MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Visualização Interativa de Dados e Cenários Ambientais Usando Controle de Gestos

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA

(PIBIC/CNPq/INPE)

Heitor Guerra Carneiro (Faculdade de Tecnologia de São José dos Campos, Bolsista PIBIC/CNPq)

E-mail: heitorguerrac@gmail.com

Pedro R. Andrade (CCST/INPE, Orientador)

E-mail: pedro.andrade@inpe.br

Julho de 2015

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA

(PIBIC/CNPq/INPE)

PROJETO

VISUALIZAÇÃO INTERATIVA DE DADOS E CENÁRIOS AMBIENTAIS USANDO CONTROLE DE GESTOS

PROCESSO: 800011/2014-7

Heitor Guerra Carneiro (Faculdade de Tecnologia de São José dos Campos, Bolsista PIBIC/CNPq)

E-mail: heitorguerrac@gmail.com

Pedro R. Andrade (CCST/INPE, Orientador)

E-mail: pedro.andrade@inpe.br

Julho de 2015

DADOS DE IDENTIFICAÇÃO

Título do Projeto:

VISUALIZAÇÃO INTERATIVA DE DADOS E CENÁRIOS AMBIENTAIS USANDO CONTROLE DE GESTOS

Processo CNPq Nº: 800011/2014-7

Bolsista: **Heitor Guerra Carneiro**

Acadêmico do curso de Análise e Desenvolvimento de Sistemas

Faculdade de Tecnologia de São José dos Campos – FATEC

Orientador: **Dr. Pedro Ribeiro de Andrade Neto**

CCST/INPE

Locais de Trabalho/Execução do Projeto:

Centro de Ciência do Sistema Terrestre – CCST/INPE

RESUMO

O objetivo deste projeto é desenvolver uma ferramenta computacional para a visualização interativa de dados espaciais. O ambiente computacional adotado utiliza a linguagem de programação Java, o sensor de gestos Microsoft Kinect, a biblioteca para a interação por gestos SimpleOpenNI e o globo virtual NASA World Wind 2. O resultado apresentado foi uma ferramenta que possui mecanismos para controle e seleção de um conjunto de dados espaço-temporais usando gestos. Os comandos gestuais desenvolvidos permitem mover o mapa, aproximar e afastar a visualização, alterar os dados a serem visualizados, visualizar um mesmo dado em diferentes tempos e selecionar um dado especifico. Conclui-se que o uso de dispositivos interativos é promissor em estudos de cenários complexos, por permitir uma melhor interação e visualização de informações com mais detalhes.

ABSTRACT

The objective of this project is to develop a computational tool for interactive visualization of spatial data. The adopted computational environment uses the Java programming language, the Microsoft Kinect gestures sensor, the library for gestural interaction SimpleOpenNI and the virtual globe NASA World Wind 2. The presented result was a tool that has mechanisms to control and select a set of spatio-temporal data using gestures. The designed gestural commands let you move the map, zoom in and out, change the visualized data, view the same data in different times and select a specific data. It is concluded that the use of interactive devices in studies of complex scenarios, is promising for allowing a better interaction and display of higher detailed information.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág.

SUMÁRIO

Pág.

1 INTRODUÇÃO

A comunicação de resultados científicos para a comunidade em geral é difícil e até arriscada (Weingart et al., 2000). Muitos são os fatores que levam as pessoas a não acreditarem nas mudanças climáticas, tais como: invisibilidade das causas, impactos distantes, falta de gratificação por ações de mitigação, complexidade e incerteza, bem como sinais inadequados indicando a necessidade de mudança (Moser, 2010). O uso de ferramentas computacionais com mecanismos para controle por gestos é uma abordagem promissora para facilitar a comunicação de resultados de pesquisas científicas, por permitir a interação dos usuários com grandes volumes de dados e por ser uma ferramenta de fácil aprendizado (Fuhrmann, MacEachren, Dou, Wang & Cox 2005).

O controle de gesto é uma abordagem promissora e permite a integração de diferentes tipos de sistemas, ampliando a interação de uma interface natural com o usuário, superando questões de acessibilidade e possibilitando uma melhora na aprendizagem. O desenvolvimento de ferramentas interativas tem aumentado significativamente nos últimos anos. Este avanço se deve ao surgimento de dispositivos de fácil aquisição, como exemplo o sensor de movimento Kinect (Microsoft, 2010) e o controle Wii Remote (Nintendo, 2006), dispositivos que possibilitam a interação com ferramentas computacionais utilizando movimentos corporais.

A visualização interativa através do uso de gestos pode facilitar o processo de comunicação de resultados científicos, pelo fato de auxiliar no dinamismo do conteúdo, proporcionando ao usuário uma maior interação com a informação. Este aumento na visibilidade, qualidade e acesso possibilita a percepção de um maior realismo nos dados observados.

Este documento descreve, através do desenvolvimento de um software, um conceito recente em que um comando do mundo real é transmitido através de gestos com auxílio de um equipamento. Nesta perspectiva, o objetivo deste projeto é desenvolver uma ferramenta computacional com mecanismos para controle de gestos com o objetivo de

visualizar dados espaço-temporais resultantes de pesquisas produzidas pelo Centro de Ciência do Sistema Terrestre (CCST/INPE).

2 DESENVOLVIMENTO

A ferramenta desenvolvida permite que o usuário interaja com o computador através do reconhecimento de comando produzidos através de movimentos corporais. A transição de dados e manuseio do mapa são realizadas usando apenas gestos com as mãos.

2.1. Requisitos do Sistema

O conjunto de atividades descritos neste item têm como finalidade o explicar das restrições de funcionamento do software desenvolvido, usando diagramas UML 2.0.

2.1.1. Requisitos Funcionais

A interação do usuário com a ferramenta é possível mediante ao uso de movimentos corporais. O diagrama de caso de uso (figura 1) representada os possíveis comandos executados pelo usuário, estes comandos são especificações de ações que o sistema deve ser capaz de executar, desta forma cada gesto reconhecido apresenta uma ação que altera o dado visualizada no globo virtual. Pode-se observar no diagrama que o requisito definido como setIDraw tem como dependência um dos gestos reconhecidos.

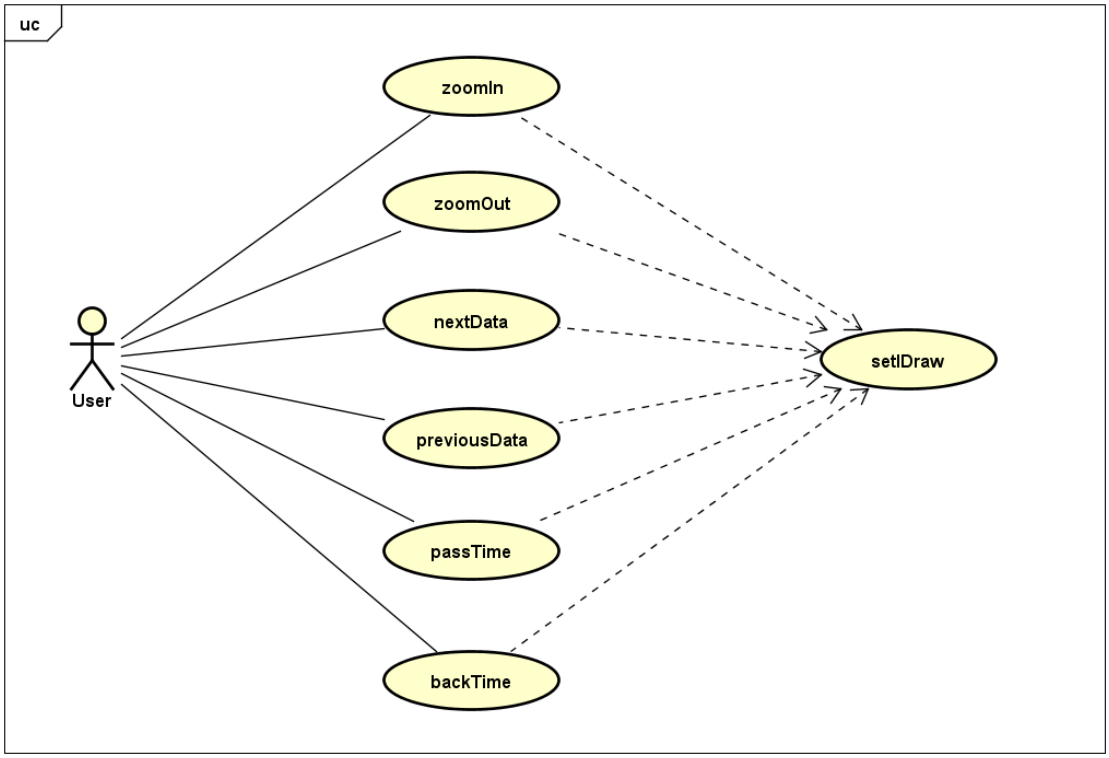

Figura 1 - Diagrama de caso de uso dos requisitos funcionais.

2.1.2. Requisitos Não-Funcionais

As características que descrevem a qualidade e usabilidade da ferramenta foram definidas na tabela 1, a qual representa as restrições de serviços para o sistema desenvolvido. Como exemplo, foi definido por meio dos requisitos funcionais que o sistema tem seis entradas gestuais possíveis, este conjunto de entradas cria a restrição que o sistema deve reconhecer todos movimentos corporais definidos.

Atributo	Descrição	Requisito
Reconhecimento de Gestos	O sistema deve reconhecer pelo menos 6 movimentos.	Requisito do Produto
Tempo de Resposta	Ao detectar um movimento, o sistema deve atualizar o mapa em menos de 1 segundo.	Requisito do Produto
Tipo de Interface	O controle da ferramenta será gestual.	Requisito do Produto
Tolerância a Falhas	O algoritmo de reconhecimento deve reconhcer ao menos 70 % dos movimentos realizados.	Requisito do Produto
Plataformas Operacionais	Microsoft Windows 7	Requisito do Produto
Facilidade de Uso	Usuários deverão conseguir operar o sistema após 10 minutos de treinamento.	Requisito do Produto
Implementação	O sistema deverá ser desenvolvido na linguagem de programação Java.	Requisito Organizacional
Padrão de Desenvolvimento	Uso de programação orientada a objeto e padrão de projeto Model-View-Controller.	Requisito Organizacional
Direitos Legais	O sistema deve ser de código aberto.	Requisito Externo

Tabela 1 - Tabela de requisitos não funcionais do sistema.

2.2. Estrutura do Sistema

A estrutura do sistema desenvolvido utiliza os recursos de orientação a objeto da linguagem de programação Java e padrões de projeto, para facilitar o processo de alteração e atualização do código-fonte.

A documentação descrita neste item usa notação UML 2.0.

2.2.1. Arquitetura de Software

A arquitetura da infraestrutura do sistema utiliza o padrão MVC (Model-View-Controller), que separa a representação da informação da interação do usuário, em vista

de desacoplar a identificação dos gestos da visualização, possibilitando alterações em quaisquer camadas independentemente.

2.2.2. Modelo do Domínio

A universidade de Muenster, na Alemanha (Bartoschek et al., 2013) disponibilizou um protótipo de uma ferramenta para visualização interativa de dados espaciais para o INPE, o Triangle of Sustainability. O equipamento é composto por: um computador, um projetor, um aparelho Microsoft Kinect, e uma tela de cerca de 2m de largura por 2m de altura onde os mapas são projetados.

O protótipo estudado para o desenvolvimento deste projeto foi desenvolvido há três anos atrás, sendo necessário atualizar o software para as novas versões dos drivers utilizados. Foi verificada uma incompatibilidade entre as novas versões do Java com os drivers OpenNI e NITE, necessários para o rastreamento das articulações do corpo humano e para o reconhecimento de gestos. Desta forma optou-se por substituir o OpenNI e NITE pelo driver Kinect SDK e por a biblioteca SimpleOpenNI. Com isso, foi necessário reescrever o código responsável pelo controle gestual.

A figura 2 descreve a classe principal do protótipo para iniciar o sensor Kinect e o reconhecimento de movimentos corporais. O diagrama de classes do Modelo do Domínio completo pode ser acessado em: http://i.imgur.com/lB1r3dL.png.

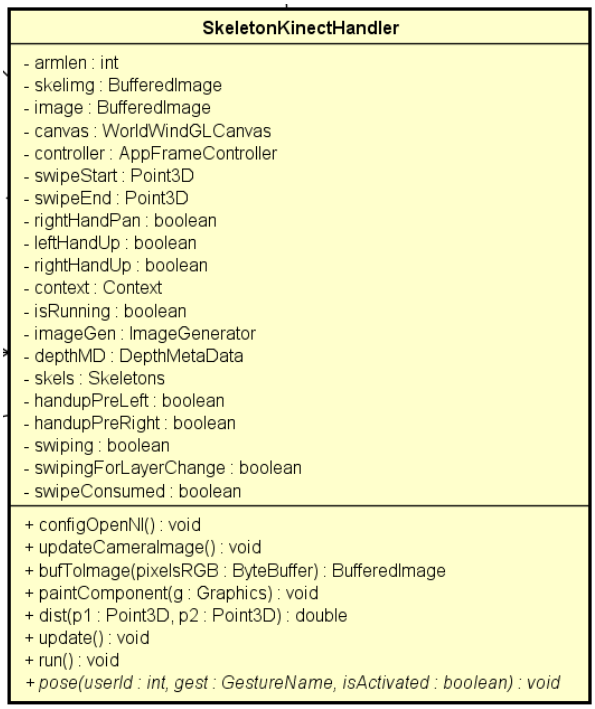

Figura 2 - Classe responsável por inicializar o sensor de movimento Kinect.

O protótipo do software desenvolvido era de difícil alteração e atualização, pois não fazia uso dos recursos de orientação a objeto e utilizava listas como estrutura de armazenamento de dados espaciais, presentes em uma única classe.

O software desenvolvido neste projeto permite o reconhecimento de gestos de maneira dinâmica, a arquitetura de software desenvolvida é de fácil alteração e inserção de novos movimentos. Foi desenvolvido também um componente responsável por gerenciar e realizar as buscas em banco de dados.

2.2.3. Diagrama de Integração

O principal recurso interativo da ferramenta é o reconhecimento de gestos corporais. O diagrama de sequência do reconhecimento de gestos (figura 3) modela os aspectos dinâmicos do sistema, mostrando a interação organizada em uma sequência temporal, apresentando o usuário e os objetos que participam do reconhecimento de gestos e a sequência de retorno das informações trocadas.

A primeira interação do usuário com o sistema ocorre no inicializar dos objetos responsável por gerenciar a janela gráfica e o reconhecimento de gestos. O sensor de movimento Kinect atualiza a uma frequência de 30 quadros por segundo. Cada quadro de posicionamento do usuário é recebido e inserido em uma lista. Esses dados são usados para o desenho da representação do usuário na tela e para iniciar a análise de movimento. Quando um movimento corporal é detectado, uma notificação do sistema é enviada e a sequência de processamento é recomeçada.

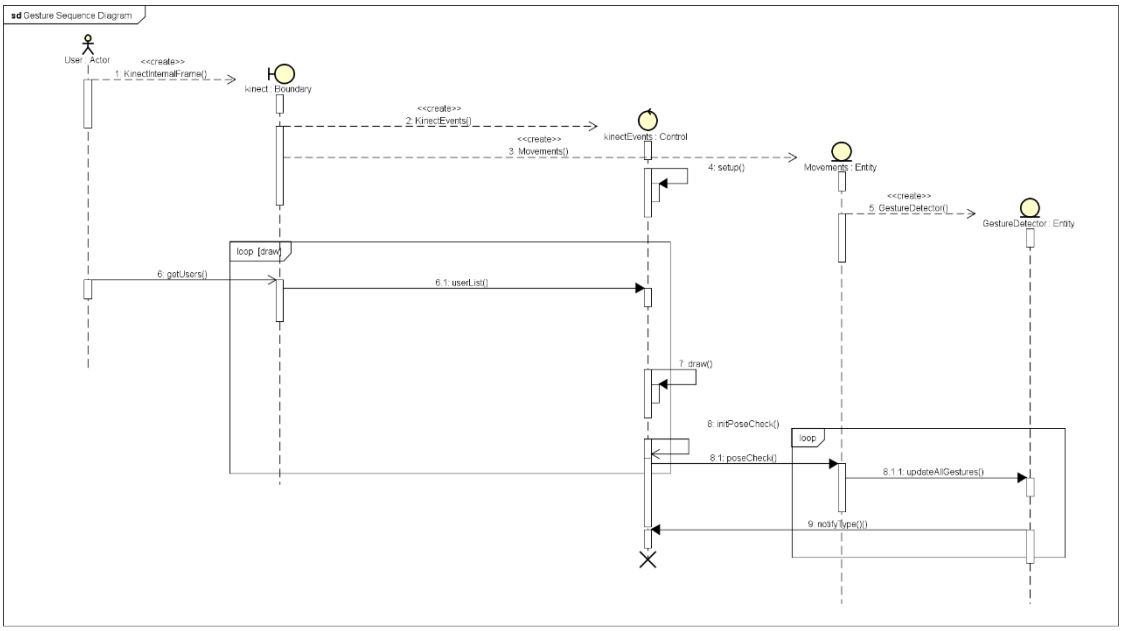

Figura 3 - Diagrama de sequência de reconhecimento de gestos.

2.2.4. Diagramas de Classe

Como descrito no cronograma do projeto, foi desenvolvido primeiro o sistema de visualização. Novos gestos foram inseridos a ferramenta, e a estrutura de criação dos mapas foi adequada ao padrão MVC. Para facilitar o entendimento, o diagrama de classe do projeto foi divido em duas partes, controle gestual e visualização de dados espaço-temporais.

A figura 4 apresenta o núcleo responsável pelo reconhecimento de gestos. O diagrama completo do controle gestual pode ser acessado em: http://i.imgur.com/cmgRhEz.png.

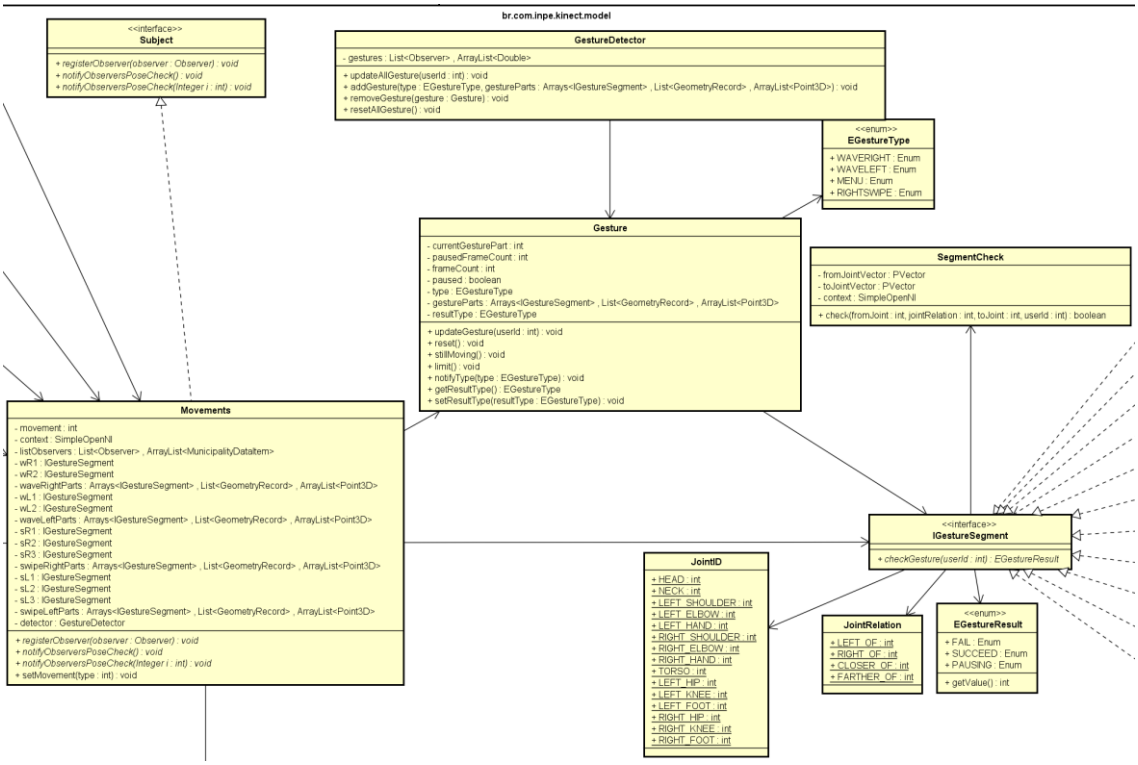

Figura 4 - Parte do diagrama de classe do controle de gestos.

A figura 5 apresenta a parte responsável por criar o globo virtual e visualizar as alterações de dados espaço-temporais. O diagrama de visualização de dados espaço-temporais pode ser acessado em: http://i.imgur.com/duaDtmd.png.

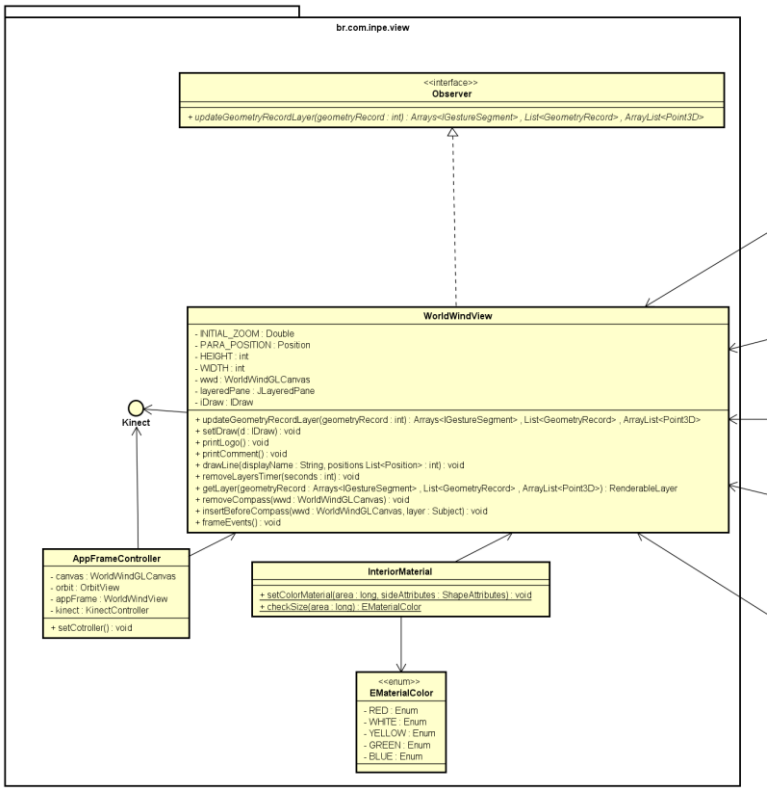

Figura 5 - Parte do diagrama de classes globo virtual.

2.2.5. Diagrama de Atividade

A integração entre o controle gestual e a visualização de dados é representado pelo diagrama de atividade (figura 6), em que apresenta o fluxo de troca de informações de uma atividade para a outra no sistema.

O processo começa no inicializar do sensor de movimentos e da janela gráfica, a qual exibe os dados de posicionamento do usuário. Cada informação recebida pelo sensor é desenhada e verificada em busca por gestos definidos. O reconhecimento de um gesto implica na alteração das informações exibidas no globo virtual.

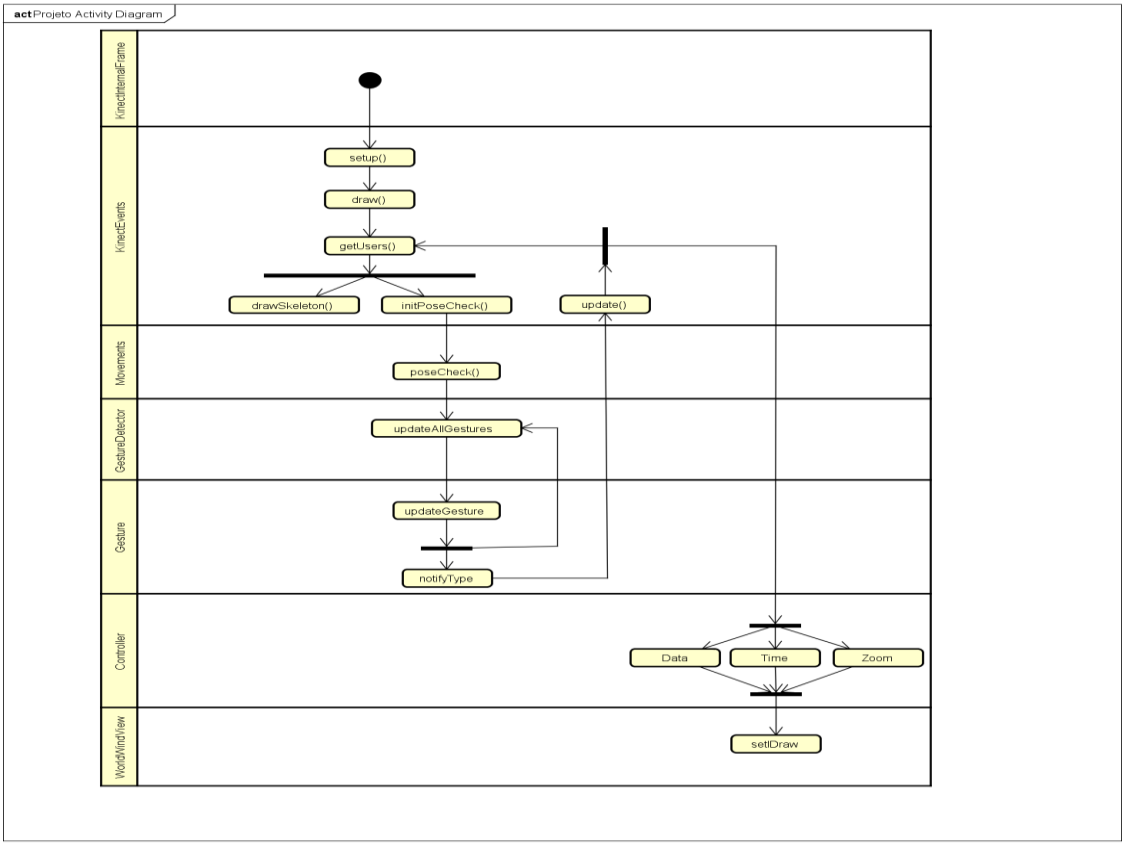

Figura 6 - Diagrama de atividade sobre o fluxo do sistema.

2.2.6. Modelo de Dados

O protótipo desenvolvido pelos pesquisadores da Universidade de Munster incorporava o uso único de Linked Open Data (Kauppinen et al., 2014) para a importação e visualização de dados espaciais, limitando-se para o estado do Pará. Os dados escritos em linguagem RDF (Resource Description Framework) ocasionavam lentidão no carregamento e interpretação dos dados devido ao alto consumo de memória.

A próxima etapa do projeto será criar um banco de dados espaço temporal com a biblioteca TerraLib, desenvolvida pela DPI/INPE, devido ao seu suporte para armazenamento e acesso à dados espaço-temporais, possuindo estruturas de dados eficientes tanto no consumo de memória quanto no tempo de processamento.

Para as realizações de testes, foi utilizado o banco de dados orientado a objeto db4o (Versant Corporation, 2004).

2.3. Implementação, Resultados e Discussão

2.3.1. Ambiente de Desenvolvimento

O software foi desenvolvido realizando as etapas de planejamento, análise de requisitos, projeto, codificação e teste. Optou-se a metodologia ágil de desenvolvimento (Martin, 2003). Para a execução deste projeto foram utilizados os seguintes softwares:

- Sistema Operacional Windows;
- Linguagem de Programação Java;
- IDE de Desenvolvimento Eclipse;
- Sistema de Versionamento de Código-Fonte GitHub.

Devido as dependências de drivers para o desenvolvimento utilizando o sensor Microsoft Kinect é recomendado o uso do sistema operacional Windows.

2.3.1.1. Sistemas e Componentes externos utilizados

Em vista de completar e facilitar o desenvolvimento do sistema, foi utilizado os seguintes componentes:

- GDAL;
- World Wind Java SDK 2;
- Microsoft Kinect;
- Kinect SDK 1.8;
- SimpleOpenNI;

O World Wind Java SDK 2 fornece um globo virtual para a visualização de mapas e informações geográficas. A biblioteca GDAL é utilizada para a importação de dados espaço-temporais (GDAL, 2014). O sensor Microsoft Kinect possui uma câmera de alta resolução e um conjunto de sensores capazes de detectar até vinte articulações do corpo

humano com uma taxa de captura de dados de trinta quadros por segundo. O Kinect SDK fornece os drivers de uso para o sistema. A biblioteca SimpleOpenNI simplifica e acelera o desenvolvimento de aplicações com o sensor Kinect.

2.3.2. Implementação

A partir das definições feitas nas etapas de planejamento e análise de requisitos, o desenvolvimento foi divido em duas partes: interação por gestos e a visualização de dados espaço-temporais.

2.3.2.1. Controle de Gestos

Foi investigado novos gestos para a visualização espaço–temporal. Inicialmente foi definido dois modos de controle para a ferramenta, um modo gestual e outro virtual, usando botões virtuais na tela. A figura 7 mostra o modo gestual. A integração entre o controle virtual e gestual foi feita definindo botões acima da cabeça do usuário, em que quando são pressionados o movo virtual fica ativo e os botões de configuração aparecem na tela (figura 8), dando ao usuário um controle mais complexo sobre o mapa.

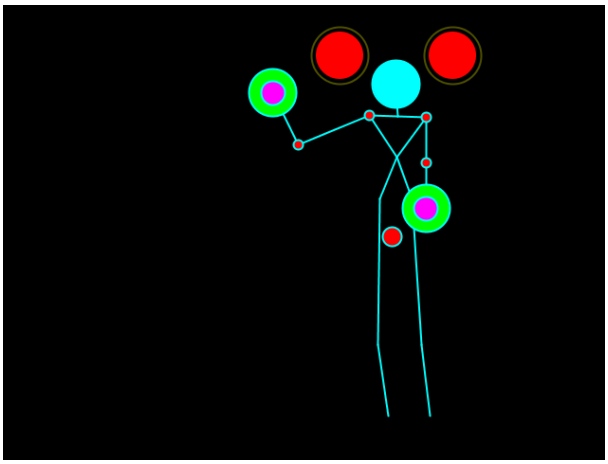

Figura 7 – Modo gestual de visualização. Os dois círculos são usados para ativar o modo virtual caso seja selecionado.

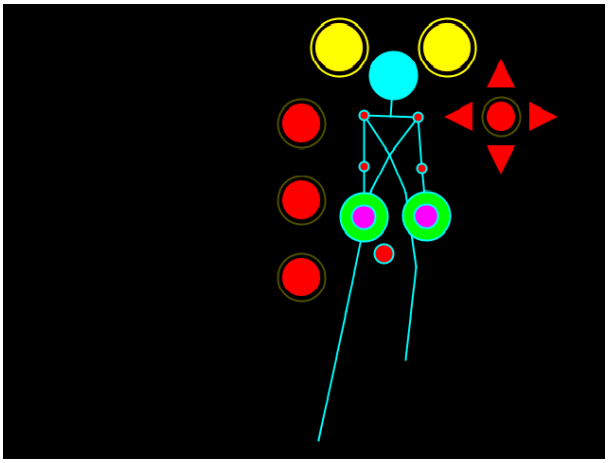

Figura 8 – Modo virtual de visualização. Novos botões são adicionados à tela de forma a fornecer novas opções de comando para o usuário.

Porém, observou-se problemas técnicos, o uso destes botões virtuais atrasava o tempo de reconhecimento dos gestos, ocasionando movimentos involuntário no mapa. Para evitar erros de interpretação do software, foi definido como padrão o modo gestual de visualização. Com isso, novos movimentos foram inclusos ao sistema.

Para o manuseio do mapa foi definido um modo de visualização e outro de configuração. O modo de configuração é definido com a mão esquerda posicionada na altura do ombro. Ele permite as alterações de dados espaço-temporais e a visualização de um mesmo dados em diferentes tempos. Existem movimentos para alterar os dados a serem visualizados, visualizar os dados em diferentes tempos e selecionar um dado especifico. Os movimentos do modo de visualização são o de mover o mapa, simulando o movimento do mouse, e o de aproximar e afastar a visualização.

2.3.2.2. Algoritmo de Reconhecimento de Gestos

Em vista de criar uma estrutura de fácil atualização, cada gesto é divido em segmentos, estes implementam a interface IGestureSegment que possui somente a assinatura do método checkGesture. O método checkGesture utiliza os recursos da classe SegmentCheck, em que recebendo como parâmetro os valores referentes as articulações a serem comparadas e o fator de comparação, representado pelas constantes da classe

JointRelation, retorna um valor booleano conforme a análise feita. Este valor determina se o usuário executou corretamente um dado segmento, caso o conjunto de segmentos analisados apresente valores verdadeiros o movimento é detectado.

A base de toda a análise de movimentação acontece na classe Gesture, que recebe como parâmetros uma matriz de segmentos e o gesto a ser comparado. O método updateGesture verifica em que estado se encontra cada segmento de gesto (FAIL, SUCCEED, PAUSING), caso o resultado desta verificação seja falho a análise é recomeçada (figura 9). A classe responsável por gerenciar os gestos é a GestureDetector, que a parir do valor recebido do sensor, faz a chamada do método updateGesture (figura 10).

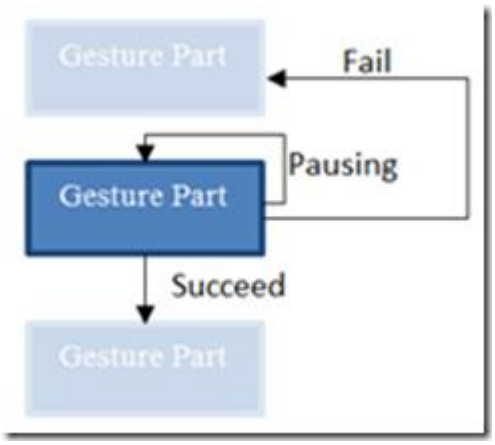

Figura 9 - Representação dos resultados e loop de processamento.

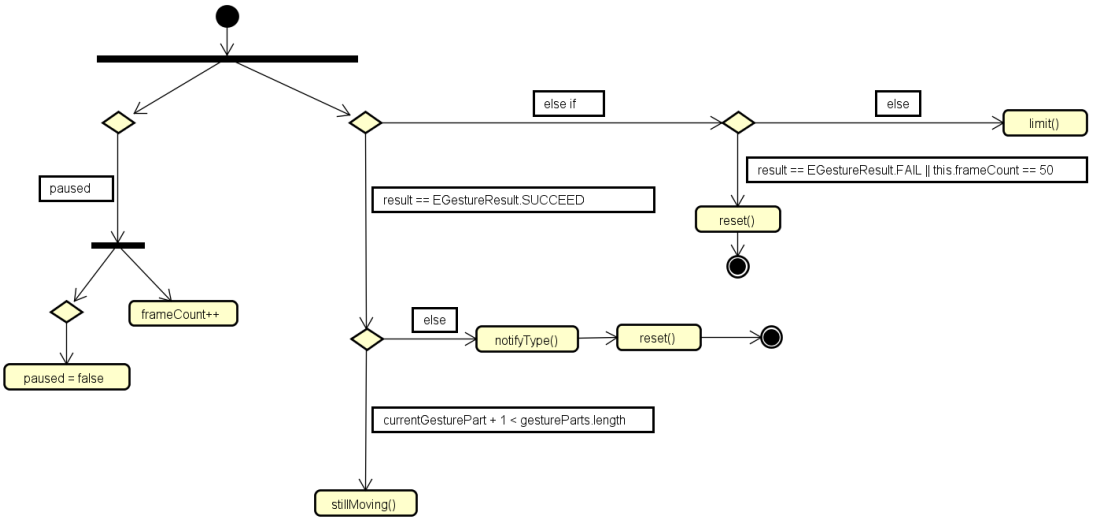

Figura 10 - Diagrama de atividade sobre o algoritmo de reconhecimento de gestos.

A classe principal do Model (Model-View-Controller) é Movements, o qual é responsável por realizar a chamada do método que verifica os gestos e notificar o gesto que obtiveram como resultado sucesso.

A API SimpleOpenNI foi desenvolvida inicialmente para a IDE Processing, software desenvolvido para projetos visuais, e sua inicialização é por PApplet. Com isso, possui algumas características como métodos obrigatórios (setup e draw) e o uso de generalização efetuado pela classe Processing.

A classe KinectEvents é responsável por: inicializar o sensor de movimento Kinect, desenhar o esqueleto do usuário na tela, realizar a chamada para a verificação de gestos e a direcionar as ações gestuais recebidas para as classes que controlam o globo virtual.

As classes do pacote Controller são responsáveis por fazer a chamada de métodos da classe AppFrameController, cujo ocasiona as mudanças dos dados espaço-temporais e o zoom do globo virtual.

2.3.2.3. Desenho de Mapas no Globo Virtual

A filtragem de dados para os desenhos dos mapas é realizada na classe WorldWindModel, em que com o uso de threads anônimas realiza as buscas no framework de banco de dados orientado a objeto db4o. Os dados selecionados são enviados para a View do globo virtual utilizando o método notifyObserverGeometryRecord.

A classe WorldWindView inicializa o globo virtual e é responsável por plotar os mapas e figuras na tela. Cada tipo diferente de objeto desenhado na tela recebe como parâmetro uma interface do pacote de controllers, as classes que implementam essa interface executam os métodos em threads, as quais permitem a criação de: mapas altimétricos, comentários, desenhos de linhas, desenhos de polígonos e anotações.

A cor de cada ponto do mapa é definida a partir de um parâmetro do tipo long do método estático da classe InteriorMaterial. Como parâmetro de teste utilizou-se alguns dados geográficos e um enum EMaterialColor para exibir as legendas das respectivas cores na tela.

As movimentações do globo virtual ocorrem mediante a classe AppFrameController, que possuí métodos para alterar zoom, latitude, longitude e rotação do globo.

2.3.3. Testes

O software foi desenvolvido com base no TDD (Test Driven Development) e os testes foram divididos em duas partes, testes estruturais e testes funcionais. O teste estrutural foi realizado utilizando o framework JUnit 4 e o teste funcional a estrutura de tabela de decisão.

2.3.3.1. Plano de Testes

Através da verificação dos requisitos do sistema desenvolvido, foram identificados os requisitos técnicos e funcionais do sistema e com isso a lista de possíveis casos de teste descritos na tabela 2.

Nº	Descrição	Saída Esperada
001	Reconhecimento de gestos único.	Nome do Gesto
002	Reconhecimento de gestos simultâneos.	Nome do Gesto
003	Reconhecimento de múltilplos usuários.	Nome do Gesto
004	Reconhecimento de gestos em sequência	Nome do Gesto

Tabela 2 - Tabela de Plano de Testes.

2.3.3.2. Execução do Plano de Testes

A tabela 3 apresenta os testes realizados no sistema tendo como base o Plano de Testes:

Nº	Descrição	Saída Esperada	Resultado do Teste	Comentário
001	Reconhecimento de gestos único.	Nome do Gesto	Reprovado	
002	Reconhecimento de gestos simultâneos.	Nome do Gesto	Reprovado	Gestos reconhecidos, mas é necessario tratar o retorno.
003	Reconhecimento de múltilplos usuários.	Nome do Gesto	Aprovado	Gestos reconhecidos, mas o sistema apresenta lentidão.
004	Reconhecimento de Gestos em Sequência	Nome do Gesto	Reprovado	

Tabela 3 - Tabela de Execução do Plano de Testes.

Em virtude de detalhar os testes realizados, cada teste descrito no plano de teste foi realizado com o número de 20 repetições.

A tabela 5 apresenta os testes de reconhecimento de gestos únicos realizados, em que cada gesto do sistema foi testado separadamente. O resultado do teste foi que todos os gestos desenvolvidos obtiveram uma taxa de 100% de reconhecimento.

Reconhecimento de Gestos Únicos			
Nº de Testes	**Nome do Gesto**	**Saída Esperada**	**Resultado do Teste**
020	Wave Left	Nome do Gesto	100%
020	Wave Right	Nome do Gesto	100%
020	Menu	Nome do Gesto	100%
020	Swipe Left	Nome do Gesto	100%
020	Swipe Right	Nome do Gesto	100%

Tabela 4 – Tabela de reconhecimento de gestos únicos.

A tabela 5 apresenta o teste de reconhecimentos de gestos simultâneos, em que dois gestos são realizados ao mesmo tempo, como o único movimento que utiliza uma das mãos é o gesto denominado Wave, que desloca a mão na horizontal repetidas vezes, houve apenas um caso de teste, obtendo como resultado uma taxa de 100% de reconhecimento.

Reconhecimento de Gestos Simultâneos			
Nº de Testes	**Nome do Gesto**	**Saída Esperada**	**Resultado do Teste**
020	Wave Left e Wave Right	Nome do Gesto	100%

Tabela 5 – Tabela de reconhecimento de gestos únicos.

A tabela 6 apresenta os testes de reconhecimento de gestos em sequência, em que dois gestos são realizados em sequência. O resultado do teste foi uma taxa de 100% de reconhecimento.

Reconhecimento de Gestos em Sequência			
Nº de Testes	**Nome do Gesto**	**Saída Esperada**	**Resultado do Teste**
020	Wave Left e Wave Right	Nome do Gesto	100%
020	Wave Right e Menu	Nome do Gesto	100%
020	Wave Left e Swipe Left	Nome do Gesto	100%
020	Swipe Left e Swipe Right	Nome do Gesto	100%

Tabela 6 - Tabela de reconhecimento de gestos em sequência.

2.3.4. Resultados

O projeto desenvolvido tem como resultado uma ferramenta computacional para visualização interativa de resultados de pesquisa do CCST (figura 11 e 12). O projeto é de código livre e está disponível no GitHub: https://github.com/hguerra/InteractiveDataVisualization.

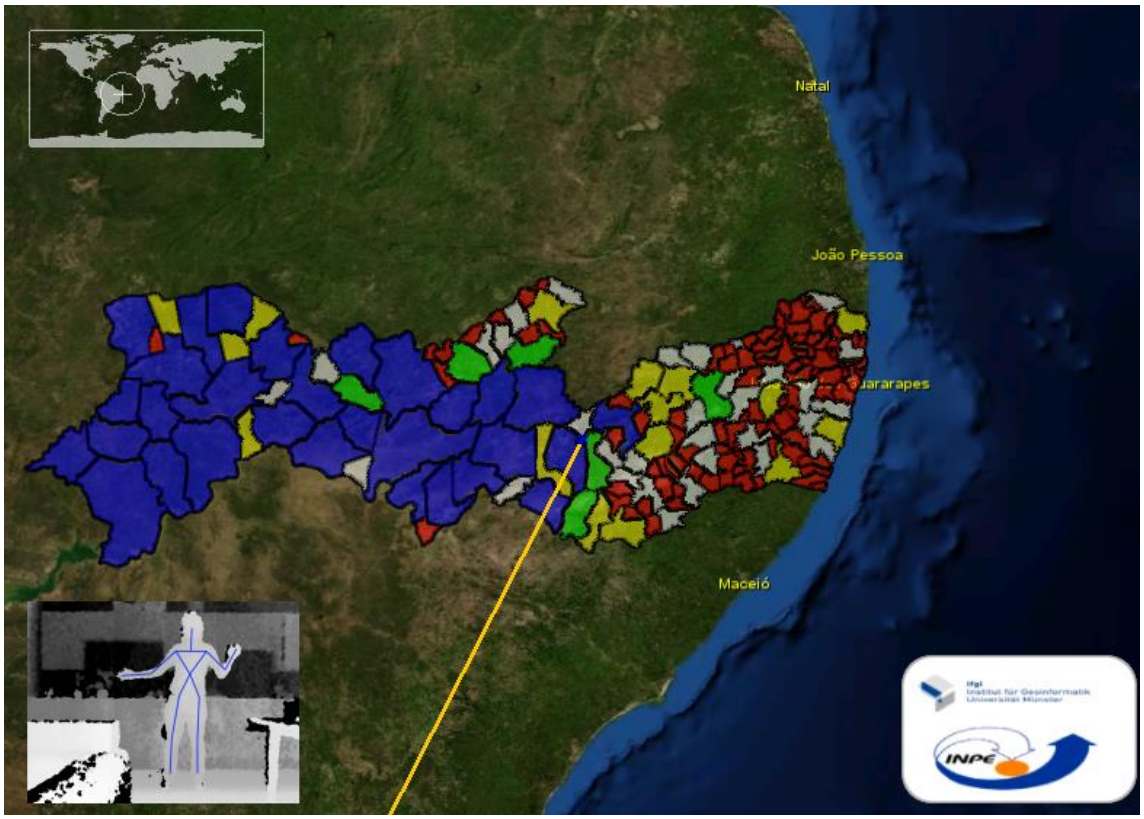

Figura 11 – Ferramenta em funcionamento.

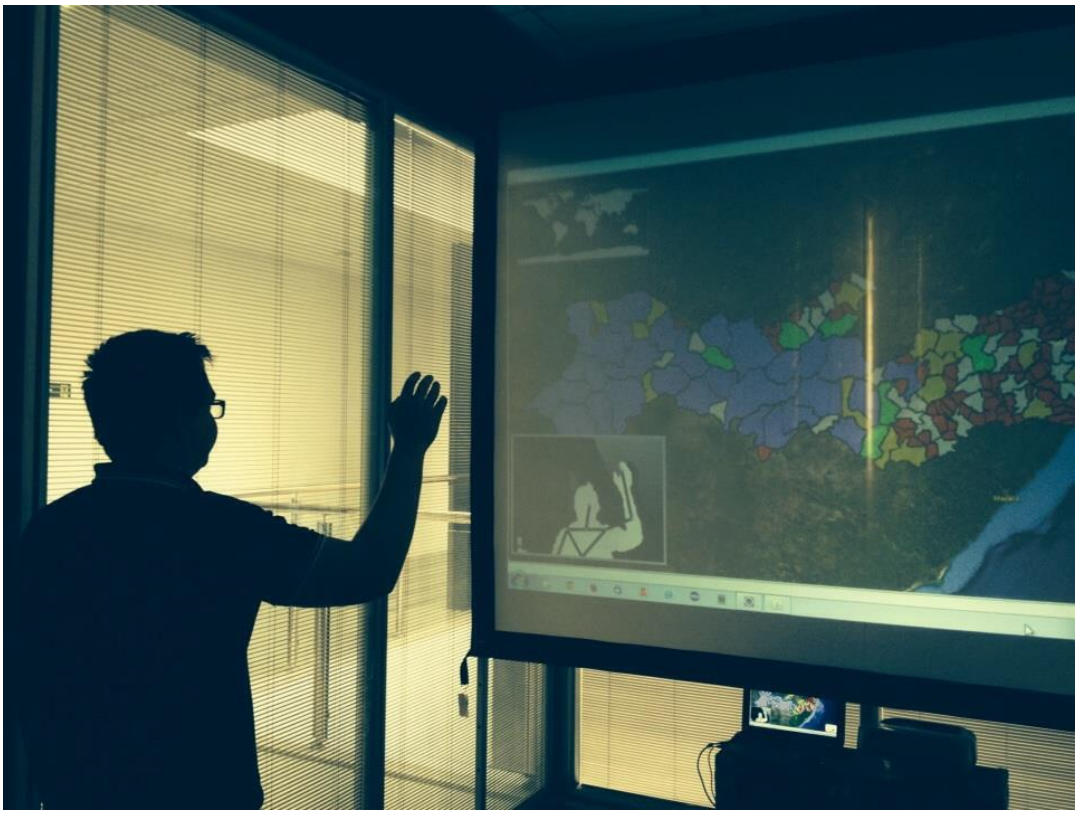

Figura 12 – Exemplo do uso do Triangle.

O mecanismo de controle de gesto desenvolvido neste projeto possibilita a adaptação em outros sistemas e a adição de novos movimentos de maneira simples. O algoritmo responsável pelo reconhecimento considera o número de imagens capturadas pela câmera e a movimentação dos membros superiores do usuário.

A ferramenta implementada possui vários gestos que ativam determinados comandos de visualização (figura 13). Por exemplo, gesto de deslocar a mão direita esta associado ao comando de mover o mapa; o gesto de aproximar e afastar as mãos esta associado à o diminuir (figura 13.a) e aumentar (figura 13.b) o zoom, respectivamente; o gesto segurar o mapa com uma mão e deslizar na horizontal com a outra esta associado a alterar os dados a serem visualizados(figura 13.c e 13.d); o gesto de segurar o mapa com uma mão e deslizar na vertical com a outra esta associado a visualizar um mesmo dado

em diferentes tempos(figura 13.e e 13.f); o gesto de empurrar efetuado com a mão direita esta associado à seleção de dados.

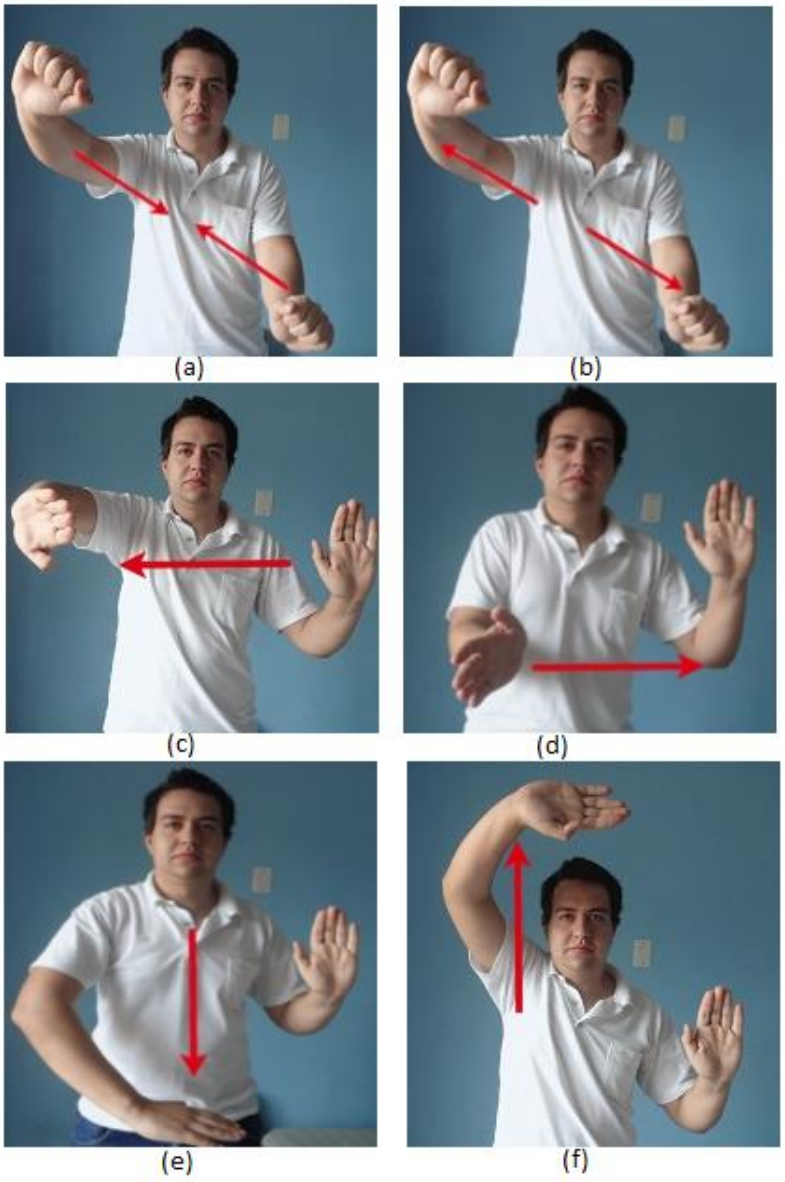

Figura 13 – Gestos da ferramenta. (a) diminuir o zoom, (b) aumentar o zoom, (c) retroceder tempo, (d) avançar tempo, (e) retroceder dado, (f) avançar dado.

Os testes qualitativos realizados indicam que os movimentos para a utilização do sistema são confortáveis e de fácil aprendizado.

2.3.5. Discussão

Neste primeiro ano de projeto, foi criado algumas versões de algoritmos de reconhecimentos de gestos e opções de controle utilizando o sensor Microsoft Kinect.

Os resultados deste projeto levam às seguintes questões:

- Reconhecimento de Gestos: Em alguns momentos problemas foram apresentados quando há o rastreamento de múltiplos gestos ao mesmo tempo, o algoritmo retorna ações simultâneas para o sistema.
- Reconhecimento de Múltiplos Usuários: Existe a possibilidade do reconhecimento de múltiplos usuários, mas isto ocasiona lentidão no reconhecimento de gestos e uma possível ação involuntária ao sistema.
- Banco de Dados: Não foram realizados testes com busca de dados espaço-temporais em banco de dados extensos, existindo a possibilidade de latência do sistema.

3 CONCLUSÃO

Este projeto estudou o uso de controle de gestos para a visualização de cenários ambientais. Foi proposto um algoritmo, para o controle gestual, adaptável e de fácil inserção de novos movimentos. Os gestos utilizados foram inspirados em comandos multi-touch para dispositivos moveis, por isso são de fácil aprendizado.

O projeto apresentou evidências que dispositivos interativos podem ser promissores em estudos de cenários complexos, por permitir a interação dos usuários com o objeto de estudo de maneira mais detalhada e uma melhor visualização da informação. Entretanto, movimentação por gestos não substituem métodos convencionais de interação com o sistema. Por exemplo, nem todos os usuários possuem agilidade e destreza muscular ou espaço físico necessário para realizar os movimentos, levando em consideração a distância mínima necessária para o uso do sensor Kinect.

4 TRABALHOS FUTUROS

Como trabalhos futuros, será desenvolvida uma ferramenta para criação e gerenciamento de um banco de dados espaciais para serem usados pelo dispositivo de visualização usando a biblioteca TerraLib, desenvolvida pelo INPE.

O novo cronograma, incluindo a nova atividade, esta na tabela abaixo. A documentação esta sendo escrita em paralelo às demais atividades de forma a reduzir o tempo de execução do projeto.

	Ago/Set	Out/Nov	Dez/Jan	Fev/Mar	Abr/Mai	Jun/Jul
1. Finalizar os gestos do *Triangle of Sustainability*	■					
2. Estudar a biblioteca TerraLib e o aplicativo TerraView		■				
3. Portar o Triangle para TerraLib e criar um banco de dados de teste		■	■			
4. Reuniões com pesquisadores do CCST				■		
5. Criar banco de dados espaço-temporal				■	■	
6. Produção de um artigo cientifico					■	
7. Relatório final						■
8. Documentação	■	■	■	■	■	■

Tabela 7 – Cronograma de atividades.

REFERÊNCIAS BIBLIOGRÁFICAS

Bartoschek, Thomas, G. Paper, C. Kray, J. Jones and T. Kauppinen. Gestural Interaction with Spatiotemporal Linked Open Data. OSGEO Journal, 11/2013, Open Source

Geospatial Foundation. 2013.

NASA. NASA World Wind, http://worldwind.arc.nasa.gov/java/. Acessado em 09-07-2015.

GDAL. Geospatial Data Abstraction Library, Java bindings. http://gdal.org/java/. Acessado em 09-02-2015.

SimpleOpenNI. https://code.google.com/p/simple-openni/. Acessado em 09-07-2015.

Processing. https://processing.org/. Acessado em 09-07-2015.

Borenstein, Greg. Making Things See. Maker Media, Inc., 1005 Gravenstein Highway

North, Sebastopol, CA 95472.

APÊNDICE A – IMPLANTAÇÃO

A.1 Configuração da biblioteca SimpleOpenNI

Para o funcionamento dos recursos da ferramenta é necessário a instalação dos seguintes aplicativos. O documento terá como modelo a instalação em uma máquina com Windows 32-bit:

- Java JDK
- Kinect SDK 1.8

Ao criar um novo projeto no Eclipse o primeiro passo é importar as bibliotecas do SimpleOpenNi e o Core do Processing.

Na opção Properties do projeto, selecione a opção Java Build Path e na guia Libraries escolha a opção Add External JARs. As bibliotecas do Processing podem ser encontradas dentro da pasta da aplicação, selecionando a pasta core e em seguida library, as bibliotecas que devem ser adicionadas são: core, glugen-rt e jogl-all.

Em seguida, na pasta SimpleOpenNI em library deve ser adicionado: SimpleOpenNI.

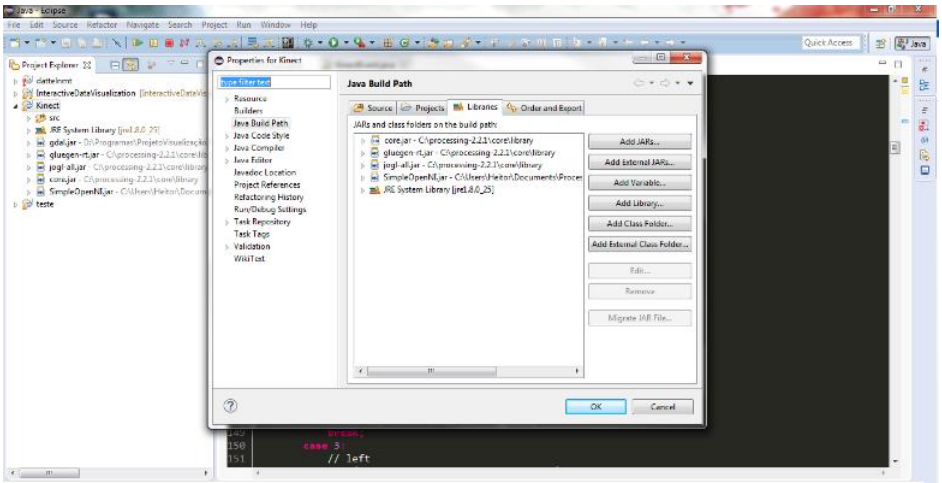

Figura A.1 - Java Build Path.

As aplicações desenvolvidas no Eclipse utilizando Processing possuem algumas características:

- Extends PApplet
- Chamadas usando PApplet.main()
- Existem métodos obrigatórios: setup() e draw()

As aplicações sem o método main podem ser executadas da seguinte maneira, com o botão direito do mouse sobre a classe, na opção Run as, selecione Java Applet.

```
1  package br.com.kinect.exemples;
2
3  import processing.core.*;
4  import SimpleOpenNI.*;
5  /**
6   *
7   * @author Heitor Guerra Carneiro.
8   * @version 1.0
9   * @since February 2015.
10  */
11 public class DepthMapAndCamera extends PApplet{
12
13         SimpleOpenNI kinect;
14
15         public void setup(){
16
17                 size(640*2, 480);
18
19                 kinect = new SimpleOpenNI(this);
20                 kinect.enableDepth();
21                 kinect.enableRGB();
22         }
23         public void draw(){
24                 kinect.update();
25                 image(kinect.depthImage(), 0, 0);
26                 image(kinect.rgbImage(), 640, 0);
27         }
28 }
29
```

Figura A.2 - Exemplo de aplicação sem o método main.

A seguir a estrutura de uma aplicação para ser iniciada como Java Application.

```
public class myStandaloneApp extends PApplet{
    public static void main(String[] args){
        String fullClassNameWithPackage = myStandaloneApp.class.getName(); // Get
our package-qualified class name
        PApplet.main(fullClassNameWithPackage); // Launch a PApplet with this clas
s used for event callbacks
    }
    public void setup(){
        // Called once to initialize window state, data structures, etc.
    }
    public void draw(){
        // Called each time a frame must be drawn; encompasses entire Update/Draw l
oop
    }
}
```

Figura A.3 - Estrutura de aplicação com o método main.

A.2 Configuração do Nasa World Wind SDK 2.0

O arquivo compactado deve ser extraído e os seguintes arquivos devem ser adicionados como biblioteca externa ao projeto:

- gdal.jar
- glugen.jar
- jogl.jar
- worldwind.jar
- worldwinx.jar

Para usuários de Windows, as bibliotecas glugen-rt.dll, jogl.dll, jogl_awt.dll, jogl_cg.dll devem ser adicionadas a pasta de bibliotecas Java. Exemplo para versão de Windows 32 bits: C:\Program Files (x86) \Java\jre7\bin.