
PROTÓTIPO DE UM SISTEMA *EYE-TRACKING* PARA AVALIAÇÃO QUANTITATIVA DE INTERFACES DE MAPAS

EVERTON BORTOLINI (1)
LUCIENE STAMATO DELARAZI (1)
DANIEL RODRIGUES DOS SANTOS (1)
SEBASTIÁN PATRICIO DUEÑAS OVIEDO (1)

(1) Universidade Federal do Paraná
Setor de Ciências da Terra
Departamento de Geomática, Curitiba - PR
evertonbortolini@hotmail.com, luciene@ufpr.br, danielsantos@ufpr.br, sduenasoviedo@gmail.com

Este trabalho tem por objetivo apresentar o desenvolvimento de um protótipo para *eye-tracking*, com suporte do sensor comercial *Kinect*. O *eye-tracking* consiste em uma ferramenta para registrar os movimentos dos olhos, assim, possibilitando acompanhar o comportamento visual do usuário ao interagir com uma interface. É importante a avaliação de interfaces de forma a propor correções e otimização nas mesmas, sendo aplicável a qualquer situação em que haja interação entre um usuário e determinado ambiente, seja ele virtual ou analógico. A aplicação do *eye-tracking* se dará, no caso deste projeto, na avaliação de interfaces de mapas, para o qual é necessário ter conhecimento prévio na área de Interação Humano-Computador (IHC) para entendimento quanto aos métodos avaliativos. O uso da ferramenta *eye-tracking* possibilita fazer uma avaliação quantitativa das interfaces de mapas, ao contrário dos métodos tradicionais, nos quais predominam avaliações qualitativas. Os métodos de avaliação quantitativos permitem conhecer numericamente o comportamento de determinado usuário ao observar e tentar compreender um mapa, assim, torna-se um método mais confiável para a extração de informações e tomada de decisões. Em consideração ao desenvolvimento do protótipo, o método elaborado tem por fundamento básico extrair automaticamente as pupilas de um usuário e registrar seus movimentos em tempo real. O desenvolvimento do protótipo *eye-tracking* está sustentado em linguagem de programação C++, compilador *Microsoft Visual Studio 2008*, integrado com as bibliotecas *Open Source Computer Vision (OpenCV)* e *Microsoft Face Tracking Software Development Kit for Kinect for Windows (Face Tracking SDK)*. A escolha do sensor *Kinect* é justificada, principalmente, pelo baixo custo em comparação a um *eye-tracking* comercial, somada à possibilidade de programação para a criação de aplicações diversas. O método desenvolvido é dividido em seis etapas: 1) *Extração automática do rosto do operador* – neste caso, usa-se uma função existente no OpenCV; 2) *Extração automática do par de olhos do operador* – também usa-se uma função disponível no OpenCV, sendo os olhos automaticamente separados em direito e esquerdo; 3) *Segmentação do objeto de interesse (pupila do usuário)* – neste caso, emprega-se funções de suavização pelo cálculo da mediana da vizinhança e a limiarização para identificação da pupila, além da função *Canny* para detecção de bordas, sendo, ao final do processo, definidos os olhos do usuário pelos pixels de borda; 4) *Ajustamento dos pixels de borda que descrevem a pupila extraída* – aplica-se um modelo matemático, como o da equação da elipse e as observações advindas dos pixels de borda da pupila extraída, são utilizados para ajustar as observações pelo Método dos Mínimos Quadrados (MMQ); 5) *Transformar os pixels de borda das pupilas, esquerda e direita, automaticamente extraídas e ajustadas para a tela do computador* – aqui, um modelo de mapeamento polinomial é aplicado empregando como observações os pixels de bordas ajustados e, como controle, pontos randomicamente gerados e visíveis na tela do computador. O MMQ é novamente empregado, porém, neste caso, para estimar os parâmetros da equação polinomial. Posteriormente, os pixels de borda são transformados para o sistema de coordenadas da tela do computador; 6) *Avaliação da interface* – com o uso do protótipo pretende-se avaliar, com vários usuários, as interfaces de mapas e aplicativos que contém os mesmos. No presente estágio da pesquisa, estão sendo realizados ensaios com uma *webcam*, sendo a integração com o sensor *Kinect* o objetivo a ser perseguido posteriormente.