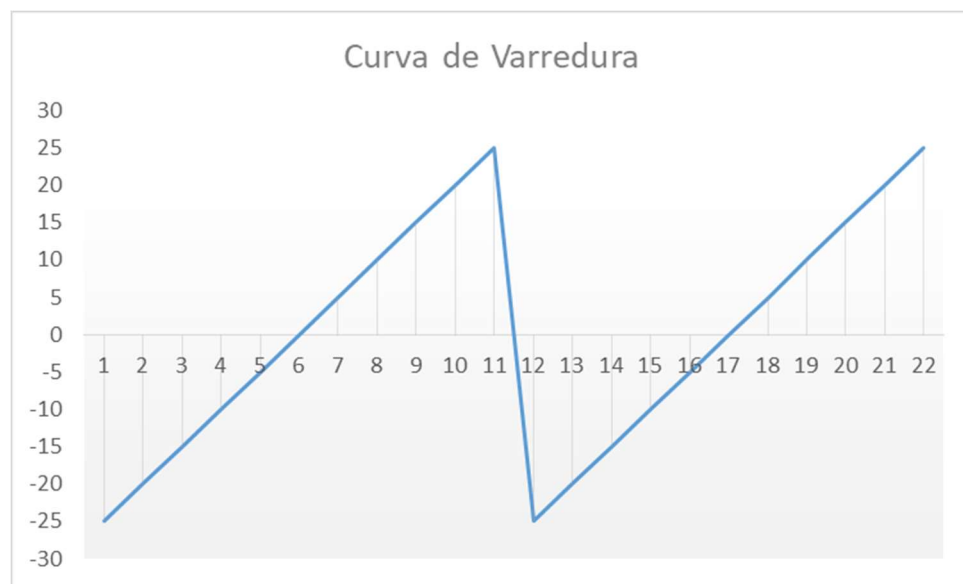


Gráfico 1 – Curva de elevação do Hardware

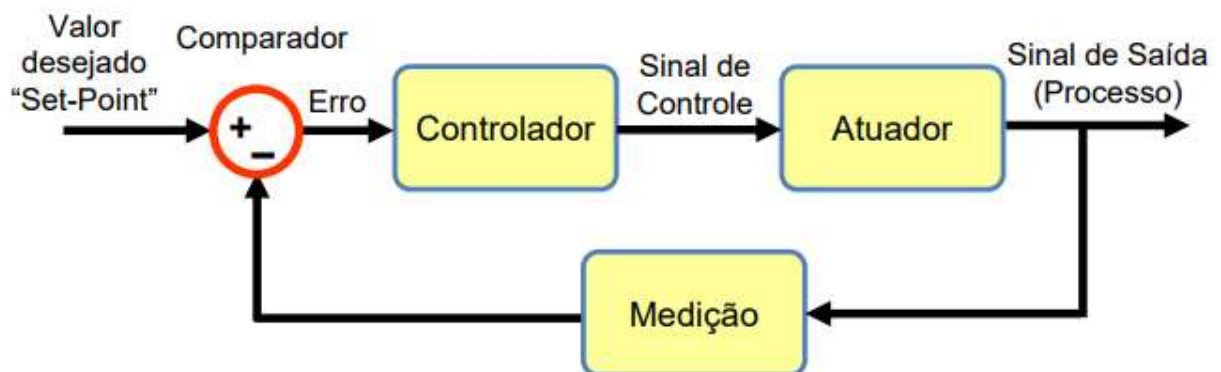


Figura 7 – Sistema de Malha fechada (Maria, Universidade Federal de Santa, 23)

4.4 Calibração com Acelerômetro Externo

O Software permite ao usuário calibrar o sistema por meio de um dado de aceleração externa, é realizada uma comparação entre o deslocamento gerado por um membro coletado pelo Kinect e a Integral da aceleração gerada pelo sensor externo. O deslocamento gerado por ambos os sistemas são comparadas para ajuste da curva do Kinect. Para o atual projeto foi usado um acelerômetro com resolução de 12bits mais software que permite conexões com sensores via bluetooth onde os usuários possam usar sensores com maiores resoluções para fazer o ajuste da curva do Kinect.

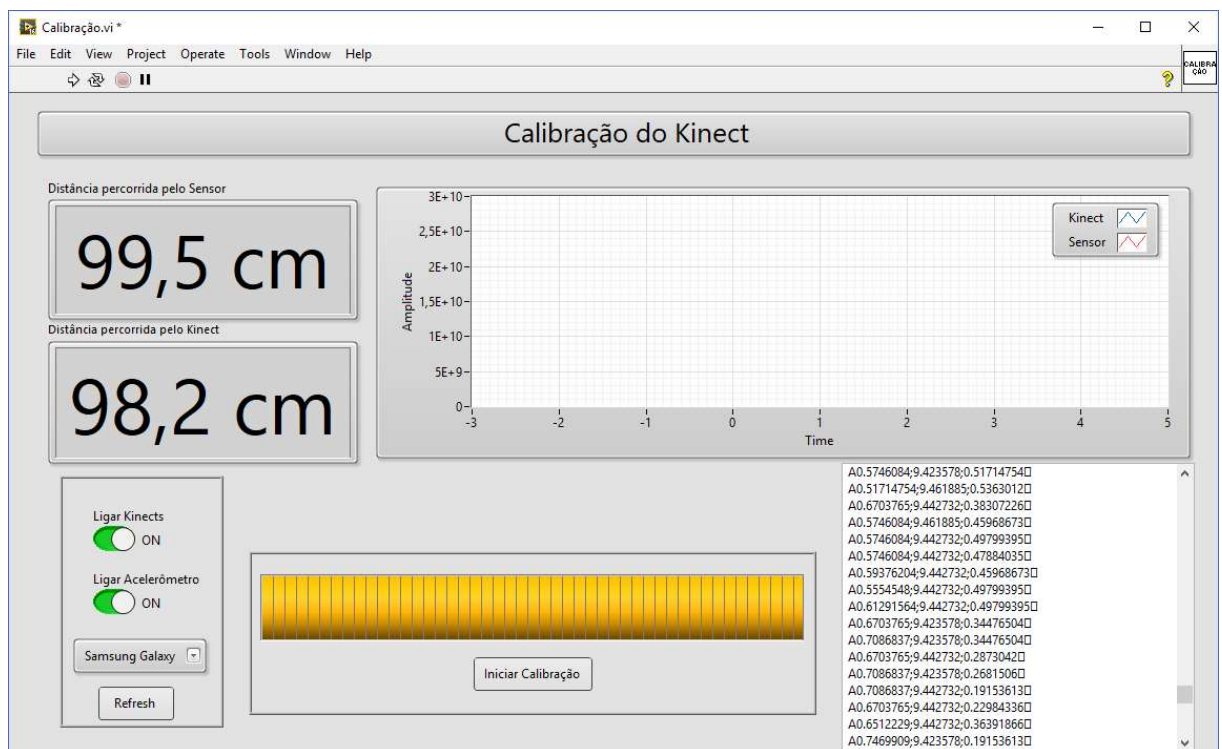


Figura 8 – Tela de Calibração do Kinect (distância em cm)

4.5 Calibração por meio de um corpo Humano

Para o software *Body Analyser* calcular em tempo real o deslocamento, velocidade e a aceleração dos movimentos com maior precisão é necessário que a pessoa que será analisada fique na posição vertical com os braços próximos ao corpo e execute a função calibrar corpo. O mesmo fará uma média com 1.700 imagens do corpo para obter distâncias dos membros que servirão como base de comparação durante a análise. Os dados coletados devem estar em uma margem de confiança de 95% do para que seja aceito e entre para o histórico.

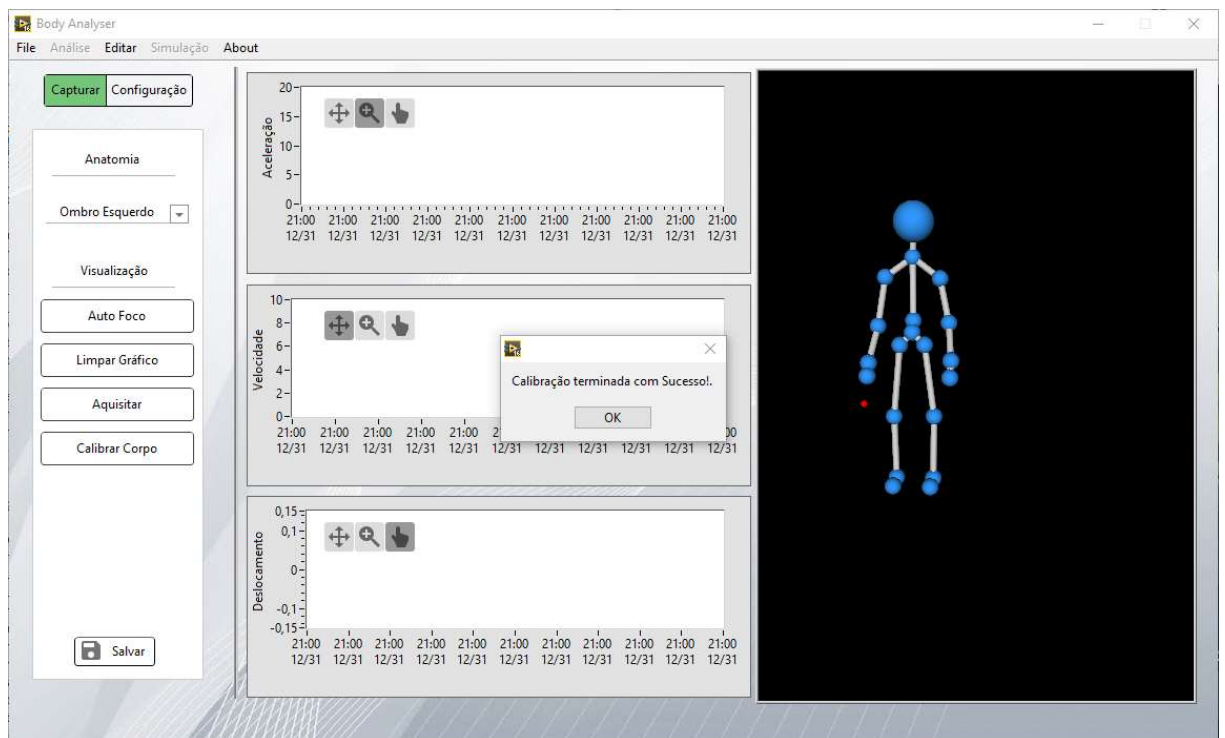


Figura 9 – Tela de Calibração do Corpo

4.6 Microsoft SDK V1.8

Para realizarmos a coleta de dados do Kinect é realizada uma chamada a DLL Kinect para gerar eventos de novos dados e usar os dados históricos para análise do movimento. A biblioteca usada neste projeto para aquisição de dados foi a Kinesthesia desenvolvida pela Universidade de Leeds departamento de Engenharia Mecânica.

4.7 Anatomia

O Body analyser permite ao usuário selecionar entre 20 articulações do corpo humano para calcular e mostrar em tempo real os dados.

4.8 Análise de Caso

Para tal análise vamos usar como estudo o joelho esquerdo onde o analisado teve uma fratura no membro com limitação na abertura da perna. Como parâmetro será usado o membro oposto.

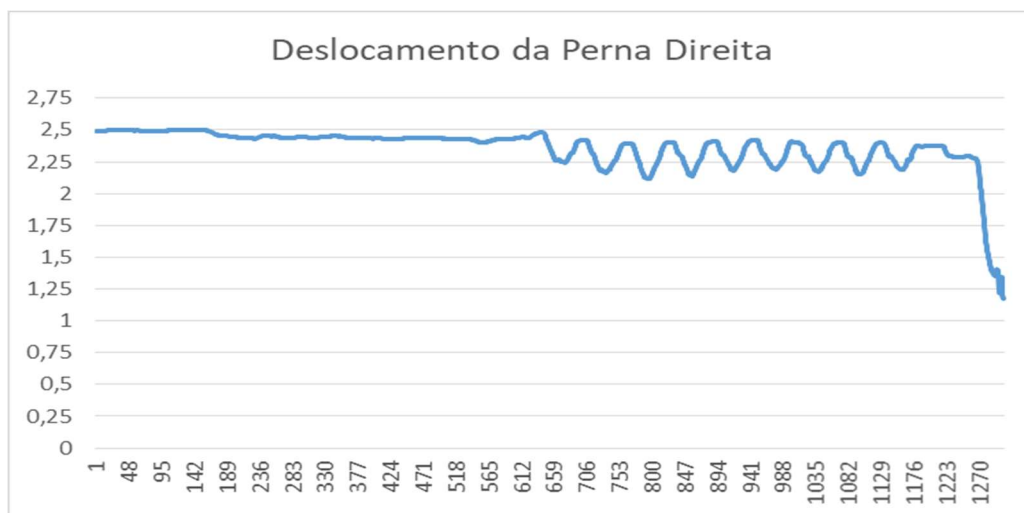


Gráfico 2 - Deslocamento máximo de 23 cm fechamento máximo de 90°.Eixo x número de amostras e eixo y distância percorrida em Cm.

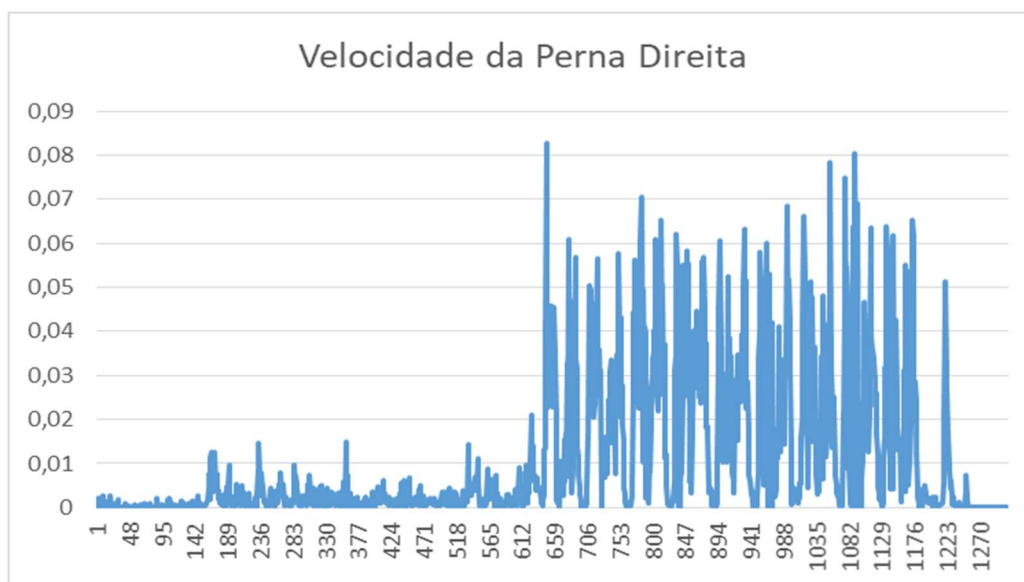


Gráfico 3 – Curva de Velocidade em cm/s da perna direita

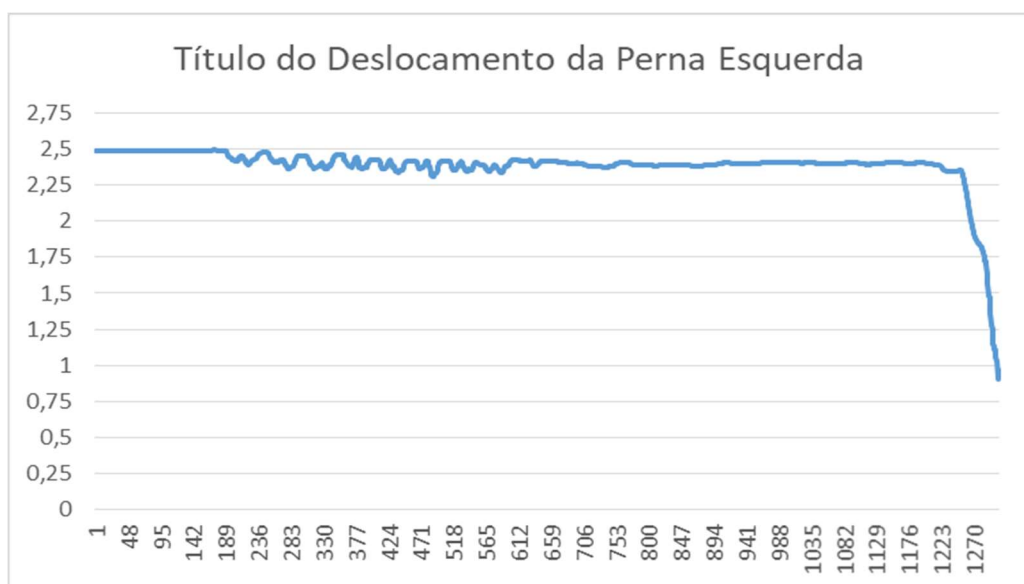


Gráfico 4- Deslocamento máximo de 9 cm fechamento máximo de 135°. Eixo x número de amostras e eixo y distância percorrida em Cm.

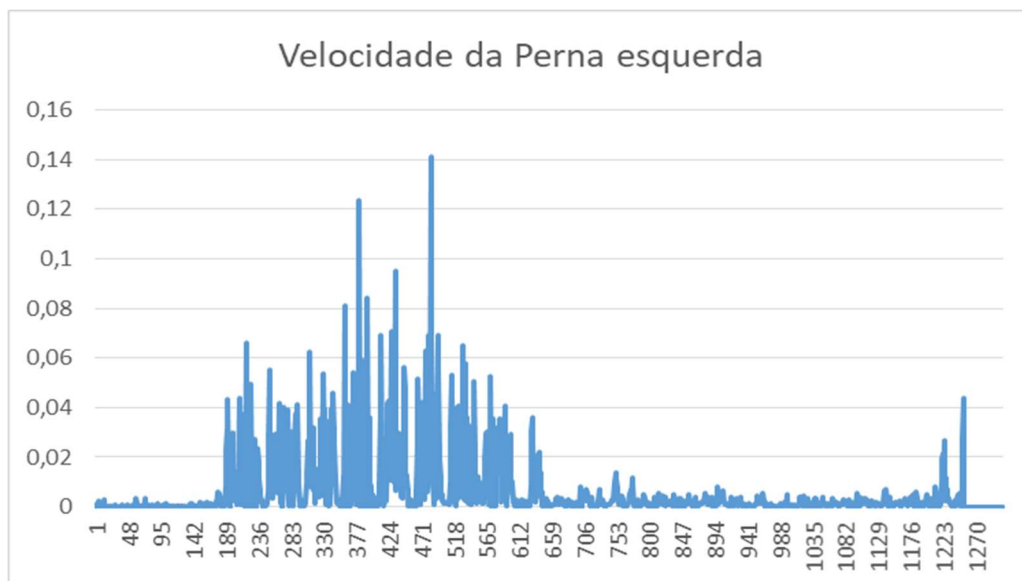


Gráfico 5 – Curva de Velocidade em cm/s da perna esquerda

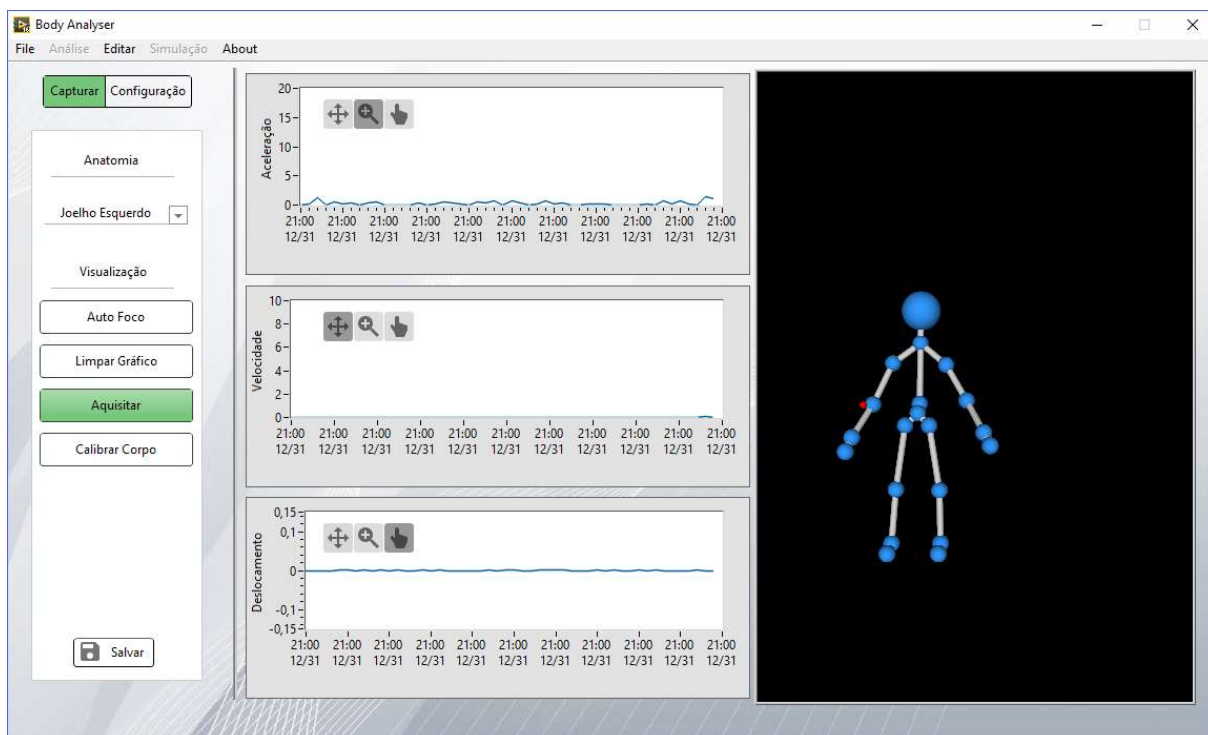


Figura 10 – Tela de aquisição de dados em tempo real

4.9 Análise do estudo de Caso

Com base nos dados obtidos usando o software Body Analyser foram realizados alguns tratamentos de dado para extrair informações dos dados brutos.

Os gráficos de deslocamento nos mostram a amplitude do movimento ao compararmos os dois percebemos de forma gráfica que a perna direita teve um deslocamento maior em uma seção repetida de movimento. Percebemos também que a perna esquerda teve uma abertura muito inferior comparada com o membro oposto mas desenvolveu uma velocidade equiparada e em alguns momentos superior a perna oposta. Podemos então avaliar apenas de forma analítica que a limitação do membro consiste apenas em ângulos inferiores a 135° e que ao realizar movimentos entre 135° e 180° a abertura equipara com a perna boa.

Diversos tratamentos de dados poderiam ser realizados para avaliar o movimento dentre eles se destacam:

1. Análise do deslocamento do membro
2. Análise da abertura angular do membro
3. Cálculo de distância dos membros, como por exemplo encontrar o comprimento da canela
4. Estudo da explosão do soco de um lutador

4.10 Custo final do Projeto

Item	Descrição	Quantidade	Valor Unitário	Valor Final
1	Kinect 360	1	R\$ 250,00	R\$ 250,00
2	Tripé 70 cm para Kinect	1	R\$ 50,00	R\$ 50,00
Valor Total				R\$ <u>300,00</u>

Tabela 2 – Descrição dos custos do projeto

5. Discussão e Conclusão

O objetivo principal do projeto é estudar com precisão os movimentos humanos, com os experimentos mostrados nos capítulos anteriores foram possíveis coletar com sucesso e tratar os dados para gerar informações confiáveis.

O estudo da análise de caso nos mostrou que um simples tratamento de dados é suficiente para extrair informações cruciais sobre o movimento.

O projeto desenvolvido nesse curso teve como necessidade chegar a um custo baixíssimo para viabilizar a ideia, pois a grande maioria dos entrevistados na área de fisioterapia destacou pontos importantes sobre os equipamentos tais como, peças de reposição são caras e a grande maioria importadas. O estudo não vem de forma alguma dar alta para um paciente, mas sim ajudar o especialista a entender a funcionalidade do tratamento ou o avanço que o mesmo está gerando.

Todos os movimentos devem ser obrigatoriamente coletados face a face (Sensor paralelo ao analisado), senso possível no Software *Body Analyser* rotacionar o avatar.

6. Referência Bibliográfica

- Affonso, P. H. (2014). Sistema de Controle de atitude de baixo custo usando rodas de reação. São Carlos, São Paulo, Brasil.
- Ahmed, F., Paul, P. P., & Gavrilova, M. L. (21 de 05 de 2017). Kinect-Based Gait Recognition Using Sequence of the Most Relevant Joint Relative Angles. Calgary, Canada. Fonte: https://www.researchgate.net/publication/282503184_Kinect-Based_Gait_Recognition_Using_Sequence_of_the_Most_Relevant_Joint_Relative_Angles?_sg=hqOGipeUxR3iOIYmL03zWbJcyi_RLT0H4HQTEAXz2rFORNcS-O1lhiclbqDYMK8Ggh3b-hu7gsIQOjhScfUvTg
- Amarino. (23 de 10 de 2017). *Amarino*. Fonte: Amarino: <http://www.amarino-toolkit.net/>
- Borentein, G. (2012). *Making Things See*. Canada: O'Reilly Media.
- Bosch. (1 de 08 de 2014). *Data Sheet BMA255*. Fonte: Mouser do Brasil: http://www.mouser.com/ds/2/783/BST-BMA255-DS004-05_published-786492.pdf
- Faria, D. (Junho de 2014). *Ánalyse e Processamento de Imagem*. Porto, Portugal.
- Instruments, N. (21 de 05 de 2017). <http://www.ni.com>. Fonte: <http://www.ni.com/newsletter/51141/pt/>
- MacCormik, J. (21 de 05 de 2017). *How does Kinect Work?* Fonte: <http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>
- Maria, Universidade Federal de Santa. (2017 de 10 de 23). *Automação Industrial*. Fonte: Universidade Federal de Santa Maria: http://coral.ufsm.br/beltrame/arquivos/disciplinas/medio_automacao_industrial/Aula03_Controlo_Automatico02.pdf
- Microsoft. (21 de 05 de 2017). <https://www.microsoft.com>. Fonte: <https://www.microsoft.com/en-us/download/details.aspx?id=44561>
- Tannenbaum, B. (21 de 05 de 2017). <https://www.mathworks.com>. Fonte: <https://www.mathworks.com/videos/using-microsoft-kinect-with-matlab-and-simulink-92855.html>
- Venturi, J. J. (2015). *Álgebra Vetoria e Geometria análítica*. Curitiba: Livrarias Curitiba.
- Vieira, L. P., & Aguiar, C. E. (Outubro de 2013). Experimentos com acelerômetros de Tablets e Smartphone. Rio de Janeiro, Universidade Federal do Rio de Janeiro, Brasil.

Apêndice A: Script Principal Matlab Coleta de dados do Kinect

```
%Monitoramento dos Movimentos
%Sensor Kinect v1.0
%Script MatLab coletar os dados do kinect
%TCC 2017

clc;

%obtem dados do hardware conectado
hwInfo = imaqhwinfo('kinect');
hwInfo.DeviceInfo(1);
hwInfo.DeviceInfo(2);
colorVid = videoinput('kinect',1);
depthVid = videoinput('kinect',2);
%Configura o modo do trigger
triggerconfig([colorVid depthVid],'immediate');
colorVid.FramesPerTrigger = 1;
depthVid.FramesPerTrigger = 1;

%inicia captura
start([colorVid depthVid]);

[colorFrameData,colorTimeData,colorMetaData] = getdata(colorVid);
[depthFrameData,depthTimeData,depthMetaData] = getdata(depthVid);

%Para Captura
stop([colorVid depthVid]);
depthSrc = getselectedsource(depthVid);
```



```

%Configura a quantidade de amostras a serem coletadas (Quantidade de
%Frames)
depthSrc.TrackingMode = 'Skeleton';
colorVid.FramesPerTrigger = 300;
depthVid.FramesPerTrigger = 300;

%Inicia coleta das amostras pré configuradas
start([colorVid depthVid]);

%Abre Preview das imagen em tempo real
preview(colorVid);

%Construção da matriz com os dados coletados
[frameDataColor] = getdata(colorVid);
[frameDataDepth, timeDataDepth, metaDataDepth] = getdata(depthVid);
metaDataDepth;
anyPositionsTracked = any(metaDataDepth(95).IsPositionTracked ~= 0);
anySkeletonsTracked = any(metaDataDepth(95).IsSkeletonTracked ~= 0);
trackedSkeletons = find(metaDataDepth(95).IsSkeletonTracked);
jointCoordinates = metaDataDepth(95).JointWorldCoordinates(:, :,
trackedSkeletons);
jointIndices = metaDataDepth(95).JointImageIndices(:, :, trackedSkeletons);
image = frameDataColor(:, :, :, 95);

nSkeleton = length(trackedSkeletons);

util_skeletonViewer(jointIndices, image, nSkeleton);

% Para a coleta dos dados
stoppreview([colorVid depthVid]);
%Limpa memória
flushdata([colorVid depthVid]);

```

```
delete([colorVid depthVid]);  
clear colorVid depthVid;
```

Apêndice B : Script de tratamento dos dados Matlab

```
%Monitoramento dos Movimentos  
%Sensor Kinect v1.0  
%Script MatLab para avaliar os moviementos  
%TCC 2017  
  
clc ; % Limpa Command Window  
velocidade=0;  
aceleracao=0;  
load skeleton.mat % Carrega Matriz com os dados coletados do Kinect via  
SkeletonViewer.m  
  
% Encontra tamanho da matriz e limpa gráfico  
for limpa=1:length(metaDataDepth)  
    teste=metaDataDepth(limpa).JointWorldCoordinates(1);  
    if teste== 0  
  
        else  
            break;  
        end  
    end  
end  
  
% Faz varredura na matriz e encontra um esqueleto válido  
for c=1:length(metaDataDepth)  
  
    if metaDataDepth(c).JointWorldCoordinates(1) == 0
```

```

else

% Dividi a matriz em pequenos arrays com os dados espaciais dos membros
a=metaDataDepth(c).JointWorldCoordinates(1,1:3); %Centro do quadril
b=metaDataDepth(c).JointWorldCoordinates(2,1:3); %Espinha
z=metaDataDepth(c).JointWorldCoordinates(3,1:3); %Centro do Ombro
d=metaDataDepth(c).JointWorldCoordinates(4,1:3); %Cabeça
e=metaDataDepth(c).JointWorldCoordinates(5,1:3); %Ombro esquerdo
f=metaDataDepth(c).JointWorldCoordinates(6,1:3); %Cotovelo Esquerda
g=metaDataDepth(c).JointWorldCoordinates(7,1:3); %Pulso Esquerdo
h=metaDataDepth(c).JointWorldCoordinates(8,1:3); %Mão Esquerda
i=metaDataDepth(c).JointWorldCoordinates(9,1:3); %Ombro Direito
j=metaDataDepth(c).JointWorldCoordinates(10,1:3); %Cotovelo Direito
k=metaDataDepth(c).JointWorldCoordinates(11,1:3); %Pulso Direito
l=metaDataDepth(c).JointWorldCoordinates(12,1:3); %Mão Direita
m=metaDataDepth(c).JointWorldCoordinates(13,1:3); %Quadril esquerdo
n=metaDataDepth(c).JointWorldCoordinates(14,1:3); %Joelho Esquerdo
o=metaDataDepth(c).JointWorldCoordinates(15,1:3); %Tornozelo Esquerdo
p=metaDataDepth(c).JointWorldCoordinates(16,1:3); %Pé esquerdo
q=metaDataDepth(c).JointWorldCoordinates(17,1:3); %Quadril Direita
r=metaDataDepth(c).JointWorldCoordinates(18,1:3); %Joelho Direito
s=metaDataDepth(c).JointWorldCoordinates(19,1:3); %tornozelo Direito
t=metaDataDepth(c).JointWorldCoordinates(20,1:3); %Pé Direito


subplot(1,2,1);

curve = animatedline('LineWidth',2);
set(gca,'Xlim',[-1 1],'Ylim',[-1 1],'Zlim',[0 5]);
grid off;
view(0,90);

```

```

hold on;

cla;

scatter3(a(1),a(2),a(3),300,20,'filled'); %Centro do quadril
scatter3(b(1),b(2),b(3),300,20,'filled'); %Espinha
scatter3(z(1),z(2),z(3),900,20,'filled'); %Centro do Ombro
scatter3(d(1),e(2),d(3),300,20,'filled'); %Cabeça
scatter3(e(1),e(2),e(3),300,20,'filled'); % Ombro Esquerdo
scatter3(f(1),f(2),f(3),300,20,'filled'); % Cotovelo Esquerdo
scatter3(g(1),g(2),g(3),300,20,'filled'); % Pulso esquerdo
scatter3(h(1),h(2),h(3),300,20,'filled'); %Mão Esquerda
scatter3(i(1),i(2),i(3),300,20,'filled'); % Ombro Direto
scatter3(j(1),j(2),j(3),300,20,'filled'); % Cotovelo direito
scatter3(k(1),k(2),k(3),300,20,'filled'); % Pulso direito
scatter3(l(1),l(2),l(3),300,20,'filled'); %Mão Direita
scatter3(m(1),m(2),m(3),300,20,'filled'); %Quadril esquerdo
scatter3(n(1),n(2),n(3),300,20,'filled'); %Joelho Esquerdo
scatter3(o(1),o(2),o(3),300,20,'filled'); %Tornozelo Esquerdo
scatter3(p(1),p(2),p(3),300,20,'filled'); %Pé esquerdo
scatter3(q(1),q(2),q(3),300,20,'filled'); %Quadril Direita
scatter3(r(1),r(2),e(3),300,20,'filled'); %Joelho Direito
scatter3(s(1),s(2),s(3),300,20,'filled'); %tornozelo Direito
scatter3(t(1),t(2),t(3),300,20,'filled'); %Pé Direito
title('Movimento do Corpo Humano');
drawnow limitrate; % imprime na tela em tempo real a atualizaçao dos dados
hold off;

subplot(4,2,2); % Gráfico 1
set(gca,'Xlim',[0 length(metaDataDepth)],'Ylim',[0 1]);
hold on;

```

```

grid on;
if c == limpa
    cla;
else
end
if velocidade == 0
    velanterior=0;
    deslocaanterior=0;
else
    velanterior=velocidade;
    deslocaanterior=deslocamento;
end
%Cálcula o deta de tempo , Tempo= Tempo atual - Tempo Anterior
tempodif=(metaDataDepth(c).AbsTime - metaDataDepth(c-1).AbsTime);
tempofinal=(tempodif(5)+tempodif(6));
dados=metaDataDepth(c-1).JointWorldCoordinates(11,1:3);
%Cálcula o delta da posição, Posição=posição atual - posição anterior
deslocamentoanterior=sqrt((dados(1)^2)+(dados(2)^2)+(dados(3)^2));
deslocamentoatual= sqrt((k(1)^2)+(k(2)^2)+(k(3)^2));
deslocamento=abs(deslocamentoatual-deslocamentoanterior);
%Cálcula a velocidade , Velocidade = Delta posição/ delta tempo
velocidade=(deslocamento/tempofinal) % plota a velocidade no gráfico
if velocidade > 10
    velocidade=0;
end
if deslocamento > 0.1
    deslocamento=0;
end
scatter(c,velocidade);
line([c-1 c],[velanterior velocidade]);

```

```
title('Velocidade');  
hold off;  
drawnow limitrate;
```

```
subplot(4,2,4); % Gráfico 2
```

```
set(gca,'Xlim',[0 length(metaDataDepth)],'Ylim',[0 10],'Zlim',[0 3]);  
grid on;  
hold on;  
view(0,90);  
if c == limpa  
    cla;  
else  
end  
if aceleracao == 0  
  
    acelanterior=0;  
else  
    acelanterior=aceleracao;  
  
end  
%calcula a aceleração , Aceleração = delta Velocidade/delta tempo  
aceleracao = (velocidade/tempofinal) ;  
if aceleracao > 20;  
    aceleracao=0;  
end  
scatter(c,aceleracao); % plota a aceleração no gráfico
```

```
line([c-1 c],[acelanterior aceleracao]);  
title('Aceleração');
```

```
drawnow limitrate;  
hold off;
```

```
subplot(4,2,6); % Gráfico 3
```

```
if c == limpa
```

```
    cla;
```

```
else
```

```
end
```

```
hold on;
```

```
grid on;
```

```
set(gca,'ylim',[0 0.01],'Xlim',[0 length(metaDataDepth)]);
```

```
scatter(c,deslocamento); % Plota deslocamento no gráfico
```

```
line([c-1 c],[deslocaanterior deslocamento]);
```

```
title('Deslocamento');
```

```
hold off;
```

```
subplot(4,2,8);
```

```
if c == limpa
```

```
    cla;
```

```
else
```

```
end
```

```
grid on;
```

```
hold on;
```

```
view(42,20);
```

```
set(gca,'Xlim',[-1 1],'Ylim',[-1 1],'Zlim',[0 4]);
```

Apêndice C : Acesso

Para obter uma cópia completa do projeto desenvolvido em labview, acesso o link abaixo:

