

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

Junho de 2016

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

Junho de 2016

DADOS DE IDENTIFICAÇÃO

Título do Projeto:

VISUALIZAÇÃO INTERATIVA DE DADOS E CENÁRIOS AMBIENTAIS
USANDO CONTROLE DE GESTOS

Processo CNPq N°: 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

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. 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 OpenNI 2 e o globo virtual NASA World Wind Java SDK 2. A ferramenta desenvolvida 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 e visualizar um mesmo dado em diferentes tempos. A ferramenta desenvolvida se mostrou uma solução atrativa e intuitiva para exibir dados espaço-temporais em exposições e eventos relacionados

ABSTRACT

Interactive visualization through the use of gestures can be used to facilitate scientific results communication process by improving content dynamism and user interaction. The objective of this project is to develop a computational tool for interactive visualization of spatial data. The adopted computational environment uses Java programming language, Microsoft Kinect gestures sensor, OpenNI 2 library for gestural interaction and NASA World Wind Java SDK 2 virtual globe. It resulted 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 and view the same data at different times. The developed tool proved to be an attractive and intuitive solution for displaying spatio-temporal data in related exhibitions and events.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1- Equipamento utilizado pelo Triangle of Sustainability. (a) computador, (b) projeto, (c) sensor Microsoft Kinect, (d) tela de 2m x 2m.	20
Figura 2- Arquitetura do Sistema	21
Figura 3- Diagrama de classes simplificado do módulo de visualização de dados espaço-temporais	23
Figura 4- Diagrama de classes simplificado do módulo de controle por gestos	24
Figura 5- Diagrama de sequência do reconhecimento de gestos	25
Figura 6- Modelo de classes de vegetação do Brasil.....	26
Figura 7- Modelo de cenários do uso da terra e emissões de gases do efeito estufa na Amazônia brasileira.....	26
Figura 8- Representação do agrupamento dos dados em subconjuntos	27
Figura 9- Gestos da ferramenta. (a) diminuir o zoom, (b) aumentar o zoom, (c) retroceder tempo, (d) avançar tempo, (e) retroceder dado, (f) avançar dado	28
Figura 10- Representação gráfica do gesto Wave	30
Figura 11- Representação dos resultados possíveis da análise dos KeyFrame	32
Figura 12- Fluxograma do algoritmo de reconhecimento de gestos	33
Figura 13- Visualização do dado Vegetation Sceneario 2015.....	35
Figura 14- Gráfico de pizza criado com o dado Vegetation Sceneario 2015.....	37
Figura 15- Gráfico de barras criado com os dados: Vegetation Sceneario 2015, Vegetation Sceneario 2030 e Vegetation Sceneario 2045	37
Figura 16- Demonstração do Triangle em execução	38
Figura 17- Documentação online do Triangle.....	39

Figura 18- Interface de configuração do Triangle	40
Figura 19- Execução do gesto de mover o mapa. (a) visualização antes do comando ser acionado, (b) visualização após o comando ser acionado	41
Figura 20- Execução dos gestos: (a) gesto de alterar o dado, (b) gesto de visualizar um mesmo dado em diferentes tempos	42
Figura 21- Execução do gesto de Zoom. (a) gesto de aproximar a visualização, (b) gesto de afastar a visualização	43

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1- Tabela de requisitos funcionais do sistema	17
Tabela 2- Tabela de requisitos não funcionais do sistema	18

SUMÁRIO

		<u>Pág.</u>
1	INTRODUÇÃO	14
2	DESENVOLVIMENTO	17
2.1.	Requisitos do Sistema	17
2.2.	Estrutura do Sistema.....	18
2.2.1.	Modelo do Domínio	19
2.2.2.	Arquitetura de Software	20
2.2.3.	Diagramas de Classe	22
2.2.4.	Diagrama de Sequência.....	24
2.3.	Implementação	25
2.3.1.	Configuração do Banco de Dados.....	27
2.3.2.	Visualização Interativa.....	27
2.3.2.1.	Controle por Gestos.....	30
2.3.2.2.	Reconhecimento de Poses	31
2.3.2.3.	Reconhecimento de Gestos	33
2.3.2.4.	Visualização de Dados Espaço-Temporais	34
2.3.3.	Resultados	38
3	CONCLUSÃO	45
3.1.	Contribuições	45
3.2.	Publicação	46
4	TRABALHOS FUTUROS	47

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 por gestos 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 baixo custo, como por exemplo o sensor de movimento Kinect (MICROSOFT, 2010) e o controle Wii Remote (NINTENDO, 2006).

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. Segundo Norman (2010) sistemas interativos possibilitam o aprendizado através de exploração, considerando um recurso importante para o futuro da comunicação holística e interação das pessoas com a tecnologia.

Desta forma, este projeto tem por objetivo apresentar um ambiente computacional para a visualização interativa de dados e resultados de cenários ambientais para o Brasil utilizando mecanismos de controle por gestos. Estes cenários são pesquisas produzidas pelo Centro de Ciência do Sistema Terrestre (CCST/INPE), possuindo informações no estado-da-arte sobre as mudanças ambientais no Brasil. Eles são produzidos através da

simulação de modelos do sistema terrestre que exploram tendências futuras baseadas em estudos sobre a alteração ou não de determinados comportamentos, regras e normas, como por exemplo, a implementação do novo Código Florestal (lei 12.651/12).

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

A Tabela 1 apresenta os requisitos funcionais da ferramenta. Estes requisitos especificam ações que o sistema deve ser capaz de executar.

Tabela 1- Tabela de requisitos funcionais do sistema

Requisito	Descrição
Dados Espaço-Temporais	O sistema deve suportar dados no formato shapefile, além de possibilitar a conversão entre formatos vetoriais.
Reconhecimento de Gestos	O sistema deve possuir gestos para mover o mapa; diminuir e aumentar o zoom, visualizar um mesmo dado em diferentes tempos, além de alterar os dados a serem visualizados.
Visualização de Dados Espaço-Temporais	O sistema deve suportar visualização e seleção de múltiplos dados, possibilitando combiná-los em diferentes cenários ambientais.

A Tabela 2 apresenta os requisitos não funcionais da ferramenta. Estes requisitos especificam as restrições de serviços para o sistema desenvolvido.

Tabela 2- Tabela de requisitos não funcionais do sistema

Requisito	Descrição
Tempo de Resposta	Ao detectar um movimento, o sistema deve atualizar o mapa em menos de um minuto.
Tipo de Interface	A interação do usuário com a ferramenta deve ser mediante o uso do sensor Microsoft Kinect.
Plataforma Operacional	Os sistemas operacionais suportados são: Microsoft Windows 7, 8 ou 10.
Implementação	O sistema deve ser desenvolvido na linguagem de programação Java.
Arquitetura de Software	Uso de programação orientada a objetos e padrão de projetos MVC.
Direitos Legais	O sistema deve ser de código aberto.

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. 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 nome Triangle vem as três dimensões da sustentabilidade: ambiental, econômica e social. 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 (Figura 1).

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. Com isso, foi necessário reescrever o código responsável pelo controle por gestos da ferramenta. O Triangle incorporava o uso de Linked Open Data (KAUPPINEN et al., 2014) para a importação e visualização dos dados espaciais, limitando-se ao estado do Pará. Os dados escritos em linguagem *Resource Description Framework* (RDF) impossibilitavam a inserção de dados para serem visualizados em tempo de execução, além de ocasionar lentidão no carregamento e interpretação devido ao alto consumo de memória. Esta ferramenta era de difícil alteração e atualização, devido a falta do uso de padrões de projeto e a depreciação da versão utilizada da API OpenNI, compatível somente com o sistema operacional Windows 7.

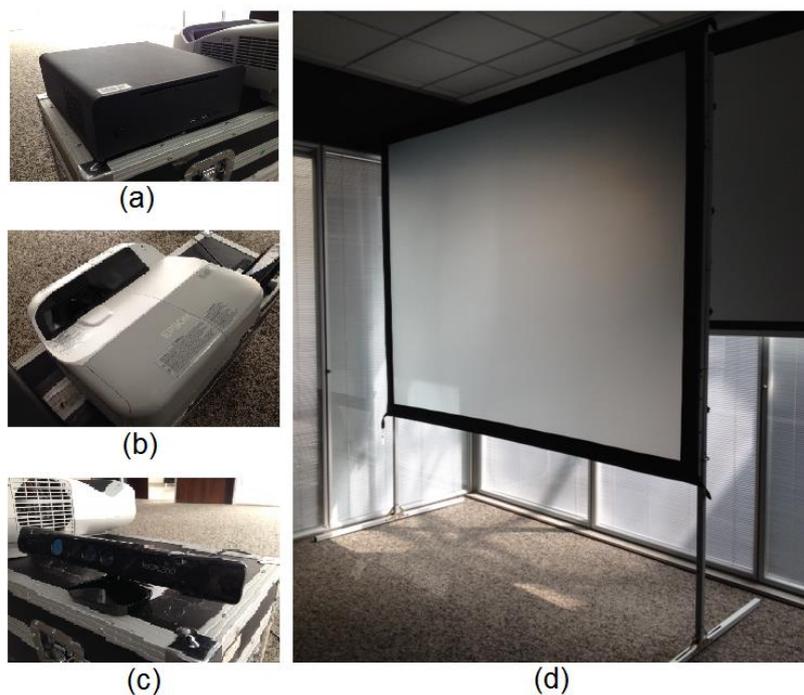


Figura 1- Equipamento utilizado pelo Triangle of Sustainability. (a) computador, (b) projeto, (c) sensor Microsoft Kinect, (d) tela de 2m x 2m.

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.

2.2.2. Arquitetura de Software

A arquitetura da infraestrutura do sistema utiliza o padrão de projetos *Model-View-Controller* (MVC), em vista de desacoplar a identificação dos gestos da visualização, possibilitando alterações em quaisquer camadas independentemente. A Figura 2 ilustra a arquitetura da aplicação.

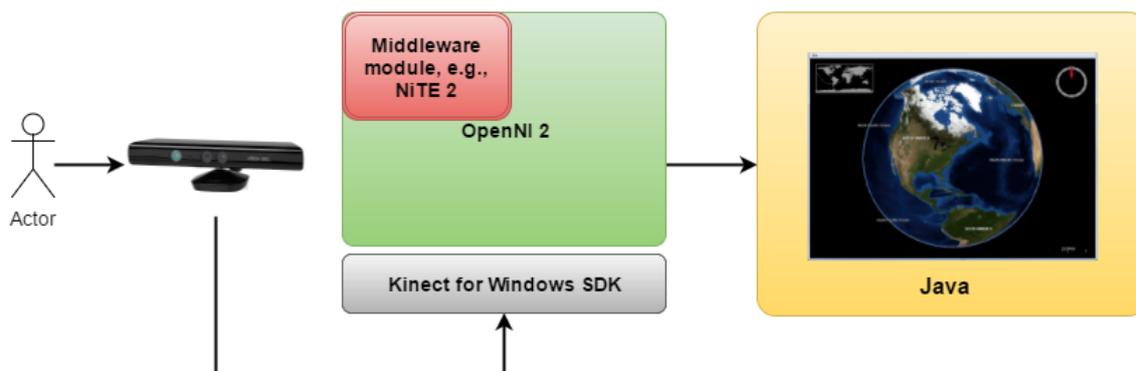


Figura 2- Arquitetura do Sistema

O software foi desenvolvido realizando as etapas de planejamento, análise de requisitos, projeto, codificação e teste. Optou-se por 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.
- GDAL;
- World Wind Java SDK 2;
- Microsoft Kinect;
- Microsoft Kinect SDK 1.8;
- OpenNI 2;
- NiTE 2;

O World Wind Java SDK 2 é uma coleção de componentes que exibem de forma interativa informações geográficas 3D em aplicações Java ou Java Applets (NASA, 2011). A biblioteca GDAL é utilizada para a importação de dados espaço-temporais (GDAL, 2000). 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 (MICROSOFT, 2010). O

Kinect SDK 1.8 fornece os drivers de uso para o sistema (MICROSOFT, 2013). A biblioteca OpenNI 2 abrange desde a comunicação com dispositivos de interação humana, como por exemplo sensores de profundidade, bem como interfaces com aplicações de terceiros, como o middleware NiTE 2 (OpenNI, 2013).

2.2.3. 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 dividido em duas partes, controle por gestos e visualização de dados espaço-temporais.

A Figura 3 apresenta o diagrama de classes simplificado do módulo de visualização de dados espaço-temporal, este módulo tem a finalidade de criar e manipular as *layers* (camadas de visualização) inseridos no globo virtual. Os *layers* são criados através do objeto *Data*. Este possui atributos como: referência bibliográfica dos dados, endereço do shapefile em disco e propriedades dos polígonos. Caso exista alteração na visualização dos componentes, estas alterações são notificadas para tela utilizando o padrão de projeto *Observer*.

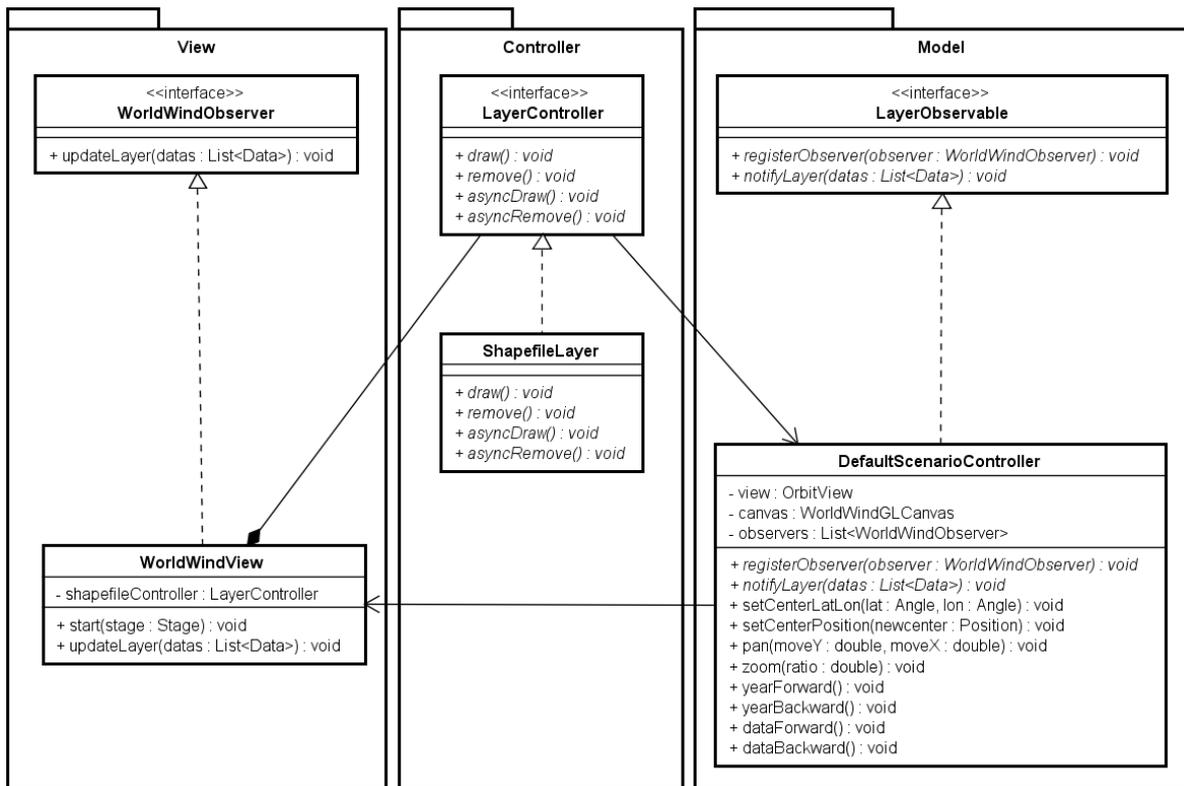


Figura 3- Diagrama de classes simplificado do módulo de visualização de dados espaço-temporais

A Figura 4 apresenta o diagrama de classes simplificado do módulo de controle por gestos. Este módulo é responsável pela interação do usuário com a ferramenta. Os gestos são validados pela implementação de GestureObservable. Caso algum movimento seja detectado a tela Kinect4jView é notificada, acionando seu respectivo Controller.

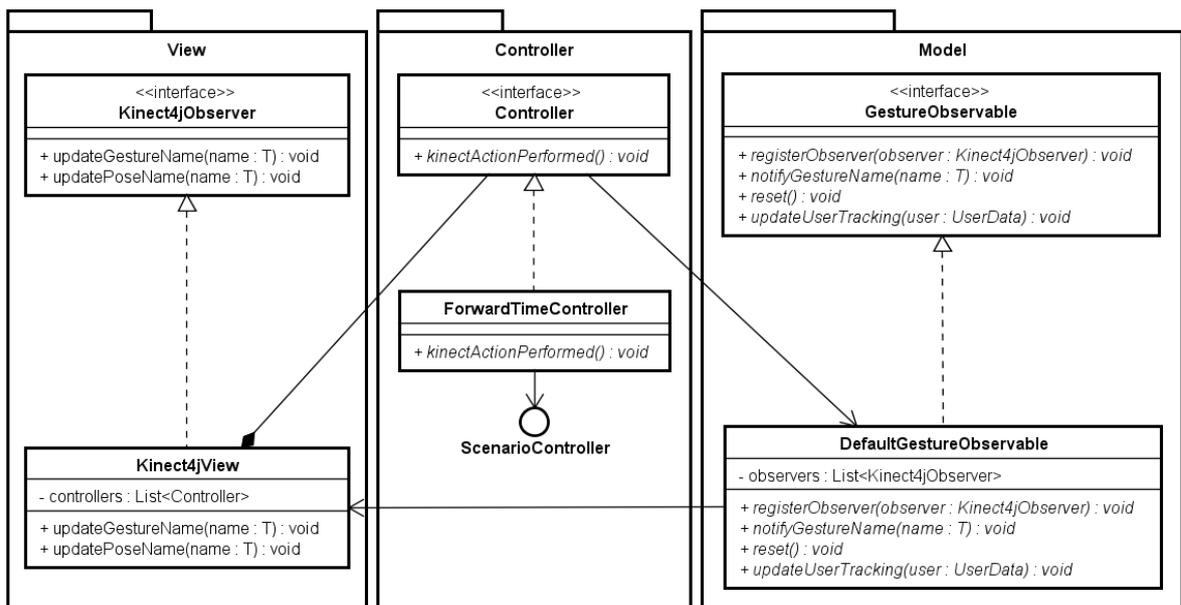


Figura 4- Diagrama de classes simplificado do módulo de controle por gestos

2.2.4. Diagrama de Sequência

O principal recurso interativo da ferramenta é o reconhecimento de gestos corporais. O diagrama de sequência do reconhecimento de gestos (Figura 5) 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 trinta quadros por segundo. Cada quadro de posicionamento do usuário é recebido e processado. 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.

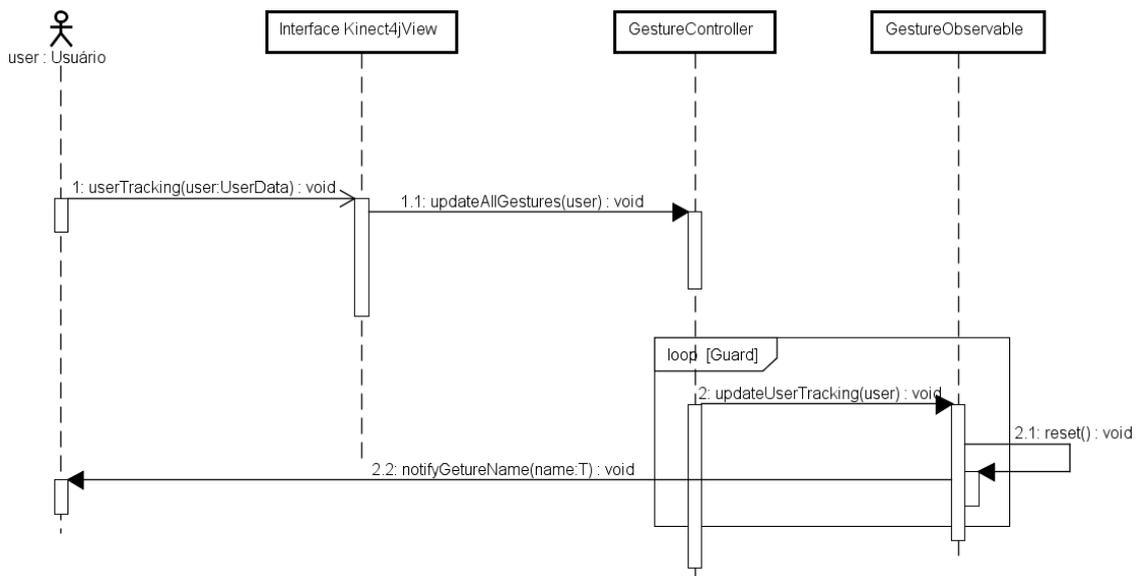


Figura 5- Diagrama de seqüência do reconhecimento de gestos

2.3. Implementação

A partir das definições feitas nas etapas de planejamento, análise de requisitos e projeto, a implementação foi dividida em duas partes: configuração do banco de dados e visualização interativa. Os dados espaço-temporais utilizados para a implementação deste projeto são resultados de pesquisas produzidas por pesquisadores do CCST/INPE:

Vegetation Scenario:

Estes dados correspondem a valores de classes de vegetação do Brasil simulados com o modelo Inland (Interação Biosfera-Atmosfera) (CARDOSO et al., 2013). A Figura 6 ilustra os cenários para os anos de 2000, 2015, 2030 e 2045 respectivamente.

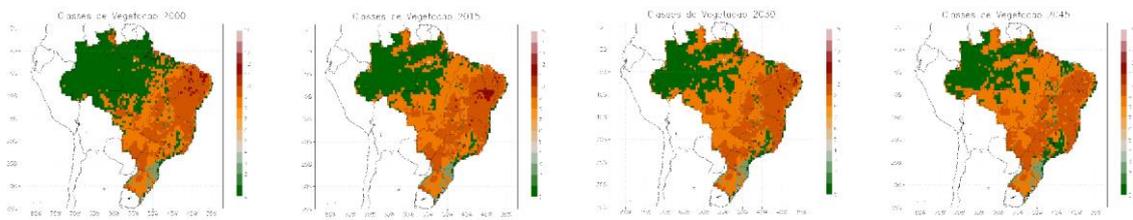


Figura 6- Modelo de classes de vegetação do Brasil

Deforestation Scenario:

Estes dados correspondem ao estudo dos cenários de uso da terra e as emissões de gases do efeito estufa na Amazônia brasileira (AGUIAR et al., 2015). A Figura 7 ilustra os cenários de desmatamento para o ano de 2050. O modelo descrito como *Scenario A* (Cenário A – Sustentabilidade) representa um futuro com um progresso significativo nas dimensões socioeconômicas e ambientais. O modelo descrito como *Scenario C* (Cenário C – Fragmentação) representa um cenário oposto, em que ocorreu um retorno de altas taxas de desmatamento e desrespeito com o Código Florestal, em conjunto com o processo de urbanização e intensificação dos problemas sociais. O modelo descrito como *Scenario B* (Cenário B – Razoável) representa um cenário intermediário, combinando as premissas dos dois cenários mais extremos. Esse cenário também considera a aplicação do Código Florestal, com taxas de desmatamento legal em torno de 4.000 km² / ano a partir de 2020.

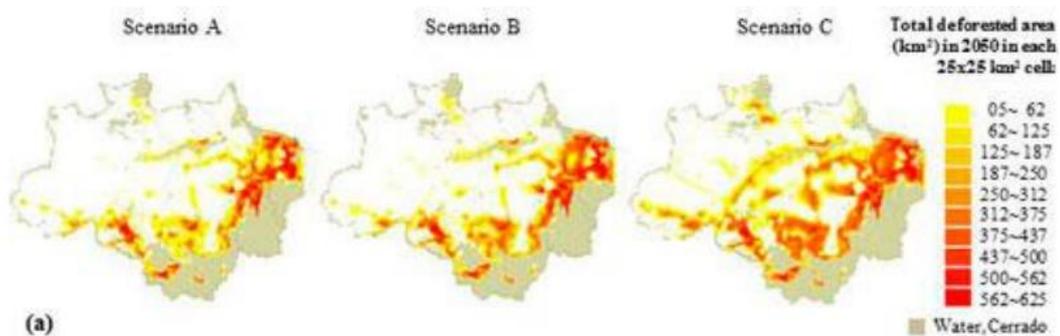


Figura 7- Modelo de cenários do uso da terra e emissões de gases do efeito estufa na Amazônia brasileira

2.3.1. Configuração do Banco de Dados

Os dados inseridos na ferramenta são classificados pelo nome do arquivo. Caso o prefixo do nome dos dados adicionados sejam iguais, o dado é agrupado em um subconjunto; o sufixo do nome do dado é utilizado para representar a sequência de exibição, ou seja, o aspecto temporal da ferramenta. A Figura 8, ilustra o exemplo da inserção dos dados: “vegtype_2000”, “vegtype_2005” e “vegtype_2010”. Este agrupamento possibilita a ferramenta considerar o dado como temporal.

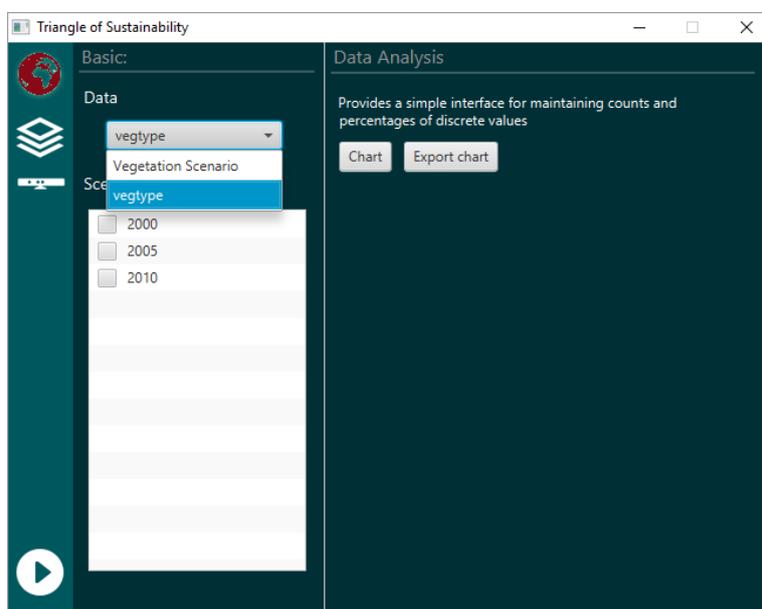


Figura 8- Representação do agrupamento dos dados em subconjuntos

2.3.2. Visualização Interativa

A ferramenta possui gestos que ativam comandos de visualização. Optou-se por gestos que se assemelham aos utilizados por dispositivos multi-touch. Por exemplo, o gesto da pinça para controlar o zoom e o arrastar para navegação.

Para a visualização interativa foi definido um modo de seleção e outro de visualização dos dados. O modo de seleção do dado a ser visualizado é definido com a mão esquerda posicionada na altura do ombro, e permite as alterações de dados espaço-temporais e a

visualização de um mesmo dados em diferentes tempos. O modo de visualização possibilita mover o mapa, simulando o movimento do mouse, e aproximar e afastar a visualização.

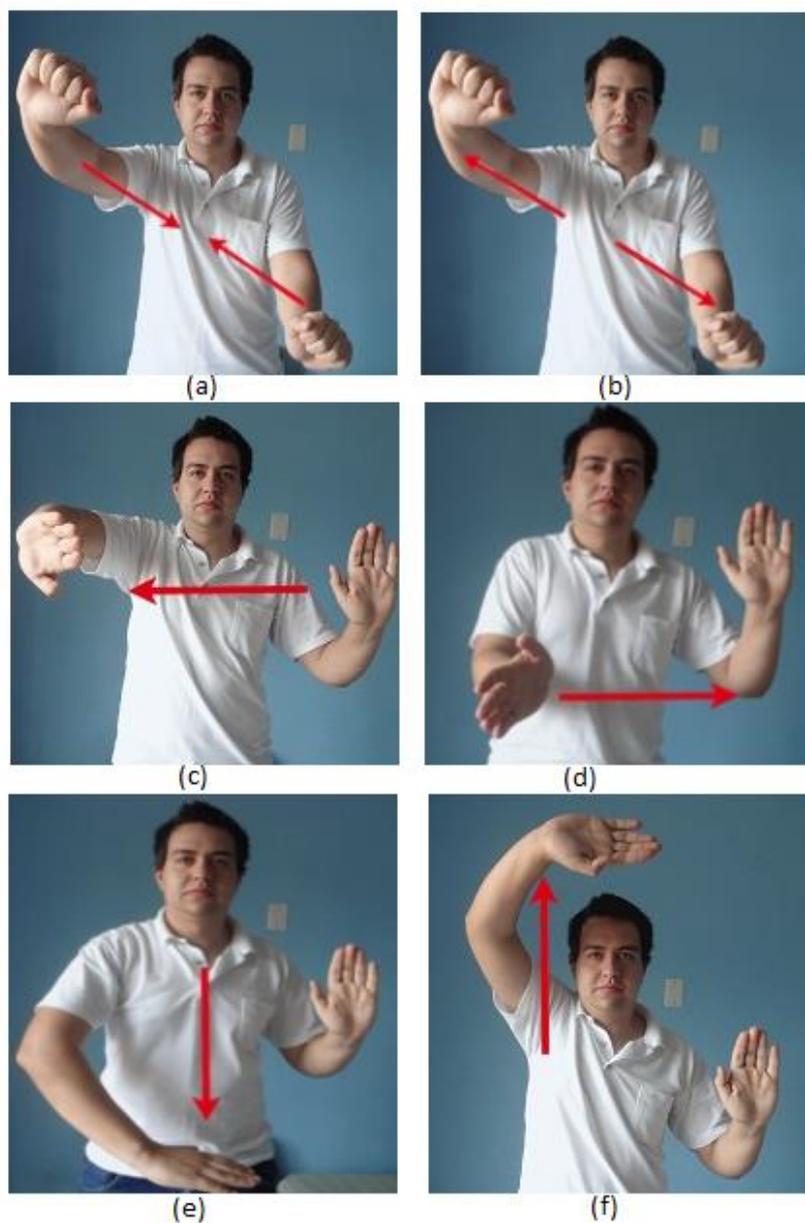


Figura 9- Gestos da ferramenta. (a) diminuir o zoom, (b) aumentar o zoom, (c) retroceder tempo, (d) avançar tempo, (e) retroceder dado, (f) avançar dado

Pan:

O gesto consiste em mover a mão direita erguida em qualquer direção. Este gesto permite ao usuário controlar qual seção do mapa será visualizada.

Zoom Out / Zoom In:

O gesto é usado para aproximar e afastar a visualização do mapa. Para executar este gesto, o usuário tem que mover as duas mãos e antebraços simultaneamente. Caso o movimento inicie com as mãos afastadas (Figura 9a), a visualização do mapa é afastada; caso o movimento inicie com as mãos próximas (Figura 9b), a visualização do mapa é aproximada.

Swipe Right / Swipe Left:

O gesto controla o aspecto temporal de um cenário ambiental. Para executar o gesto, o usuário tem que segurar o mapa com a mão esquerda e deslizar a mão direita na horizontal. Caso o movimento inicie com a mão direita a esquerda do ombro esquerdo (Figura 9c), retrocede o tempo em que o dado é visualizado; caso o movimento inicie com a mão direita a direita do ombro direito (Figura 9d), avança o tempo em que o dado é visualizado.

Swipe Down / Swipe Up:

O gesto altera um cenário ambiental visualizado. Para executar este gesto, o usuário deve segurar o mapa com a mão esquerda e deslizar na vertical com a mão direita. Caso o movimento inicie com a mão direita acima da cabeça (Figura 9e), o dado anterior do subconjunto de dados é selecionado no globo virtual; caso o movimento inicie com a mão direita abaixo do centro de gravidade (Figura 9f), o próximo dado do subconjunto de dados é inserido no globo virtual.

2.3.2.1. Controle por Gestos

Existem diversas formas para a detecção de gestos, as mais conhecidas são: redes neurais artificiais e detecção por algoritmo especializado. O processo de detecção utilizando a técnica de inteligência artificial conhecida por redes neurais artificiais é utilizado em diferentes tipos de sensores. O uso de redes neurais é eficaz, porém o processo de criação e treinamento de uma rede neural é complexo e pode ser utilizado em somente um gesto específico (CARDOSO, 2014).

A abordagem do algoritmo especializado é menos complexa, além de prover um bom resultado. Neste caso, o algoritmo interpreta um gesto como uma série de poses a serem reconhecidas em sequência em um determinado espaço de tempo (PTERNEAS, 2014). Assim, em termos de cinética, um gesto é a posição relativa de algumas articulações para um dado número de quadros. A Figura 10 ilustra a execução do gesto Wave, neste gesto a mão permanece acima do cotovelo e move-se periodicamente da esquerda para a direita.

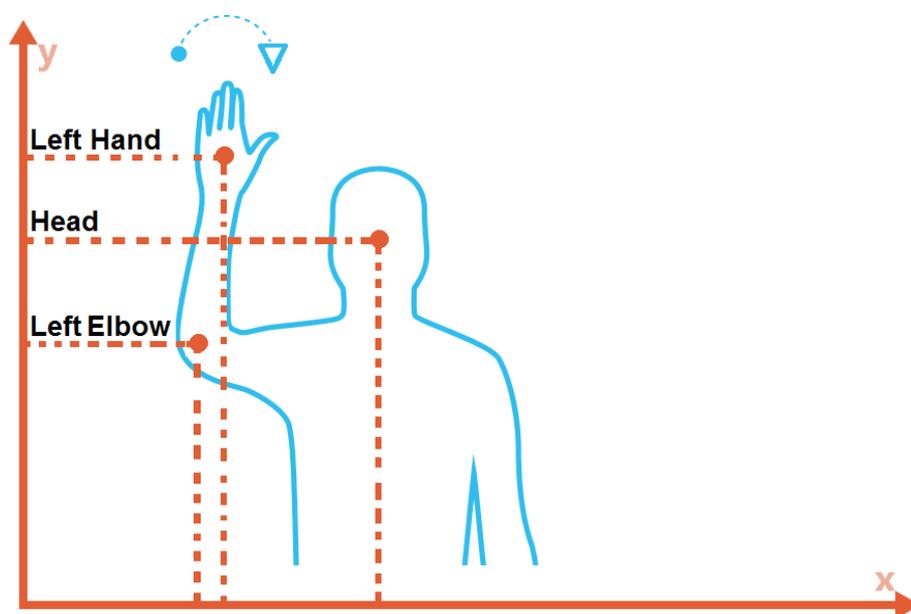


Figura 10- Representação gráfica do gesto Wave

2.3.2.2. Reconhecimento de Poses

No gesto Wave (Figura 10), cada posição da mão é uma parte discreta do gesto e pode ser definida como uma pose. Assim, a primeira pose (Listagem 1) deve conter as seguintes condições:

1. Mão acima do cotovelo (Linha 6, 7 e 8);
2. Mão à direita do cotovelo (Linha 10, 11 e 12);

```
1 class WaveLeftPose1 implements Pose{
2     protected WaveLeftPose1(Skeleton skeleton) {
3         super(skeleton);
4     }
5     public GestureResult checkPose(UserData user) {
6         boolean leftHandAboveHead = (skeleton.getPositionY(user,
7             JointType.LEFT_HAND) > skeleton.getPositionY(user,
8             JointType.HEAD));
9
10        boolean leftHandRightOfElbow = (skeleton.getPositionX(user,
11            JointType.LEFT_HAND) > skeleton.getPositionX(user,
12            JointType.LEFT_ELBOW));
13
14        if (leftHandAboveHead && leftHandRightOfElbow) {
15            return GestureResult.SUCCEED;
16        }
17        return GestureResult.FAIL;
18    }
19 }
```

Listagem 1- Implementação da primeira pose do gesto Wave

Da mesma forma, a segunda pose (Listagem 2) deve conter as seguintes condições:

1. Mão acima do cotovelo (Linha 6, 7 e 8);
2. Mão à esquerda do cotovelo (Linha 10, 11 e 12);

```

1 class WaveLeftPose2 implements Pose{
2     protected WaveLeftPose2(Skeleton skeleton) {
3         super(skeleton);
4     }
5     public GestureResult checkPose(UserData user) {
6         boolean leftHandAboveHead = (skeleton.getPositionY(user,
7             JointType.LEFT_HAND) > skeleton.getPositionY(user,
8             JointType.HEAD));
9
10        boolean leftHandLeftOfElbow = (skeleton.getPositionX(user,
11            JointType.LEFT_HAND) < skeleton.getPositionX(user,
12            JointType.LEFT_ELBOW));
13
14        if (leftHandAboveHead && leftHandLeftOfElbow) {
15            return GestureResult.SUCCEED;
16        }
17        return GestureResult.FAIL;
18    }
19 }

```

Listagem 2- Implementação da segunda pose do gesto Wave

O conjunto formado pela pose e o número de quadros limites é denominado KeyFrame (quadro chave). O diagrama (Figura 11) ilustra o resultado e impacto para cada retorno validado. Existem três possíveis resultados:

- FAIL – A pose atual diverge do especificado;
- PAUSING – A pose atual não obteve êxito em todas as condições especificadas, porém está dentro do limite definido;
- SUCCEED – O usuário concluiu a pose com êxito;

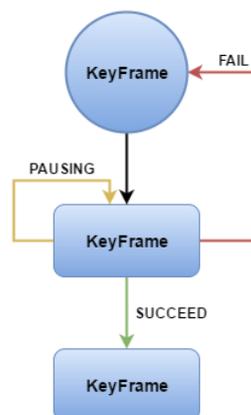


Figura 11- Representação dos resultados possíveis da análise dos KeyFrame

2.3.2.3. Reconhecimento de Gestos

Neste projeto foi utilizado a abordagem do algoritmo especializado conhecido como Keyframes (quadros chave), a Figura 12 ilustra o fluxograma da implementação do algoritmo.

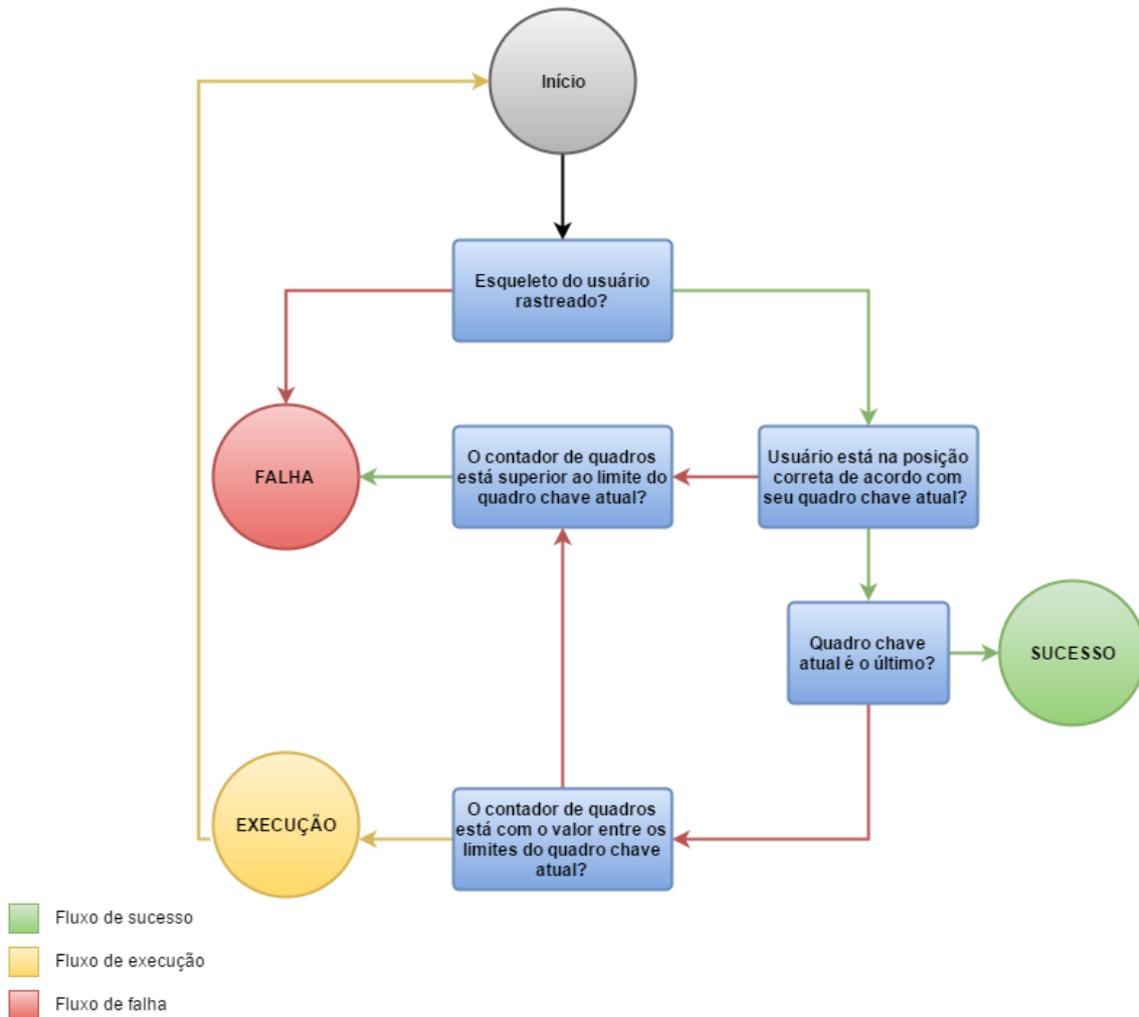


Figura 12- Fluxograma do algoritmo de reconhecimento de gestos

O algoritmo consiste em validar, caso exista um usuário rastreado, se a pose e o número de quadros limite está dentro do estipulado para o gesto em questão. Caso o usuário execute de forma correta todos os KeyFrame definidos (fluxo de sucesso descrito na Figura 12), o sistema notifica o gesto detectado e aciona um evento. A Listagem 3

demonstra o evento responsável por avançar o dado a ser visualizado no globo virtual. A Linha 10 demonstra a chamada do método responsável por avançar o tempo do cenário ambiental visualizado.

```
1 public class ForwardTimeController implements Controller {
2     private KinectView view;
3     private ScenarioController controller;
4
5     public ForwardTimeController(KinectView view) {
6         this.view = view;
7         this.controller = view.getGlobeController();
8     }
9     public void kinectActionPerformed() {
10        controller.yearForward();
11    }
12 }
```

Listagem 3- Implementação do controlador ForwardTimeController

Desta forma, utilizando o padrão de projeto *Strategy*, o módulo de controle por gestos interage com o módulo de visualização de dados espaço-temporais. O uso deste padrão possibilita a inserção de novos componentes sem grandes impactos ao sistema.

2.3.2.4. Visualização de Dados Espaço-Temporais

A ferramenta manipula dados no formato shapefile com representações poligonais. Porém, a arquitetura desenvolvida possibilita a inserção de novos componentes sem a necessidade de alterações na estrutura atual. Isto deve-se ao encapsulamento do acesso a estrutura responsável por armazenar os componentes visuais do globo virtual. Esta abstração, LayerController (Listagem 4), permite a manipulação dos componentes, tais como polígonos e anotações, a partir da criação dos layers. A Linha 2 demonstra o método responsável por inserir os layers no globo virtual; a Linha 3 demonstra o método responsável por remover os layers do globo virtual.

```

1 public interface LayerController {
2     void draw();
3     void remove();
4     default void asyncDraw() {
5         new Thread(() -> draw()).start();
6     }
7     default void asyncRemove() {
8         new Thread(() -> remove()).start();
9     }
10 }

```

Listagem 4- Interface LayerController

Os dados em shapefile possui registros referentes à geometria e à localização geográfica dos polígonos, além de outras informações relacionadas ao dado. Os registros relacionados à geometria são utilizados para desenhar os polígonos estruturados, e os registros específicos do dado são utilizados para definir a cor deste polígono. A Figura 13 ilustra o cenário a ser visualizado com inserção do dado Vegetation Scenário 2015.

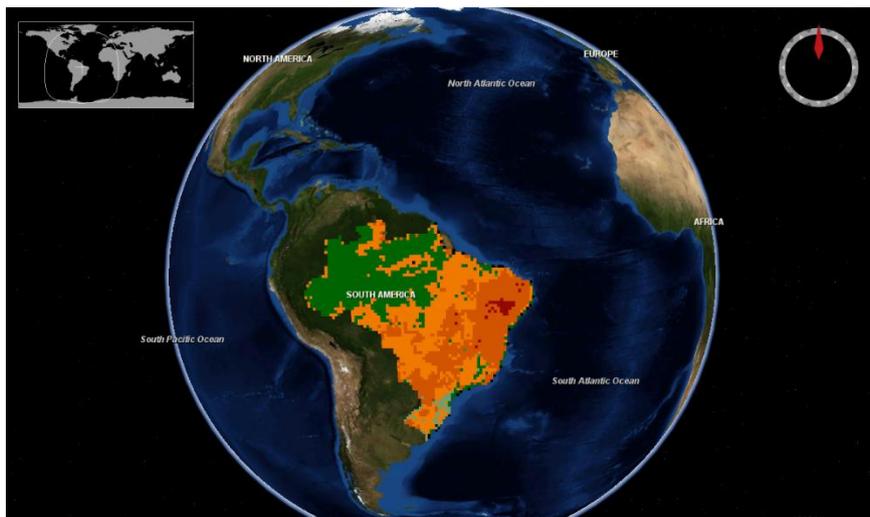


Figura 13- Visualização do dado Vegetation Scenário 2015

A distribuição de frequência é feita utilizando a API Apache Commons Math. A Listagem 5 ilustra a implementação da interface DataAnalysis, responsável por filtrar os dados e inserir os atributos a serem processados. A Linha 10 demonstra a criação do objeto Frequency, responsável por armazenar os valores de distribuição de frequência; as Linhas 15, 16 e 17 demonstram o filtro que seleciona quais registros do shapefile serão

analisados; as Linhas 18, 19 e 20 demonstram a inserção dos dados selecionados no objeto Frequency.

```
public class SimpleDataAnalysis implements DataAnalysis {
1   public Frequency createFrequency(Data data) {
2       try {
3           return getFrequency(createShapefileFromData(data),
4 data.getColumn());
5       } catch (Exception e) {
6           throw new RuntimeException(e);
7       }
8   }
9   private Frequency getFrequency(Shapefile shapefile, String column) {
10      Frequency frequency = new Frequency();
11      while (shapefile.hasNext()) {
12          ShapefileRecord record = shapefile.nextRecord();
13          record.getAttributes().getEntries()
14              .parallelStream()
15              .filter(filterByColumn ->
16                  filterByColumn.getKey()
17                      .equals(column))
18              .forEach(entry ->
19                  frequency
20                      .addValue((Comparable<?>) entry.getValue()));
21      }
22      return frequency;
23  }
24  private Shapefile createShapefileFromData(Data data) throws Exception {
25      return ShapefileController.createShapefile(data.getFilepath());
26  }
27 }
```

Listagem 5- Implementação da interface DataAnalysis

Os atributos processados são utilizados para a criação do gráfico de distribuição de frequência. Caso a análise seja realizada com apenas um dado, é exibido o gráfico de pizza (Figura 14); caso ocorra a inserção de múltiplos dados, um gráfico de barras (Figura 15) é utilizado para exibir a informação. Todos os componentes visuais utilizados são nativos da linguagem de programação Java.

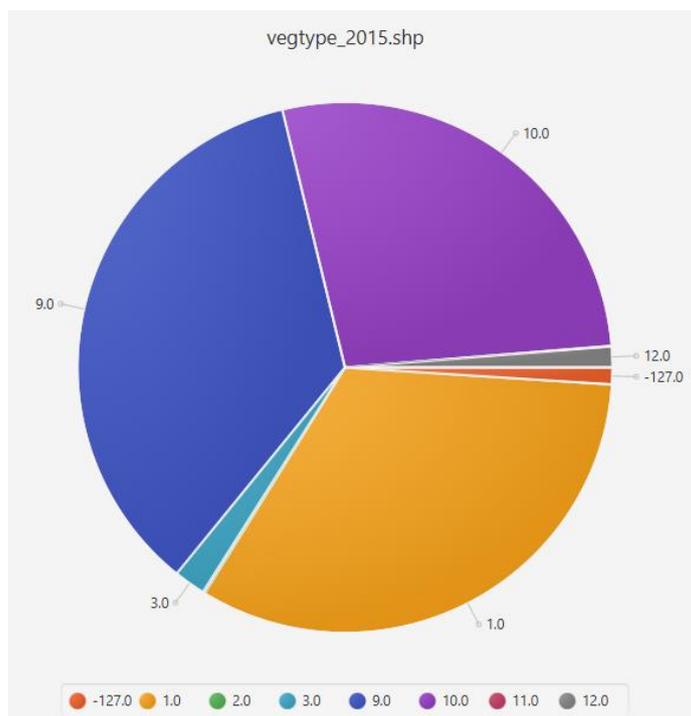


Figura 14- Gráfico de pizza criado com o dado Vegetation Scenario 2015

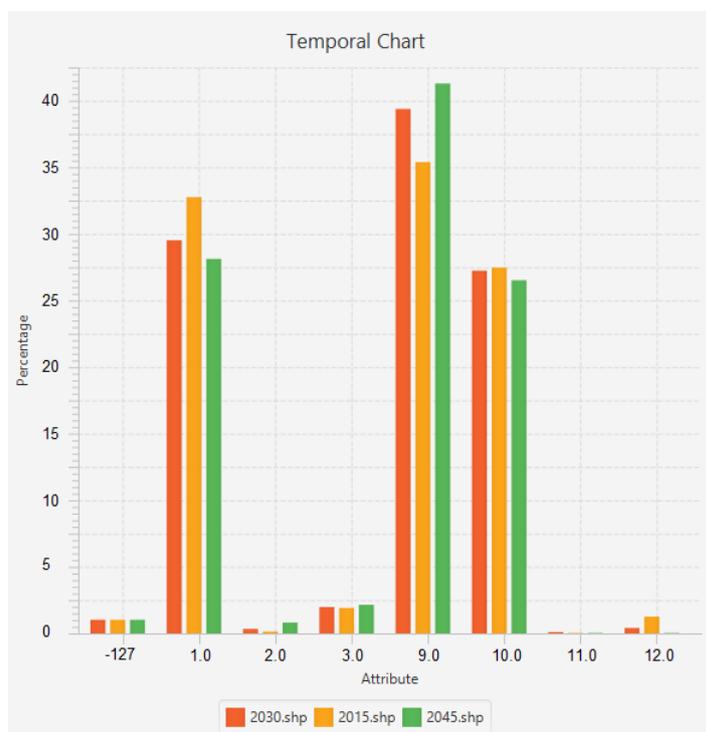


Figura 15- Gráfico de barras criado com os dados: Vegetation Scenario 2015, Vegetation Scenario 2030 e Vegetation Scenario 2045

2.3.3. Resultados

Esta seção descreve os principais resultados apresentados por este projeto. A ferramenta desenvolvida permite que o usuário interaja com o computador através do reconhecimento de comandos produzidos através de movimentos corporais. A seleção de dados e o manuseio do mapa são realizados usando apenas gestos com as mãos.

A ferramenta implementada neste projeto proporciona a interação com cenários ambientais utilizando controle por gestos. A Figura 16 demonstra a ferramenta em funcionamento; os dados visualizados são resultados de pesquisas produzidas pelo CCST, possuindo informações dos diferentes tipos de vegetação dispostos pelo Brasil. O uso de controle por gestos possibilita a interação com este subconjunto de dados em diferentes tempos.



Figura 16- Demonstração do Triangle em execução

O software desenvolvido permite o reconhecimento de gestos de maneira dinâmica. A arquitetura desenvolvida é de fácil alteração e inserção de novos movimentos. As

contribuições deste projeto são de código livre e estão disponíveis na plataforma GitHub: <https://goo.gl/QPhXZ7>. A Figura 17 apresenta a documentação online da ferramenta, também disponível no GitHub.

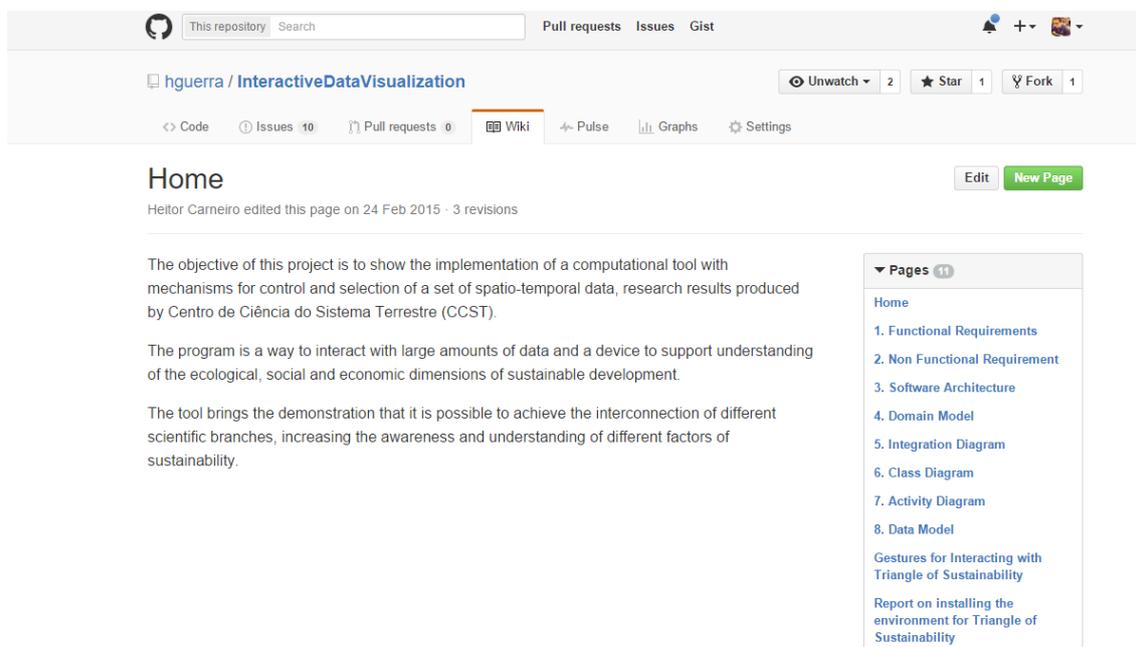


Figura 17- Documentação online do Triangle

A seleção de cenários ambientais no Triangle pode ser feita através da interface de configuração (Figura 18), esta interface possibilita selecionar os dados disponibilizados pelos pesquisadores do CCST ou qualquer dado no formato shapefile com dados poligonais. A partir da seleção do dado, é possível escolher no quadro *Scenario* (Cenário), em quais anos aquele dado será visualizado e o padrão de cores utilizado.

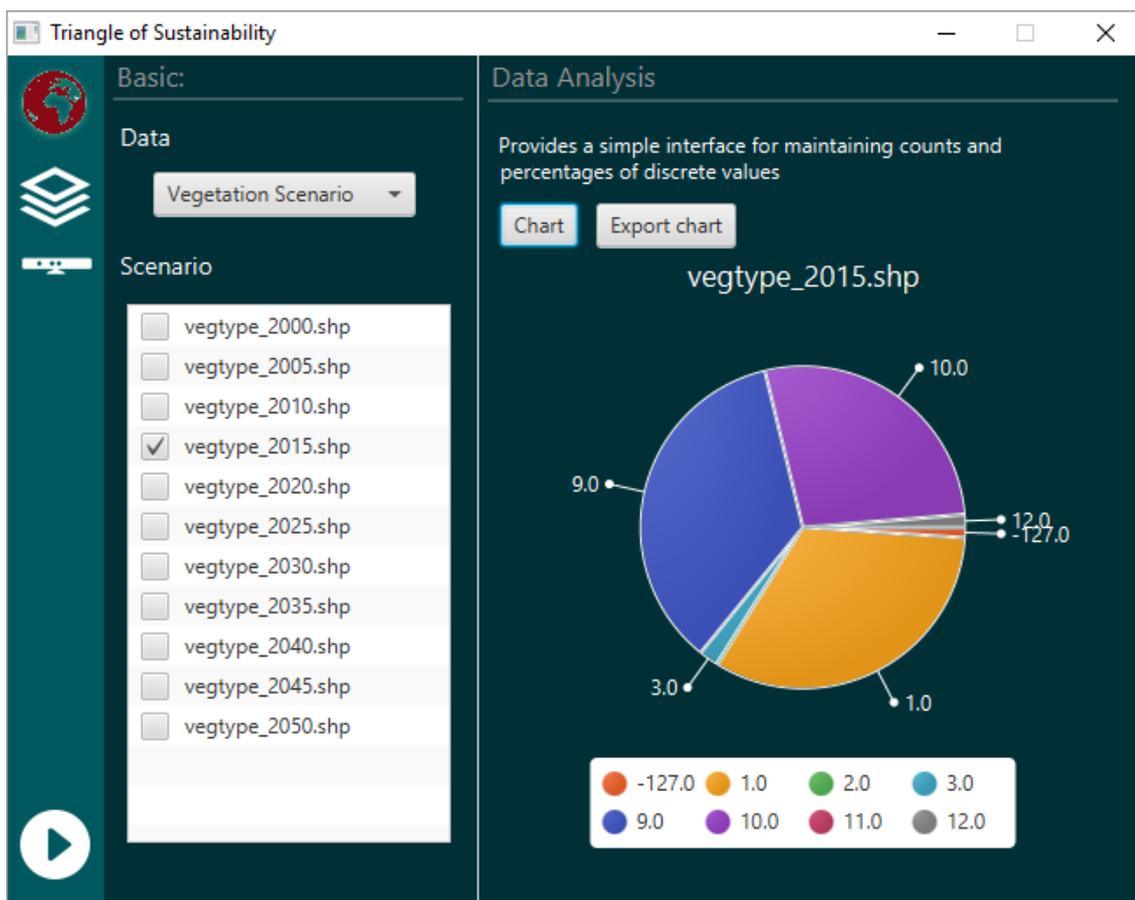
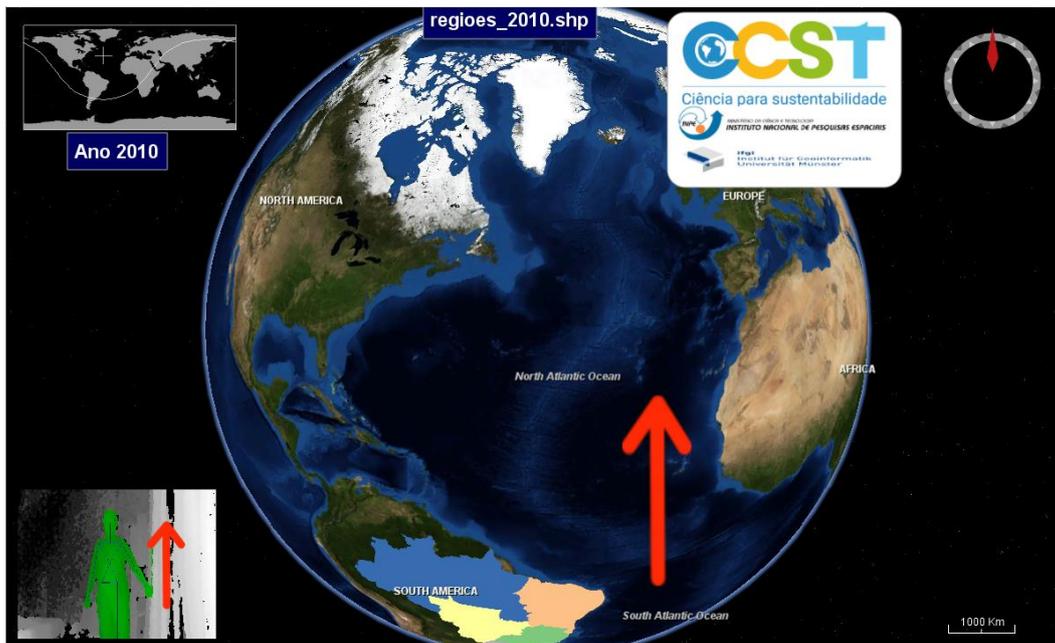
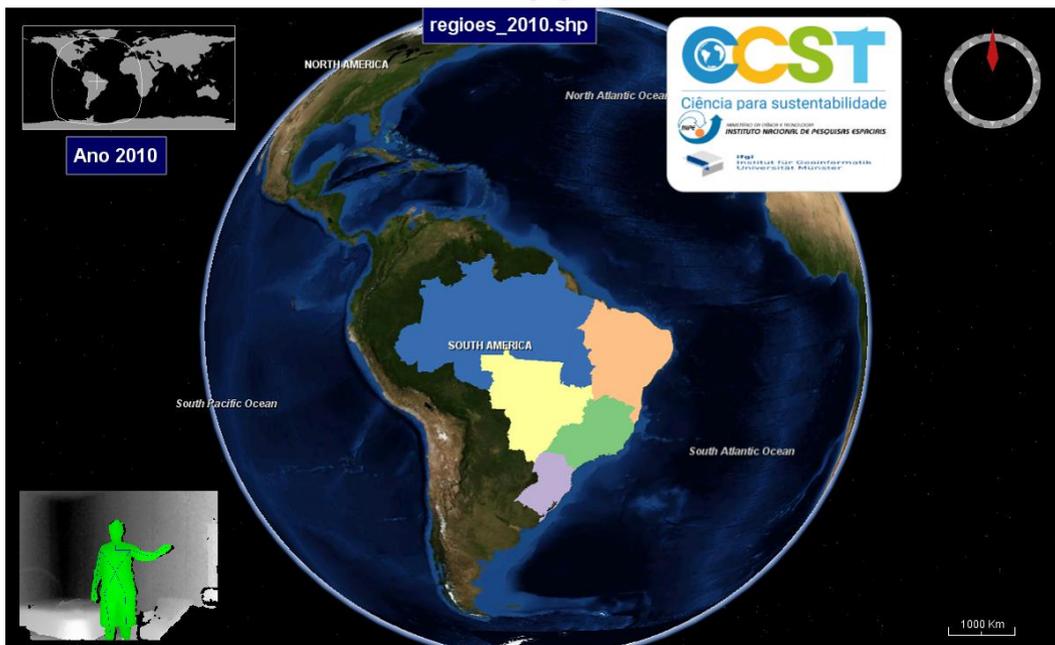


Figura 18- Interface de configuração do Triangle

A ferramenta possui vários gestos que ativam determinados comandos de visualização. Por exemplo, gesto de deslocar a mão direita esta associado ao comando de mover o mapa (Figura 19); o gesto de segurar o mapa com uma mão e deslizar na vertical com a outra esta associado a alterar os dados a serem visualizados (Figura 20a); o gesto segurar o mapa com uma mão e deslizar na horizontal com a outra esta associado a visualizar um mesmo dado em diferentes tempos (Figura 20b); o gesto de afastar e aproximar as mãos esta associado à aumentar (Figura 21a) e diminuir (Figura 21b) o zoom, respectivamente.

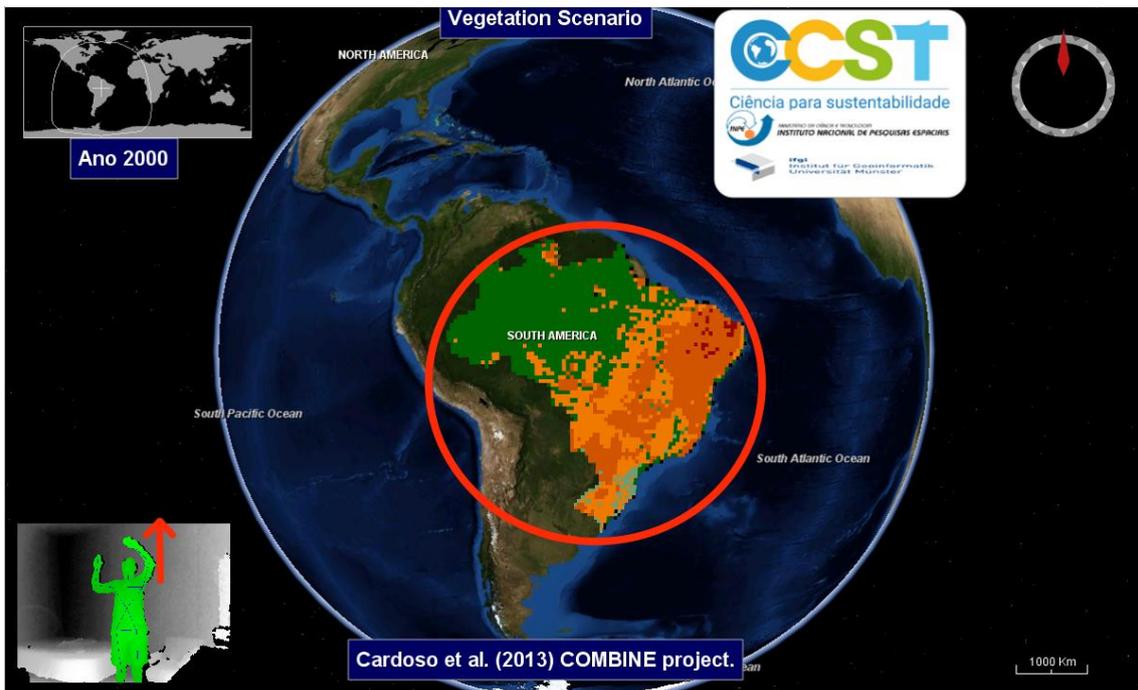


(a)

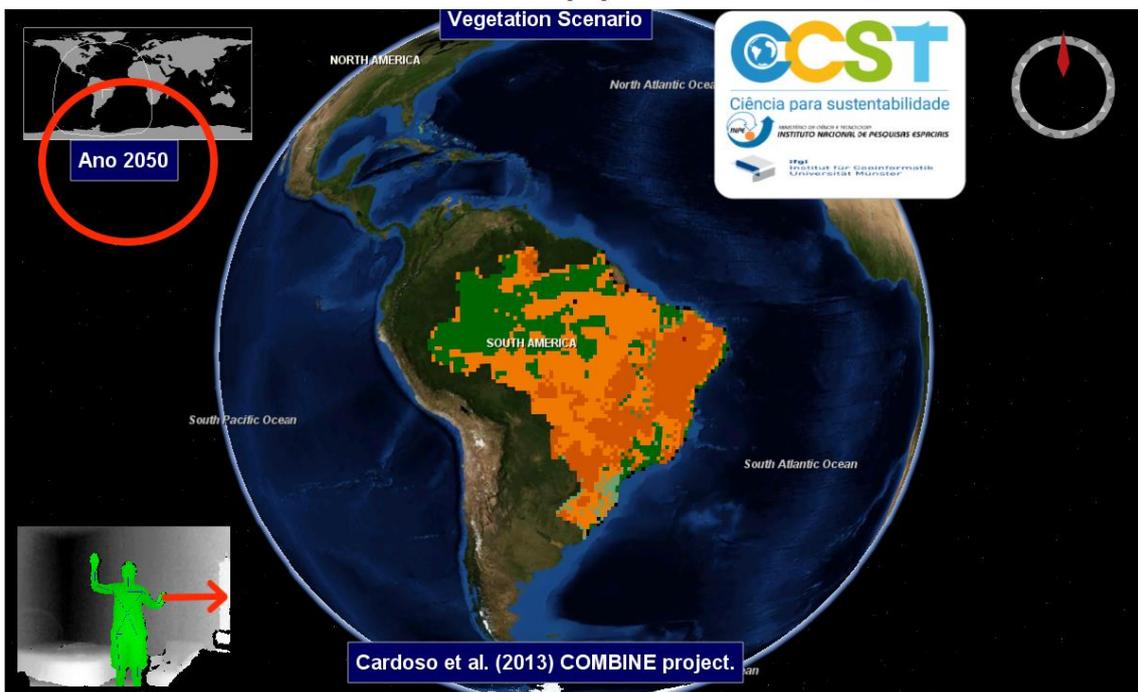


(b)

Figura 19- Execução do gesto de mover o mapa. (a) visualização antes do comando ser acionado, (b) visualização após o comando ser acionado

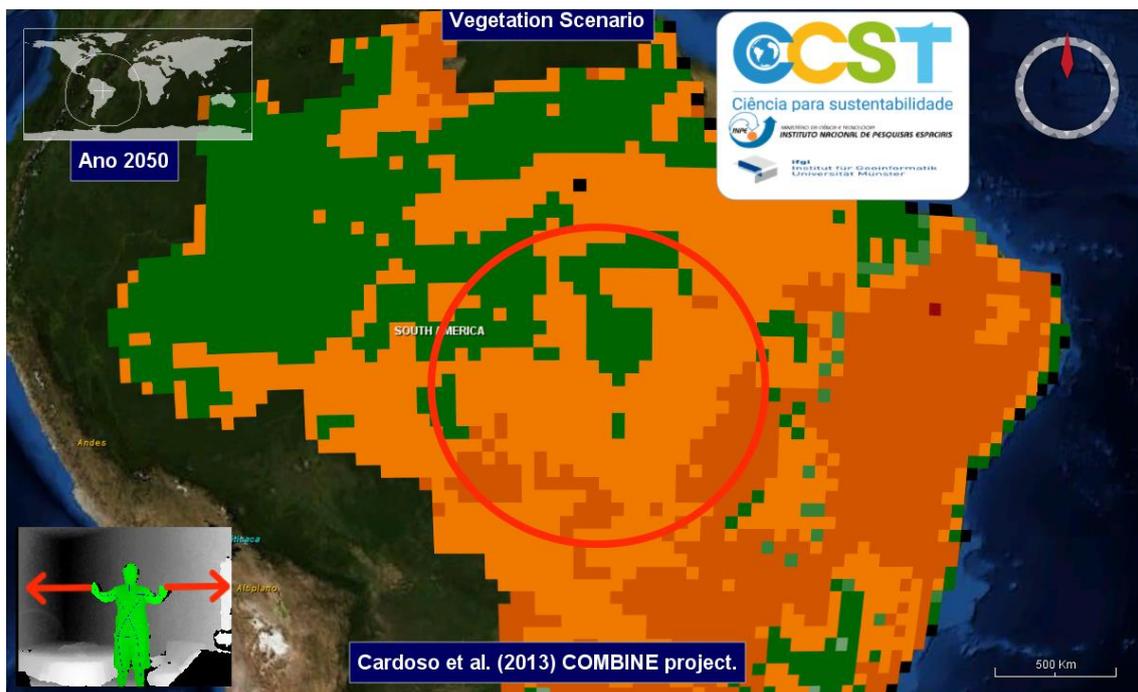


(a)

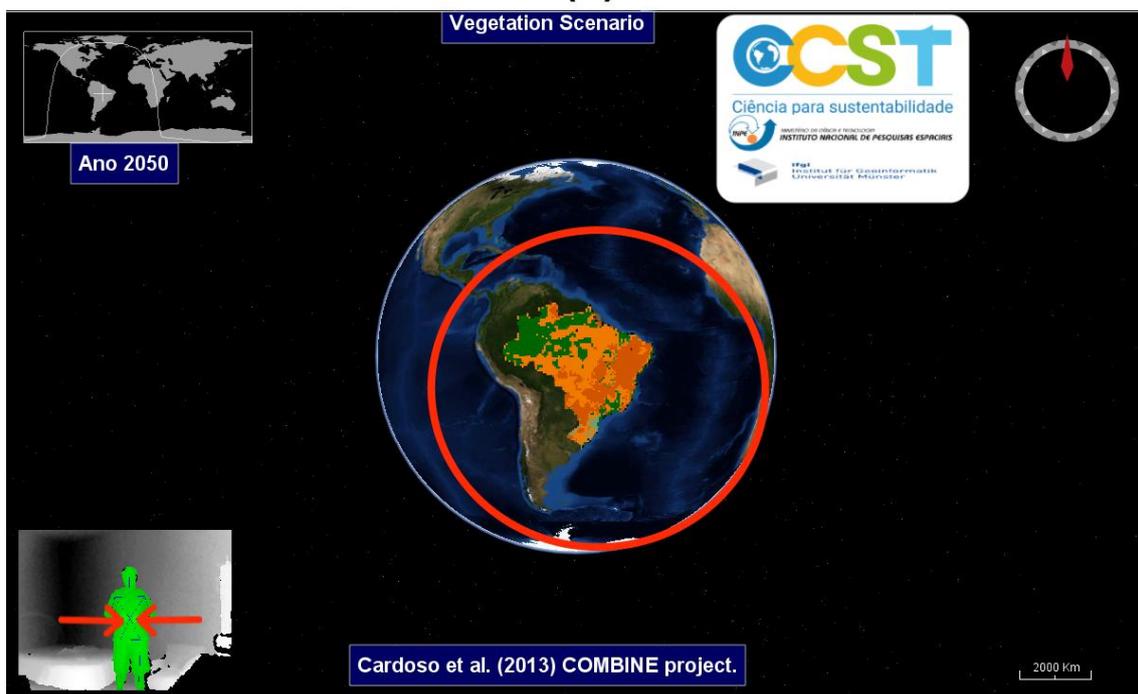


(b)

Figura 20- Execução dos gestos: (a) gesto de alterar o dado, (b) gesto de visualizar um mesmo dado em diferentes tempos



(a)



(b)

Figura 21- Execução do gesto de Zoom. (a) gesto de aproximar a visualização, (b) gesto de afastar a visualização

3 CONCLUSÃO

Este projeto apresentou o desenvolvimento de uma ferramenta de visualização interativa de dados e cenários ambientais utilizando interfaces baseadas em gestos. Foi proposto um algoritmo, para o controle por gestos, adaptável e de fácil inserção de novos movimentos. Os gestos utilizados foram inspirados em comandos multi-touch para dispositivos móveis, por isso são de fácil aprendizado. Esta ferramenta será usada por pesquisadores do INPE em eventos para divulgação de resultados dos trabalhos científicos.

3.1. Contribuições

As contribuições deste projeto são:

- Uso de interfaces naturais para o estudo de cenários complexos;
- Implementação dos gestos inspirados em comandos multi-touch;
- Implementação de um algoritmo de reconhecimento de gestos adaptável e que possibilita a fácil inserção de novos movimentos;
- Desenvolvimento da biblioteca “kinect4j” para criação de aplicações interativas utilizando o sensor Kinect;
- Disponibilidade do código fonte e documentação online da ferramenta;
- Apresentação deste projeto no Seminário de Iniciação Científica e Iniciação em Desenvolvimento Tecnológico e Inovação – SICINPE (INPE, 2015);

A partir destas contribuições pode-se concluir que:

- É possível realizar estudos de cenários ambientais complexos utilizando ferramentas interativas;
- Controle 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.

A seguir são apresentadas experiências obtidas ao longo do desenvolvimento deste projeto:

- O uso de gestos similares dificulta na detecção do movimento;
- O reconhecimento de múltiplos usuários ocasiona em ações involuntárias da ferramenta, como por exemplo, dois comandos serem acionados ao mesmo tempo, e não é recomendável.

3.2. Publicação

Um artigo foi publicado no Periódico Boletim Técnico da FATEC-SP:

CARNEIRO, Heitor Guerra; BERTOTI, Giuliano Araujo; ANDRADE, Pedro Ribeiro. VISUALIZAÇÃO INTERATIVA DE DADOS GEOGRÁFICOS UTILIZANDO O KINECT. Boletim Técnico da Faculdade de Tecnologia de São Paulo, v. 40, p. 127, 2015.

4 TRABALHOS FUTUROS

As contribuições alcançadas com este projeto não encerram as pesquisas relacionadas ao desenvolvimento de uma ferramenta para visualização interativa de dados e cenários ambientais utilizando o sensor Kinect, mas abre oportunidades para alguns trabalhos futuros. Alguns exemplos são:

- Implementar o algoritmo de reconhecimento de gestos utilizando redes neurais artificiais em busca de aumentar a acurácia;
- Desenvolver uma interface para a importação de dados espaço-temporais usando arquivos GeoTIFF, um formato muito usado como saída da execução de modelos;
- Possibilitar a criação de bancos de dados espaço-temporais e possibilitar a visualização de relatórios ou fotografias relativas aos dados visualizados;
- Fazer uma análise da usabilidade da ferramenta desenvolvida com usuários, e utilizar as informações para aperfeiçoar este projeto;
- Desenvolver uma plataforma web para visualização de dados e cenários ambientais.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, ANA PAULA DUTRA et al. Land use change emission scenarios: anticipating a forest transition process in the Brazilian Amazon?. *Global change biology*, 2015.

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.

BORENSTEIN, GREG. Making things see: 3D vision with kinect, processing, Arduino, and MakerBot. " O'Reilly Media, Inc.", 2012.

CÂMARA, GILBERTO; QUEIROZ, GILBERTO RIBEIRO DE. Banco de Dados Geográficos. Curitiba: MundoGeo, 2005.

CARDOSO, GABRIEL SCHADE. Microsoft Kinect: Crie aplicações interativas. Editora Casa do Código, 2014.

CARDOSO M.; SAMPAIO G.; SANCHES M. (2013) Assessment of additional regional feedbacks and their regional impacts in the Amazon. Deliverable D8.6 of Comprehensive Modelling of the Earth System for Better Climate Prediction and Projection (COMBINE), European Commission's 7th Framework Programme, Grant Agreement 226520.

DAVIS, B.E. GIS: a visual approach. 2nd ed. Albany: ONWORD: Thomson Learning, c2001. 438p., il. ISBN 076682764X.

ESRI, ESRI. Shapefile technical description. An ESRI White Paper, 1998.

FRANCESE, RITA; PASSERO, IGNAZIO; TORTORA, GENOVEFFA. Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI. In: Proceedings of the International Working Conference on Advanced Visual Interfaces. ACM, 2012. p. 116-123.

FREEMAN, ERIC ET AL. Head first design patterns. " O'Reilly Media, Inc.", 2004.

FUHRMANN, S., MACEACHREN, A., DOU, J., WANG, K., & COX, A. Gesture and speech-based maps to support use of GIS for crisis management: a user study. *AutoCarto* 2005.

GAMMA, ERICH. Design patterns: elements of reusable object-oriented software. Pearson Education India, 1995.

GASKINS, Tom; BROWNRIGG, Rick. NASA World Wind Java SDK 3D Earth in Your Applications and Web Pages. JavaOne, 2008.

GDAL. Geospatial Data Abstraction Library, Java bindings. Disponível em: <http://gdal.org/java/> Acesso em: 03/06/2016.

GOOGLE. Google Earth. Disponível em: <https://www.google.com/earth/> Acesso em: 08/06/2016.

GOOGLE. Google Maps. Disponível em: <https://www.google.com/maps/> Acesso em: 08/06/2016.

GOWORLWIND. goworldwind.org. Disponível em: <https://goworldwind.org/> Acesso em: 04/06/2016.

GUERRA, EDUARDO. Design Patterns com Java - Projeto Orientado a Objetos Guiado por Padrões. Casa do Código, 2012.

KRASNER, GLENN E.; POPE STEPHEN T. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of object oriented programming, v. 1, n. 3, p. 26-49, 1988.

LIU, WEIYUAN. Natural user interface-next mainstream product user interface. In: 2010 IEEE 11th International Conference on Computer-Aided Industrial Design&Conceptual Design 1. 2010.

MARTIN, R. C. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.

MATH. COMMONS. The Apache Commons Mathematics Library. Disponível em: <http://commons.apache.org/proper/commons-math/> Acesso em: 03/06/2016.

MICROSOFT CORPORATION. Human Interface Guidelines v2.0. Microsoft Corporation, 2014.

MICROSOFT CORPORATION. KinectSDK v1.8. Disponível em <https://www.microsoft.com/en-us/download> Acesso em: 03/06/2016.

MICROSOFT CORPORATION. Microsoft Surface User Experience Guidelines: Designing for Successful Touch Experiences. Microsoft Corporation, 2008.

NASA. NASA World Wind. Disponível em <http://worldwind.arc.nasa.gov/java/> Acesso em: 03/06/2016.

NINTENDO. Nintendo. Disponível em <http://www.nintendo.com> Acesso em: 03/06/2016.

NORMAN, DONALD A. Natural user interfaces are not natural. *interactions*, v. 17, n. 3, p. 6-10, 2010.

OPENNI. Open-source SDK for 3D sensors - OpenNI. Disponível em: <http://openni.ru/> Acesso em: 03/06/2016.

PTERNEAS, VANGOS. Vangos Pterneas | Kinect & Hololens. Disponível em <http://pterneas.com/> Acesso em: 06/06/2016.

WEINGART, PETER, A. ENGELS, and P. PANSEGRAU. Risks of communication: discourses on climate change in science, politics, and the mass media. *Public understanding of science* 9.3 (2000): 261-283.

WIGDOR, DANIEL; WIXON, DENNIS. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

ZHANG, ZHENGYOU. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, v. 19, n. 2, p. 4-10, 2012.

