

ESTUDO DE UMA FERRAMENTA TRADUTORA DA LINGUAGEM DE SINAIS (LIBRAS) VISANDO UMA COMUNICAÇÃO MAIS AMPLA DO DEFICIENTE AUDITIVO

Stéfany Mazon¹; Alessandra Dutra Coelho²

¹Aluna de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

²Professora da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *A informação e o conhecimento são dois dos principais meios que movem o mundo. Este mundo, banhado pela tecnologia, proporciona incontáveis facilidades e aprimoramentos na vida da população, a qual lida a cada dia com a necessidade de criação de novas tecnologias para pessoas com necessidades especiais.*

Este trabalho apresenta um estudo para desenvolvimento de um software de análise, transcrição e tradução dos símbolos presentes na linguagem brasileira de sinais (LIBRAS) de forma automatizada. O grande diferencial deste trabalho é o uso do sensor Microsoft Kinect for Windows como ferramenta para captura de imagem. Esse sistema pode auxiliar em incontáveis situações de interação entre pessoas com dificuldade de audição e fala e pessoas que não entendem a linguagem dos sinais, como em consultas médicas ou em salas de aula.

Introdução

A cada dia que passa, a população mundial preocupa-se com a acessibilidade, seja ela no caso de deficientes auditivos, físicos, visuais, entre outros. No caso dos deficientes auditivos, a Língua de sinais surgiu por volta do século XVI, na Europa. Essa língua tem como objetivo desenvolver a comunicação dos mesmos, que até então não tinham como se comunicar com o resto da população, sendo excluídos da sociedade.

Atualmente cada país possui uma língua de sinais particular, com movimentos significativos para cada expressão, ou seja, essa linguagem não é universal, nem padronizada, cada palavra pode ser interpretada de forma independente variando conforme o país (PACHECO, ESTRUC e ESTRUC, 2008).

No Brasil a língua de sinais teve seu início por volta de 1856 através do Professor francês Ernest Huet, sendo portador de deficiência auditiva, trouxe ao Brasil o alfabeto gestual Francês, e através desse alfabeto começou a surgir a Língua Brasileira de Sinais (Figura 1), até chegar ao que é atualmente, conhecida como LIBRAS (RAMOS, 2006).



Figura 1 - Alfabeto em LIBRAS (Língua de Sinais Brasileira).

Fonte: ARAUJO (2007)

Com o intuito de facilitar ainda mais a comunicação dos deficientes auditivos com o resto da população pensou-se nesse trabalho, que visa à criação de um software de tradução de Libras, isto é, o usuário poderá fazer um movimento em frente à câmera do Microsoft Kinect for Windows, e a tradução do sinal aparecerá na tela do computador como uma legenda.

Para início de projeto, foi feito um curso semestral de LIBRAS, de modo a obter-se mais informações e conhecimento a respeito dessa língua.

Material e Métodos

Microsoft Kinect

“Project Natal” como foi nomeado durante seu desenvolvimento, foi um projeto da Microsoft para criar uma nova forma de Interface Homem Máquina (IHM) voltada para o ambiente de controle em videogames. Lançado em 2010 o Kinect para Xbox 360 é uma plataforma na qual se pode interagir com jogos “sendo” o controle. O ideal por trás do lançamento foi incorporar um ambiente de imersão que competisse com seu rival direto Nintendo U da Nintendo, onde movimentos do usuário são capturados por um controle equipado com sensores de movimento.

Logo que as vendas foram iniciadas, desenvolvedores foram compelidos a trabalhar com o novo dispositivo que mostrou capacidades para atender diversas aplicações.

Pacotes de software foram desenvolvidos por entusiastas em forma de código aberto em tal volume e rapidez, que a Microsoft anunciou que desenvolveria um Standard Developer Kit (SDK) para a comunicação entre o sensor e o Windows. Em 2012 foi lançado o Kinect for Windows, uma versão com o mesmo hardware mas com diferenças significativas na liberdade a qual os programadores podem gozar ao interagir com o Kinect. Funcionalidades como o controle do estado do sensor infravermelho e a possibilidade de trabalhar com mais de um dispositivo conectado ao mesmo computador fazem parte das vantagens do Kinect for Windows. Além de ser um instrumento ideal para analisar espaços tridimensionais, o Kinect foi criado com um objetivo muito específico que o faz ser a escolha certa para a aplicação em questão. O Kinect foi concebido para identificar o corpo humano e gestos que podem ser replicados pelo mesmo. Para entender como isso se conecta à decisão de utilização deste dispositivo vale uma breve explicação de seu funcionamento a respeito de seu Hardware e Software.

A respeito do hardware pode-se citar que o aparelho é composto por uma série de sensores que trabalham juntos na coleta de dados para o desenvolvimento de informação a qual será traduzida (para a maioria das aplicações) em objetos localizados em um espaço tridimensional. O Kinect (Figura 2) conta com uma câmera RGB de 640x480 de resolução que captura imagens a 30Hz, um sensor infravermelho composto por um laser e receptor de mesma resolução da câmera de vídeo e um microfone multivetorial composto por quatro microfones individuais. Os sensores combinados tem uma área funcional determinada pela posição do usuário, que se estende de 40cm a 3m de distância do sensor em uma linha reta a partir do seu centro e um ângulo de 45° para cada lado partindo do mesmo referencial.



Figura 2 - Indicação do posicionamento dos sensores do Kinect.

Os microfones são utilizados na captação de som e identificação de sua origem medindo-se o tempo que uma onda leva para alcançar o referencial. Para detectar o ambiente a sua volta o sensor usa um laser infravermelho que projeta um padrão de pontos. O padrão resultante é então recebido por um receptor de infravermelho o qual está calibrado para receber a mesma frequência emitida. A construção do espaço é feita com base no mapa de profundidade criado a partir das variações encontradas no padrão de pontos que retorna, comparado com o padrão emitido.

Já a respeito do software pode-se citar a identificação de objetos em profundidade e a análise do padrão de pontos. O sistema de tratamento de dados feitos pelo próprio hardware do Kinect é baseado numa tecnologia da PrimeSense licenciada pela Microsoft. O princípio utilizado para a criação de um mapa de profundidade é simples, porém extremamente eficaz.

De acordo com a forma onde o padrão de pontos irá incidir e então retornar ao receptor, a geometria individual de cada ponto e as mudanças nas distâncias entre cada ponto da malha gerada entre si, resulta em uma interpretação extremamente detalhada da cena analisada. Pode-se incluir esta técnica de leitura como “visão baseada em marcadores” com a vantagem de que os marcadores serão gerados pelo padrão infravermelho.

A partir deste ponto a interpretação do que está sendo visto terá início. Com uma coleção de dados recolhidos pela Microsoft, o mapa de profundidade é comparado com milhares de exemplos de corpo humano e assim a estrutura pode ser interpretada e reconhecida. O procedimento completo de reconhecimento não foi publicado por completo e por esse motivo não é possível afirmar que a câmera RGB não é utilizada no algoritmo.

Passando-se por todo o processo já descrito o Kinect disponibiliza além do mapa de profundidade da imagem e do áudio outra estrutura de dados utilizada especificamente para dar informações de posição das juntas e número de jogadores. É detectada a presença de até seis jogadores simultaneamente e podem-se obter informações das juntas individuais e de dois jogadores concomitantemente.

As informações de cada junta são passadas na forma de um vetor constando o index do jogador e todas as 20 juntas enumeradas com a posição em coordenadas x, y e z convertidas em milímetros. O referencial de cada dimensão medida varia da seguinte forma: o eixo x se refere à distância horizontal de certo ponto do centro da imagem analisada, pontos à direita têm valor positivo e à esquerda valores negativos; o eixo y tem o mesmo referencial medindo-se a altura do ponto sendo que abaixo do centro da imagem tem valor negativo e acima valor positivo; z mede a profundidade, deste modo, esta coordenada assumirá apenas valores positivos, sendo zero a posição do próprio sensor. Esta forma de referência é simples e independe do ambiente de trabalho, entretanto, dificulta a integração de mais de um dispositivo simultaneamente.

Para a utilização ampla do Kinect em diversas aplicações, estão disponibilizados na internet para download os softwares abaixo citados.

Kinect SDK

O Kinect SDK, conforme citado acima, faz a conexão entre o Kinect e o Windows, pegando as informações do Kinect e transportando-as para os drivers.

No caso desse projeto, o driver utilizado foi o CCT.NUI.

Candescent NUI (CCT.NUI)

O CCT.NUI (Figura 3) foi a grande base do projeto. Nele é realizada a junção entre a interface, middleware e os componentes gestuais, ou seja, a captura de movimentos das mãos e corpos.

Nesse software selecionou-se como framework o Kinect SDK, citado acima.

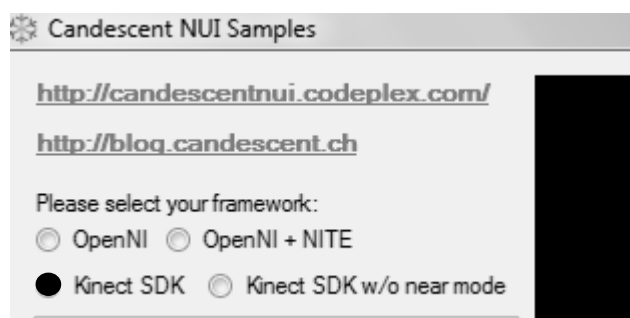


Figura 3 - Opções de frameworks a serem utilizados no CCT.NUI.

Após escolher o SDK, há algumas variedades de visualização de imagem que podem ser utilizadas no CCT.NUI, sendo elas a imagem RGB (Figura 4), imagem de profundidade (Figura 5), e a mais importante para esse trabalho, a imagem de detecção de mãos e dedos (Figura 6).

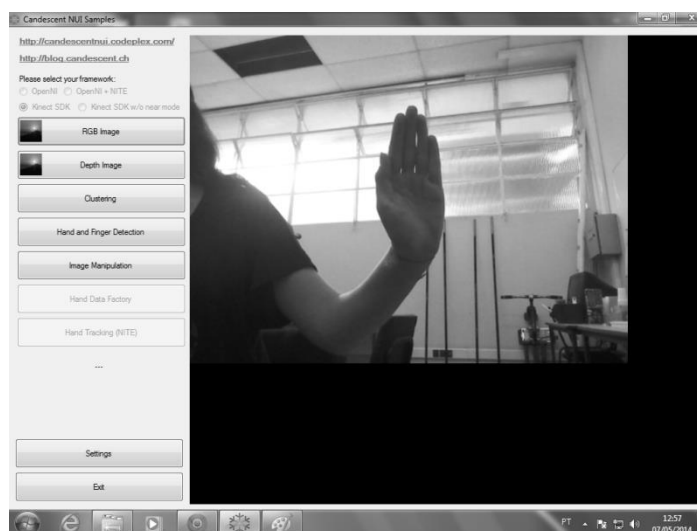


Figura 4 - Imagem RGB.

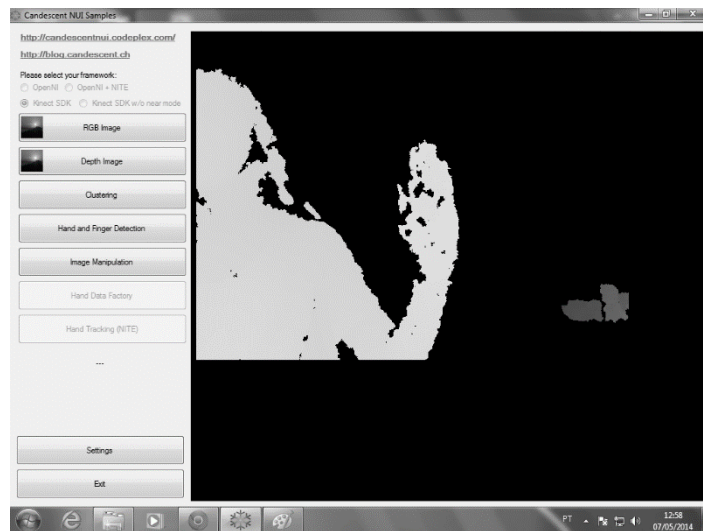


Figura 5 - Imagem em profundidade.

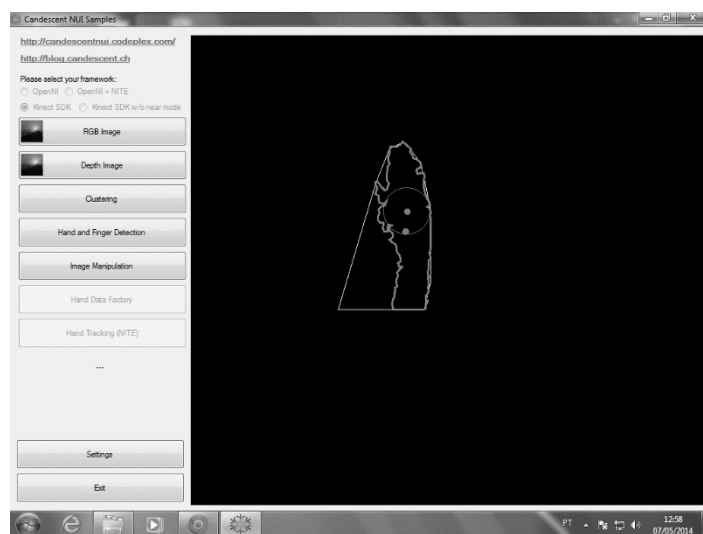


Figura 6 - Detecção de mãos e dedos.

Para essa última opção, deve-se notar que o software faz o contorno da mão, e encontra o seu centro de massa. Dois pontos extremamente importantes para a implementação desse projeto.

Kinect for Windows Developer Toolkit

O Kinect Developer Toolkit se trata de um software no qual há diversos exemplos de softwares que utilizam o Kinect, sendo eles em programação em C# ou C++. Alguns dos exemplos disponíveis foram base para a formulação desse projeto, como no caso do Skeleton Basics- D2D (Figura 7), o qual demonstra o esqueleto do usuário e os pontos de articulação.

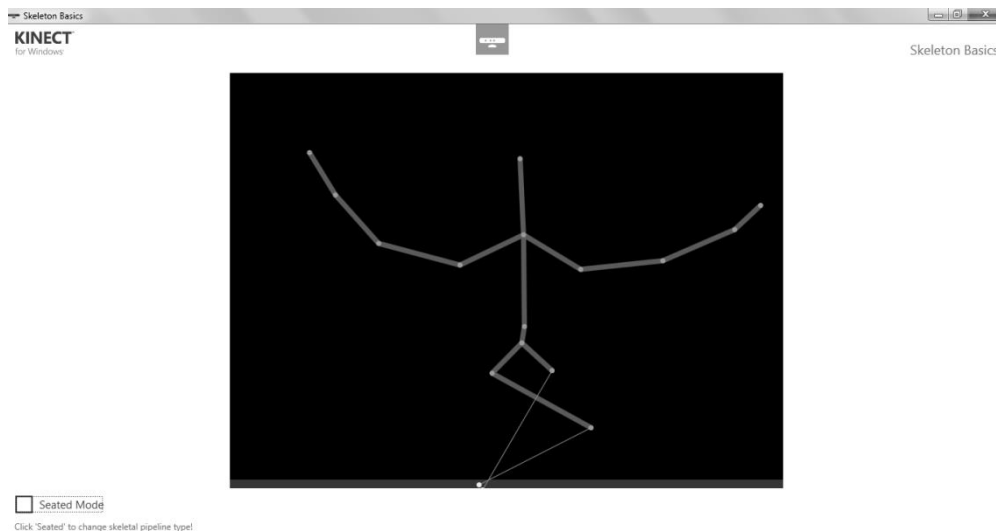


Figura 7- Skeleton Basics

Com a utilização dos programas acima citados, foi possível desenvolver o Software desse projeto que tem como objetivos:

- Dar a liberdade ao usuário para que ele faça o movimento de uma letra contida no alfabeto de LIBRAS;
- Fazer o reconhecimento do símbolo e transcrevê-lo na tela do computador.

Após o estudo de algumas plataformas de compilação em C++, foi escolhido para esse projeto o Visual Studio 2013, que por ser uma plataforma Microsoft facilita a interação código-hardware. Nesse caso, coloca-se como referência o Kinect, fazendo com que todas as operações feitas no código passem aos sensores e ao software interno.

O diagrama de atividades abaixo é responsável por representar os estados do código durante a sua execução.

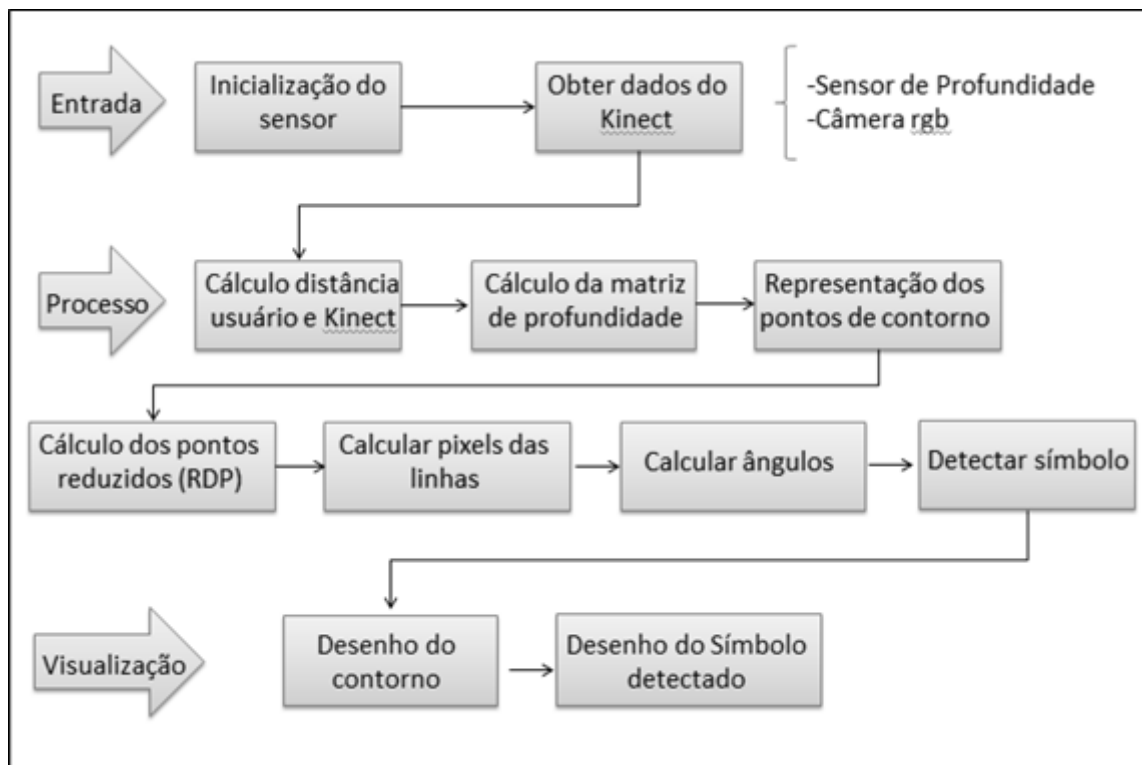


Figura 8 - diagrama de atividades.

Analisando a Figura 8 nota-se que inicialmente deve-se, por via de código inicializar o sensor, pois o Kinect possui mais de uma forma para utilização de suas câmeras. Após isso é válido citar alguns dos passos acima:

- Cálculo dos pontos reduzidos (Ramer- Douglas- Peucker): esse algoritmo trata-se de uma redução do número de pontos de uma curva. A forma inicial do algoritmo foi sugerida em 1972 por Urs Ramer em 1973 por David Douglas e Thomas Peucker e vários outros na década seguinte.

O funcionamento do algoritmo procede da seguinte forma: o primeiro ponto da linha é definido como a "âncora" e o último ponto como um "flutuador". Estes dois pontos estão ligados por um segmento de reta e distâncias perpendiculares, onde a partir deste segmento todos os pontos intermediários são calculados. Se nenhuma destas distâncias perpendiculares exceder uma tolerância especificada, o segmento de reta é considerado adequado para representar toda a linha de forma simplificada. Se esta condição não for satisfeita, então o ponto de maior distância do segmento de reta perpendicular é selecionado, tornando-se o novo ponto flutuante. Este processo continua até que os critérios de tolerância sejam atendidos, e deste modo cria-se a reta simplificada (WHYATT; WADE, 1988).

Na Figura 9 é possível visualizar um segmento de reta simplificado pelo algoritmo de Ramer-Douglas-Peucker.

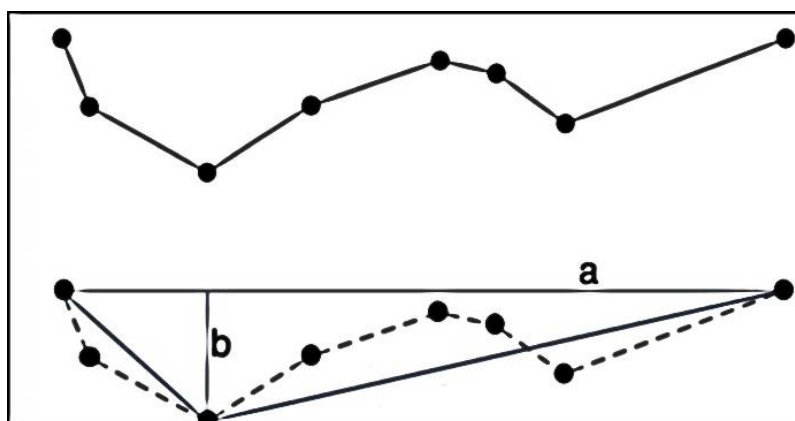


Figura 9 - Simplificação por Ramer-Douglas-Peucker
Fonte: Ramer (1972).

Para o projeto, foi utilizado esse algoritmo de modo a simplificar o contorno da mão do usuário para retas, o que facilita o cálculo dos ângulos entre as mesmas. Com a obtenção dos ângulos foi possível fazer a comparação com os dados salvos na extensão XML e deste modo, reconhecer o símbolo executado.

- Detectar símbolo: após todas as passagens demonstradas no diagrama há a detecção do símbolo desejado. É importante citar que para isso, utiliza-se, dentro da plataforma Visual Studio, a extensão Nuget, a qual dá acesso ao Libxml2. Essa é uma plataforma de XML na qual podem ser escritos os dados originais do símbolo que será comparado com o símbolo feito pelo usuário.

Resultados e Discussão

Para a implementação do projeto foram efetuados alguns testes. Primeiramente, deve-se citar que houveram alguns experimentos descartados como no caso da utilização do Framework OPENNI, o qual seria uma opção para o funcionamento do projeto, mas não

estava mais disponível para download. Além disso, notou-se que o CCT.NUI tinha um ótimo desempenho, não havendo a necessidade da procura de outro Framework.

Deve-se citar também a complexidade de programação com a plataforma XML. Mesmo com o estudo do mesmo, e implementação da extensão Nuget no Visual Studio, foi extremamente difícil manusear os dados de comparação de imagens. Não conseguiu-se obter os dados iniciais que seriam salvos nos arquivos XML e posteriormente utilizados como base para comparação.

Conclusões

Como proposto no trabalho inicial, esse projeto consistiu em um estudo de uma ferramenta de análise, transcrição e tradução dos símbolos presentes na linguagem brasileira de sinais (LIBRAS). Em função de o projeto ter uma alta complexidade, requerer amplo conhecimento de programação, e ter um tempo escasso para produção, a interface final de interação com o usuário não foi implementada, mas visa-se a implementação da mesma no futuro. Deste modo o reconhecimento dos símbolos ocorrerá de forma completa, pois até o momento, o símbolo é somente detectado pelo algoritmo, mas não é transcrito pela falta da interface.

No entanto, de modo geral os objetivos foram alcançados, com um amplo aprendizado na área de processamento de imagens e programação em linguagem de alto nível.

Deve-se citar que, a plataforma Kinect, escolhida para a implementação do projeto, é extremamente eficaz e facilita o reconhecimento de imagens, por ter inúmeros tutoriais e códigos abertos da própria Microsoft os quais facilitam o estudo da plataforma. No entanto o Kinect não é portátil restringindo a aplicação do projeto somente para lugares fixos, como no caso de locais de trabalho.

Para o futuro, espera-se uma evolução do projeto a fim de tornar a aplicação viável para smartphones e tablets, mas desde já se acredita que o projeto facilitará a inserção dos deficientes auditivos no contexto social, melhorando sua autoestima e facilitando a sua comunicação.

Referências Bibliográficas

Araújo, Ana P. Linguagem de Sinais Brasileiras (Libras); [S.l]. 2007. Disponível em: <<http://www.infoescola.com/portugues/lingua-brasileira-de-sinais-libras/>>. Acesso em: 11 de abril 2014.

Code Plex, Project Hosting for Open Source Software, Download do CCT.NUI. Disponível em: <candescentnui.codeplex.com>

Cuore, P. R. E. C. A Importância de Conhecer a Estrutura Linguística da Libras para o Educador. Campo Grande, 30 maio 2009. Disponível em: <<http://www.artigonal.com/educacao-artigos/a-importancia-de-conhecer-a-estruturalinguistica-da-libras-para-o-educador-945026.html>>. Acesso em: 20 de janeiro 2014.

Cursou, cursos online grátis, Curso de linguagem C# para iniciantes. Disponível em: <<http://www.cursou.com.br/informatica/curso-linguagem-c-iniciantes/>>

Dicionário de Libras. Dicionário de Libras, 2011. Disponível em: <<http://www.dicionariolibras.com.br>>. Acesso em: 20 de janeiro 2014.

Gonzales,Rafael C.(2011) Processamento Digital de Imagens, Pearson.
Microsoft, Kinect for Widows, Developing Kinect for Windows and real-world examples.
Disponível em: <www.microsoft.com/en-us/kinectforwindows/>

Microsoft, Kinect for Windows, Download de softwares. Disponível em:
<<http://www.microsoft.com/en-us/kinectforwindowsdev/Downloads.aspx>>

Oliveira, P. M. T. D. Sobre Surdos. Site de Jonas Pacheco, 2011. Disponível em:
<<http://www.surdo.org.br/informação.php?lg=pt&info=Historiadossurdos>>. Acesso em:
12 novembro 2013.

Pacheco, J.; ESTRUC, E.; ESTRUC, R. Curso Básico de LIBRAS. [S.l.]:
www.surdo.org.br, v. 8, 2008.

Ramer, Urs. An iterative procedure for the polygonal approximation of plane curves.
Computer Graphics and Image Processing. v. 1, p. 244-256. 1972.

Ramos, C. R. LIBRAS: A Língua de Sinais dos Surdos Brasileiros, Petrópolis, 2006.
Disponível em: <<http://www.editora-arara-azul.com.br/pdf/artigo2.pdf>>. Acesso em: 20
de janeiro 2014.

Santin, Diego M. (2013) Ferramenta Para Transcrição do Alfabeto Datilógico para texto
utilizando Microsoft Kinect.

Santos, Ana Paula T. et al , Kinlib- Protótipo de Ensino de Libras Utilizando o Kinect (2011).

SlideShare, Desenvolvendo aplicativos para o Kinect. Disponível em:
<<http://pt.slideshare.net/bcpbcp/developing-applications-for-kinect>>

Wash, R. PrimeSense Supplies 3-D-Sensing Technology to “Project Natal” for Xbox
360. Site da Microsoft Corporation, 2010. Disponível em:
<<http://www.microsoft.com/Presspass/press/2010/mar10/03-31PrimeSensePR.msp>>.
Acesso em: 23 de janeiro 2014.

Wikipédia, A Enciclopédia Livre(2014), Kinect. Disponível em:
<pt.wikipedia.org/wiki/Kinect>

Wikipédia, A Enciclopédia Livre(2014), Ramer Douglas Peucker algorithm. Disponível em:
<em.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm>