

ETRAN: расширяемое программное обеспечение для визуализации и анализа данных видеоокулографии

Мармалюк П.А., кандидат технических наук, заведующий лабораторией математической психологии и прикладного программного обеспечения Центра информационных технологий для психологических исследований ГБОУ ВПО МГППУ, Москва, Россия, pavel.marmalyuk@gmail.com

Юрьев Г.А., кандидат физико-математических наук, доцент факультета информационных технологий ГБОУ ВПО МГППУ, Москва, Россия, g.a.yuryev@gmail.com

Жегалло А.В., кандидат психологических наук, старший научный сотрудник Центра экспериментальной психологии ГБОУ ВПО МГППУ, Москва, Россия, zhegs@mail.ru

Поляков Б.Ю., студент магистратуры факультета информационных технологий ГБОУ ВПО МГППУ, Москва, Россия, deslion@yandex.ru

Панфилова А.С., кандидат технических наук, заведующий лабораторией количественной психологии Центра информационных технологий для психологических исследований ГБОУ ВПО МГППУ, Москва, Россия, panfilova87@gmail.com

Аннотация

Представлены детали программной реализации бесплатно распространяемой и расширяемой информационно-аналитической системы с открытым исходным кодом, предназначенной для анализа данных видеоокулографии. На концептуальном уровне описаны объекты ядра системы, разделяемые на три основные группы: классы базовых сущностей (эксперимент, испытуемый, испытание и т.д.), классы данных, присваиваемых сущностям, и классы операторов обработки данных. Рассмотрены основные методы и функции ядра системы, которые, используя экземпляры операторов обработки данных, решают задачи загрузки данных айтрекинга в систему, их анализа (фильтрации, сглаживания, детекции событий, оценки показателей событий и прочие) и визуализации. Кратко описан макет графического пользовательского интерфейса системы.

Ключевые слова: айтрекинг, статистическое ПО, детекция и анализ окуломоторных событий, язык программирования R, библиотека для разработки веб-приложений Shiny

ETRAN: scalable software for visualization and analysis of eye tracking data

Marmalyuk P.A., PhD (Computer Science), Head of the Laboratory of Mathematical Psychology and Applied Software of the Centre of Information Technologies for Psychological Studies, Moscow State University of Psychology and Education, Moscow, Russia, pavel.marmalyuk@gmail.com

Yuryev G.A., PhD (Physics and Mathematics), Assistant Professor, Department of Computer Science, Moscow State University of Psychology and Education, Moscow, Russia, g.a.yuryev@gmail.com

Zhegallo A.V., PhD (Psychology), Senior Researcher, Centre of Experimental Psychology, Moscow State University of Psychology and Education, Moscow, Russia, zhegs@mail.ru

Polyakov B.Yu., MA student, department of information technologies of Moscow State University of Psychology and Education, Moscow, Russia, deslion@yandex.ru

Panfilova A.S., PhD (Computer Science), Head of the Laboratory of Quantitative Psychology of the Centre of Information Technologies for Psychological Studies, Moscow State University of Psychology and Education, Moscow, Russia, panfilova87@gmail.com

Abstract

Implementation details of free, scalable and open source software system designed for eye tracking data analysis are presented. Core system objects, which are split into three main groups:

base entities classes (an experiment, a subject, a trial, etc.), classes of data, assigned to entities, and data processing operators classes, - are reviewed conceptually. Examined are methods and functions of the system core that, using instances of data processing operators, address the gaze data import, data analysis (filtering, smoothing, oculomotor events detection, estimation of events' characteristics and others) and visualization issues. A draft of graphical user interface is described shortly.

Keywords: eye tracking, statistical software, oculomotor events detection, R programming language, Shiny web application framework

Введение

Статья посвящена описанию классов базовых объектов и методов ядра бесплатно распространяемой и расширяемой программной системы с открытым исходным кодом, предназначенной для анализа данных видеоокулографии и получившей название **ETRAN** (**E**ye **T**racking **R**esults **A**nalys**e**r). Разрабатываемая система является воплощением недавно предложенной авторами концепции реализации информационно-аналитических систем, применяемых в области анализа данных видеоокулографии (Мармалюк и др., 2015; Юрьев, Панфилова, Мармалюк, 2015; Zhegallo & Marmalyuk, 2015). Первый полноценный прототип системы планируется создать к концу 2015 года. Предполагается, что система будет востребована широким кругом специалистов в рамках фундаментальных и прикладных исследований, в том числе поисковых, в области экспериментальной и дифференциальной психологии, маркетинга, эргономики, педагогики, лингвистики, и других направлений, в которых используется метод видеоокулографии.

Система разрабатывается в рамках парадигм объектно-ориентированного и функционального программирования. В объектно-ориентированной парадигме основными концепциями являются понятия объектов, классов объектов и методов работы с ними, что позволяет естественным образом представить специфику области знаний, определить основные классы объектов, такие как, например, траектория взора или пространственная область интереса, которым присущи определённые свойства и методы обработки. Такой подход обеспечивает структурированность, модульность и прозрачность программного кода, что облегчает последующие модификации и расширение реализованной системы, то есть делает процесс его сопровождения и модификации более доступным, в частности для сторонних разработчиков. Сохранение функциональной стилистики создания программ в рамках среды R делает систему более простой для конечного пользователя и предъявляет меньшие требования к уровню его знаний в области программной инженерии.

Ни одно из существующих программных решений в области анализа данных окулографического эксперимента не является законченным с точки зрения пригодности для научно-исследовательской работы. Это объясняется тем, что, несмотря на известный типичный сценарий работы специалиста при обработке экспериментальных данных, на каждом из этапов сценария спектр источников первичных данных (текстовые файлы с разделителями, xml-файлы, базы данных и т.п.) и способов их обработки слишком широк для возможности полной их программной реализации. Например, разновидностей алгоритмов обнаружения (детекции) окуломоторных событий на сегодняшний день более десяти наименований, а если учесть различные их эвристические модификации, то получится ещё более внушительное число. Очевидно, что силами одного-двух программистов невозможно за разумные сроки создать универсальную систему, включающую большинство способов обработки и учитывающую все важные нюансы. В основном по этой причине во многих прикладных системах используется модульная архитектура, облегчающая расширение функциональности путём либо прямого добавления новых алгоритмов в программный код (что затруднительно сделать без участия разработчиков системы), либо подключения так называемых плагинов (plug-ins) посредством использования заранее подготовленного разработчиками системы

интерфейса. При этом можно рассматривать компоненты базовой версии системы, предоставляемой разработчиками, как ядро, включающее определения классов объектов, очевидно необходимых для представления структуры эксперимента, его участников, используемых стимулов, зарегистрированных данных и типичных результатов их обработки, а также основные методы, позволяющие полноценно работать с объектами определённых в системе классов: методы их создания, модификации, удаления и выполнения тех или иных базовых видов анализа.

Особое внимание уделено особенностям структуры классов ядра системы, конкретным методам обработки данных и используемым в их составе алгоритмам. Однако статья будет интересна не только разработчикам программного обеспечения, но и потенциальному кругу пользователей, не занимающихся вопросами проектирования и разработки ПО, поскольку при описании компонент системы авторы не углубляются в излишние технические детали, а скорее представляют наиболее важные для взыскательного пользователя аспекты. Также в статье затрагиваются некоторые методологические нюансы, на которые должен обращать внимание каждый уважающий свой труд исследователь, например, важность этапа предобработки данных, необходимость учёта и анализа влияния различных факторов на конечный результат (в том числе настроек алгоритмов), надёжность оценки распределения визуального внимания по областям интереса и другие.

В первом разделе статьи представлено описание классов объектов ядра системы. Второй раздел посвящён описанию программно реализованных методов (функций) и алгоритмов анализа данных, находящихся на стадии тестирования и отладки. Третий раздел посвящён особенностям реализации пользовательского интерфейса системы. Четвёртый раздел раскрывает дальнейшие планы разработчиков по совершенствованию и расширению функций системы. В конце статьи делаются основные выводы и обобщаются результаты текущей работы.

Классы объектов ядра системы

В данном разделе представлено описание классов объектов ядра системы, которые можно разделить на три основные группы: классы базовых сущностей (например, эксперимент, испытуемый, испытание), классы данных, присваиваемых сущностям, и классы операторов обработки данных.

Классы сущностей

Класс “Experiment” предназначен для представления основной информации об эксперименте и привязки к нему различных релевантных сущностей (испытуемых, стимулов, испытаний) с использованием уникального номера эксперимента, присваиваемого системой при его создании и добавлении в список экспериментов (*класс “Эксперименты”*). Таким образом, в системе могут одновременно храниться и обрабатываться данные различных экспериментов, что немаловажно при необходимости сравнения различных характеристик ГДА, полученных, например, в рамках лонгитюдных исследований. На данный момент в системе можно задавать название и расширенное описание эксперимента, дату его проведения, а также ФИО или текстовые идентификаторы экспериментаторов, проводивших эксперимент. Данная информация необходима для облегчения поиска данных, хранящихся в системе.

Эксперимент состоит из испытаний, для описания которых существует *класс “Trial”*. Данный класс предназначен для представления данных об испытании, проводимом в рамках конкретного эксперимента, таких как название и описание, а также порядковый номер его проведения в ходе эксперимента, который важно учитывать в силу возможного влияния на испытуемых таких факторов, как утомляемость или порядок предъявления стимулов, используемых в испытаниях. Испытания, проводимые в рамках конкретного эксперимента, определенного в системе, привязываются к нему по

уникальному номеру эксперимента. При этом испытания всех экспериментов в системе хранятся в рамках единого списка испытаний (*класс “Испытания”*). Привычное для экспериментаторов распределение испытаний по блокам и сессиям может осуществляться путём привязки к испытаниям специальных факторов, определяющих номер блока испытаний и номер экспериментальной сессии. Подробнее о факторах будет изложено ниже.

Информация об испытуемых, принимающих участие в экспериментах, отражается в системе с помощью объектов *класса “Subject”* и хранится в списке испытуемых (*класс “Испытуемые”*). Могут быть сохранены такие данные как ФИО и дата рождения каждого испытуемого. На основе даты рождения и даты проведения эксперимента всегда возможно восстановить возраст испытуемого на момент участия в эксперименте. Для идентификации конкретного человека может быть указан уникальный текстовый код, с использованием которого можно впоследствии задавать значения многочисленных факторов испытуемого таких как, например, пол, уровень тревожности, результаты по тесту интеллекта и т.п.

В ходе эксперимента испытуемым с помощью стационарного айтрекера («башенного» или дистанционного типа) обычно предъявляются тексты, статические и динамические графические объекты (например, изображения лиц, видеоролики). С помощью мобильного айтрекера, который обычно крепится на голове испытуемого и включает в свой состав фронтальную видеокамеру, регистрация ГДА проводится в условиях наблюдения некоторой сцены реального мира (например, при езде за рулём автомобиля или в процессе выбора товаров в магазине). В разрабатываемой системе то, что наблюдается испытуемым, называется стимулом и описывается с помощью объектов *класса “Stimulus”*. Набор стимулов, используемых в экспериментальной работе, хранится в списке стимулов (*класс “Стимулы”*), в котором каждому стимулу ставится в соответствие уникальный идентификатор, что позволяет привязывать к стимулам значения числовых и нечисловых факторов, а также снижает избыточность информации в случаях, когда один и тот же стимул многократно используется в различных экспериментах. Стимулы при этом могут быть именованы и подробно описаны в комментариях для облегчения поиска.

В силу вариативности стимульного материала в системе необходимо указывать тип стимула: это либо «изображение» (тексты, как правило, сводятся к изображениям), либо «видео», либо «наблюдаемая сцена». В зависимости от типа стимула варьируется и список других его свойств, которые необходимо указывать. Если используется стимул типа «изображение» или «видео», то целесообразно также хранить путь к файлу, содержащему соответствующие данные. Данные стимула могут быть использованы в системе как для визуализации результатов измерений (в основном, при наложении траектории взора на стимул), так и для анализа характеристик стимула (например, для поиска «особых» точек изображения, которые могут рассматриваться как аттракторы зрительного внимания). Для изображений предусмотрено хранение размерностей изображения (количества пикселей по высоте и ширине, как исходные, так и используемые при экспозиции), которые при известных геометрических размерах экрана стимульного монитора и расстоянии до него могут использоваться для расчёта угловых размеров стимула или пространственных областей интереса, о которых будет сказано ниже. Для видеороликов и наблюдаемых сцен предусмотрены также такие свойства, как размерность кадра, длительность записи и число кадров, учитываемые при работе с динамическими пространственными областями интереса.

В ходе исследований с использованием результатов видеоокулографии часто необходимо определить, как внимание пользователя распределено между конкретными областями стимульного изображения или наблюдаемой сцены. Такое разделение, в общем случае, основывается на неформализуемых посылах и гипотезах, известных автору исследования. Задача разработчика программного обеспечения в подобной ситуации

заключается в том, чтобы предоставить пользователю максимально полный набор инструментов выделения так называемых областей интереса, описываемых в системе с помощью класса *“Область интереса”* и объединяемых в наборы областей интереса.

Разбиение исходного изображения на набор областей может выполняться множеством способов, в частности, одним из простейших методов является наложение на исходный стимул регулярной сетки. Наложение регулярной сетки хоть и является универсальным (независимым от стимула) способом, подходящим для разведочного анализа, но покрывает лишь незначительную часть потенциально востребованных вариантов разбиения. Как правило, руководствуясь чётко сформулированными гипотезами, экспериментатор выделяет на нём лишь несколько областей произвольной формы. В этом случае необходим обширный набор из стандартных геометрических примитивов (треугольник, прямоугольник, эллипс, круг), а для более сложных случаев – возможность выделения необходимой области при помощи многогранника. Хотя значительное количество геометрических примитивов и являются частными случаями многогранника, выделение их в отдельную палитру инструментов разметки упрощает задачу подготовки стимула для экспериментатора и часто ускоряет следующий за сбором данных процесс математической обработки материала.

Кроме этих стандартных инструментов в некоторых случаях целесообразно формировать так называемые нечёткие зоны интереса. Это области, при использовании которых оценка принадлежности конкретной точки траектории взора (или точки фиксации) области стимула делается не детерминированно (взор попал или не попал), а с некоторым весовым коэффициентом, который обычно можно интерпретировать как степень уверенности в установленном состоянии. Например, испытуемому демонстрируется изображение лица, а по зарегистрированным данным требуется установить количество фиксаций взора в области, соответствующей расположению губ. При определённом масштабе и с учётом аппаратных ошибок регистрации пользователь не может с уверенностью провести чёткую границу для выделения этой области. В этом случае можно сопоставить исследуемому фрагменту изображения некоторую весовую функцию, которая будет относить точки траектории к данной области с большей или меньшей степенью уверенности. Например, по мере удаления от центра соответствующей области правильным образом «настроенная» двумерная функция нормального распределения будет принимать всё меньшие значения. Итогом такой оценки будет не количество всех попаданий в жёстко заданную область, а вероятностный показатель предпочтения данной области.

Легко также представить случай, когда область интереса оказывается распределённой: исследователь демонстрирует изображение материнской платы ПК и заинтересован в том, чтобы исследовать распределение внимания испытуемых по областям, связанным с цепями электропитания, с микроконтроллерами, с системами охлаждения и другими. Очевидно, что каждая из таких областей может быть распределена по изображению и представлять собой набор отдельных «островков», форма которых также может быть различной. Несмотря на такое пространственное разделение, содержательно, на уровне замысла исследователя, это одна и та же область и данные, попадающие в различные её участки, должны обрабатываться соответственно.

Наконец, любой из описанных подходов к фрагментации изображения может применяться в случае динамических стимулов. Это необходимо, например, когда стимулом является видеозапись: пространственные и геометрические характеристики зоны могут меняться от кадра к кадру.

Таким образом, для пользователя элементарной составляющей разбиения является область интереса, основными свойствами которой являются имя, тип (форма, наличие временной динамики изменения) и параметры, определяющие её размеры и расположение. Так, области, внесённые с одинаковыми именами в набор, рассматриваются как распределённые. Области интереса объединяются пользователем в

группы, сопоставляемые отдельному стимулу в рамках конкретного испытания, т.е. устанавливается соответствие между группой областей интереса и экспериментальной ситуацией, а не стимулом.

Основным и самым "массивным" объектом, с которым ведётся наиболее активное взаимодействие системы при проведении анализа данных, является объект *класса "Выборка данных"*. Данный класс предназначен для представления в системе списка записей, или так называемых *комплексных наблюдений*, относящихся к классу *"Запись данных"*, каждое из которых включает в себя как «сырые» данные, полученные в результате регистрации ГДА (моменты времени, временные ряды координат взора, координат центра зрачка, размеров зрачка и т.п.), так и результаты их математической обработки. Результатами обработки могут быть параметры обнаруженных окуломоторных событий (например, амплитуды всех саккад), различные одномерные показатели (например, длительность записи, количество фиксаций в испытании, номер кластера, к которому относится комплексное наблюдение), а также многомерные результаты анализа (например, матрица оценок вероятностей переходов между областями интереса или вектор длительностей пребывания в них). Подразумевается, что комплексное наблюдение – это результат регистрации ГДА, полученный в конкретном испытании конкретного эксперимента для конкретного испытуемого (то есть тройка «эксперимент-испытуемый-испытание» однозначно идентифицирует наблюдение) и результат их последующей обработки различными методами. Комплексное наблюдение включает в себя объект *класса "Данные глаз"*, объект *класса "Данные событий"*, объект *класса "Последовательность областей интереса"*, объект *класса "Вектор показателей областей интереса"*, объект *класса "Матрица показателей областей интереса"* и список одномерных статистик, вычисляемых по данным указанных объектов. Подробнее об указанных классах будет сказано в подразделе **«Классы данных комплексного наблюдения»**

В рамках любого экспериментального исследования фиксируются значения различных числовых и нечисловых переменных, относящихся к испытуемым, испытаниям, стимулам и другим объектам, целенаправленно контролируемых в соответствии с дизайном эксперимента, либо неконтролируемых, но имеющих прямое или косвенное отношение к изучаемым явлениям. Эти переменные в различных контекстах могут рассматриваться как зависимые или независимые факторы, чьи значения необходимо вводить в систему для проведения дальнейшего статистического анализа гипотетических взаимосвязей с характеристиками ГДА, для группировки наблюдений с целью облегчения поиска или визуального анализа групповых особенностей, для кластеризации наблюдений. Информация о факторах представляется в системе с помощью объектов *класса "Фактор"*, которые сводятся в *список доступных факторов*.

В системе представляются следующие сведения о факторе: его название, словесное описание, тип (номинальный, порядковый, целочисленный, вещественный), а также названия уровней фактора (в случае номинального или порядкового фактора) и указание владельца фактора. Под владельцем фактора понимается целый класс объектов, к которому относится данный фактор: им может быть «испытуемый», «испытание», «стимул», «запись данных» или даже «окуломоторное событие» (список возможных владельцев может быть расширен). После того как некоторый фактор внесён в список допустимых факторов, становится возможным привязывать известные значения этого фактора к конкретным объектам посредством добавления уникального номера фактора, ссылки на объект и самого значения фактора (с проверкой на корректность) в таблицу значений факторов (*класс "Данные факторов"*). Стоит заметить, что таких таблиц в системе несколько: одна предназначена для хранения внешних по отношению к системе факторов (загружаемых извне факторов испытуемых, испытаний, стимулов), другая – для хранения набора вычисляемых одномерных показателей различных объектов, а остальные

– по две на наблюдение (для левого и правого глаза) – для хранения показателей окуломоторных событий.

Базовые классы основных операторов

В этом подразделе описываются классы объектов, предназначенных для представления в системе различных операторов загрузки и обработки исходных данных айтрекинга и производных результатов.

Разрабатываемая система не взаимодействует с оборудованием для регистрации ГДА, основное её предназначение – математическая обработка зарегистрированных данных, загружаемых из различных внешних источников данных. Такими источниками могут быть, например, текстовые файлы с разделителями, xml-файлы, базы данных, бинарные файлы данных и т.д. Чтобы система могла получать данные из указанных источников, необходимо, чтобы в неё были заложены соответствующие функции, существенно варьирующиеся в зависимости от типа источника. В связи с этим в системе используется модульный подход, при котором в базовой версии системы доступен лишь стандартный загрузчик, позволяющий загружать данные из текстовых файлов с разделителями, достаточно универсальный, чтобы быть пригодным в 80-95% типовых ситуаций.

Для представления алгоритма загрузки данных окулографии реализован специальный класс *“Загрузчик данных”*. Объект этого класса включает название загрузчика для его идентификации, функцию загрузчика, написанную на языке R, реализующую конкретный алгоритм загрузки сырых окулографических данных и в результате своего выполнения возвращающую объект класса *“Запись данных”*, а также список произвольных настроек, необходимых для корректной работы функции.

Таким образом, пользователю системы предоставляется возможность подключать собственные реализации алгоритмов загрузки данных без необходимости встраивать соответствующий программный код в ядро системы (однако, необходимость программировать функцию самого загрузчика, очевидно, не отпадает). На данный момент возможность подключения внешнего модуля загрузки данных находится на этапе разработки, а используемая по умолчанию функция загрузки данных из текстовых файлов с разделителями описана в следующем разделе **«Основные методы и функции ядра системы»**.

Любым измерительным системам, а особенно системам видеоокулографии, основанных на применении методов распознавания образов, свойственно наличие ошибок измерения (регистрации). В процессе регистрации ГДА факторы, влияющие на возникновение шумов и искажений в данных многочисленны: аппаратный шум, очки и контактные линзы, особенности макияжа, нарушения зрения, естественные и искусственные источники света, механические воздействия на айтрекер (тряска, удары), моргания и прищуривания и многие другие. В связи с этим общепринятой практикой является предварительный визуальный анализ и предобработка зарегистрированных «сырых» данных. Обычно это делается с целью обнаружения и фильтрации участков записи, которые нежелательно учитывать при обработке данных в силу снижения надёжности результатов анализа, а также с целью определения необходимой степени сглаживания временных рядов для получения корректных результатов некоторыми неустойчивыми к шумам алгоритмами. Записи с высокой долей «испорченных» отсчётов (от 60-70%) обычно исключают из последующего анализа. При этом ручное обнаружение ненадёжных участков записи – чрезвычайно трудоёмкое и к тому же субъективное занятие. В связи с этим, как правило, используются простые критерии для исключения нежелательных участков записи, которые легко автоматизируются: равенство нулю обеих координат взора (свидетельствующие, как правило, о невозможности детекции зрачка и других признаков глаза аппаратно-программной частью айтрекера), а также выход

координат взора за пределы размеров стимульной области. Такие участки обычно помечаются как «пропуски» данных.

Для борьбы с шумами устройства и случайными артефактами (например, кратковременными «случайными» выбросами, приводящими к экстремальным значениям моментальной скорости и ускорения), используют сглаживание данных, осуществляемое с помощью различных алгоритмов.

Учитывая изложенное выше, в системе реализованы класс *“Фильтр данных”* и класс *“Оператор сглаживания”*, чья структура аналогична структуре класса *“Загрузчик данных”* по тем же соображениям, диктуемым необходимостью обеспечить возможность подключения к системе собственных фильтров и операторов сглаживания. На данный момент возможность подключения внешних модулей фильтрации и сглаживания данных находится на этапе разработки, однако, в ядре системы уже реализован стандартный фильтр и три вида операторов сглаживания (см. следующий раздел статьи).

После того, как запись прошла предобработку, то есть нежелательные отсчёты отфильтрованы, а возможные шумы сглажены, обычно выполняется этап обнаружения окуломоторных событий, таких как фиксации, саккады и прочие. Современные системы анализа ГДА могут обнаруживать от 3 до 5 различных видов событий, а литературный поиск позволяет насчитать более десяти видов! При этом на основные виды событий (фиксация и саккада), как правило, приходится несколько уже ставших стандартными алгоритмов детекции (см. Мармалюк и др., 2015) с многочисленными настроечными параметрами. Приоритет использования алгоритмов зависит, в основном, от частоты дискретизации записи и степени зашумления данных.

Разрабатываемая авторами данной статьи система по умолчанию содержит реализации трёх алгоритмов детекции («I-DT», «I-VT», адаптивный «I-VT»), описанных в следующем разделе. Каждый из алгоритмов оформляется в виде объекта класса *“Детектор событий”*, по структуре аналогичного структуре представленных ранее классов операторов.

Специальный метод детекции, обнаруживая с помощью заданного детектора окуломоторные события, помечает соответствующие им группы отсчётов уникальными номерами и текстовыми метками. Номера и метки идентифицируют отдельные события, которые обрабатываются с помощью специального *анализатора событий*, позволяющего вычислять различные одномерные показатели событий, такие как моменты начала и конца, длительность, позиция фиксации, позиции начала и конца саккады, амплитуда и кривизна саккады и т.п. Для каждого рассчитанного значения фактора указывается, к какой группе отсчётов траектории, образующих окуломоторное событие, оно принадлежит, а также каков вид этого окуломоторного события.

Аналогом анализатора событий является *анализатор объекта* как оператор, используемый для расчёта одномерных характеристик более «крупных» структурных составляющих комплексного наблюдения, таких как траектория взора в целом, набор показателей событий заданного вида, последовательность посещённых областей интереса и т.п. Примерами рассчитываемых показателей указанных объектов являются, соответственно, длина пути траектории, средняя длительность фиксации и длительность первого посещения заданной области интереса.

Указанные анализаторы выполняют вычисления, опираясь на объект класса *“Набор подфункций”*, состоящий, очевидно, из списка объектов класса *“Подфункция”*. Класс *“Подфункция”* используется для представления в системе операторов вычисления различных показателей, примеры которых приведены выше. Объект этого класса состоит из функции, реализующей вычисление показателя, названия функции для её идентификации, описания функции для понимания смысла показателя, вида операции (анализ элементов комплексного наблюдения или анализ данных отдельных событий), вида объекта операции, конкретизирующий либо элемент комплексного наблюдения, либо конкретный вид события, к которому применять операцию, и набора настроек,

необходимых для выполнения вычислений. Так, каждая подфункция содержит всю необходимую информацию для того, чтобы выбрать из записи данных необходимый компонент и выполнить необходимые расчёты.

Как уже говорилось ранее, на стимуле, используемом в испытании, может быть выделен набор областей интереса. Используя набор областей, исследователь может анализировать особенности распределения визуального внимания испытуемых по заданным участкам в контексте гипотез конкретного исследования.

Наиболее простым результатом анализа является набор «посещённых» в процессе испытания областей интереса. Для установления последовательности попаданий точек взора (или точек фиксации взора) в области интереса реализован так называемый *детектор попаданий в области интереса*, который использует траекторию взора, зарегистрированную в рамках какого-либо испытания, а также набор областей интереса, сопоставленных стимулу, использованному в этом испытании. Результат работы анализатора представляется в виде последовательности названий (или номеров) областей интереса, соответствующей по длине числу точек исходной траектории (или числу точек фиксации). Если точка траектории не принадлежит ни одной из областей, об этом делается специальная пометка в выходных данных. Построенные последовательности могут быть как объектом для непосредственной интерпретации, так и исходным материалом для последующего применения других видов анализа. Например, такая последовательность позволяет судить о том, происходил ли анализ стимула последовательно с переходом от одного участка к другому или с большим количеством возвратов к анализу уже просмотренных областей.

Объектом непосредственного анализа может выступать не только полученная последовательность, но и производный результат, отражающий её особенности. Для реализации различных видов анализа последовательностей «посещений» областей интереса, в системе предусмотрен специальный метод – *анализатор последовательности*. С помощью анализатора по последовательности могут быть рассчитаны:

- произвольные показатели (статистики) с помощью соответствующих субфункций (например, суммарная длительность пребывания взора в области «Глаза»);
- вектора показателей (например, вектор частот попаданий в различные области);
- матрицы показателей (например, оценки вероятностей переходов между областями, матрица представления преемника).

Помимо областей, которые могут быть заданы строго определёнными геометрическими фигурами, в системе предусмотрена возможность задания нечётких областей интереса, которые, позволяют получить вероятностную оценку попадания взора в область вместо дискретной (попал или нет). Если весь набор состоит из таких областей, то вместо последовательности попаданий пользователь получит последовательность из групп областей, в каждой из которых области упорядочиваются по величине вероятностного показателя их «посещения». Возможность формирования в системе таких «нечётких» последовательностей обеспечивает основу для их изучения и разработки методов их анализа.

Классы данных комплексного наблюдения

В этом подразделе описываются классы объектов, предназначенных для представления в системе данных комплексного наблюдения, таких как упомянутые ранее «Данные глаз», «Данные событий», «Последовательность областей интереса», «Вектор показателей областей интереса», «Матрица показателей областей интереса».

«Данные глаз» – сложный класс, позволяющий полностью представить результат регистрации движений глаз в конкретном испытании, результат их обработки (специальные метки отсчётов), а также те условия (настройки), при которых осуществлялась регистрация. Опишем результат регистрации, результат обработки и условия регистрации по-отдельности.

Результат регистрации – загружаемый в систему набор временных рядов, содержащих: 1) моменты времени регистрации отсчётов записи; 2) координаты положения взора на калибровочной плоскости для левого и правого глаза (возможно, только одного); 3) величины раскрытия зрачка для левого и правого глаза; 4) дополнительные ряды, которые могут быть указаны при загрузке данных из внешнего источника (ими могут быть, например, координаты центра зрачка или ИК-блика на кадрах видеозаписи айтрекера, либо другие данные, зарегистрированные тем же айтрекером).

Результат обработки – получаемый в ходе обработки траектории взора набор временных рядов, содержащих: 1) информационные метки, указывающие на точки траекторий левого и правого глаза, относящиеся к так называемым пропускам, обнаруживаемым специальным фильтром данных и 2) информационные метки, указывающие на точки траекторий левого и правого глаза, относящиеся к остальным видам окуломоторных событий, обнаруживаемых специальным детектором.

Условия регистрации – объект класса «Conditions», предназначенного для представления в системе сведений об экспериментальных условиях, в которых производилась регистрация ГДА. Такие сведения включают:

- а. режим записи (монокулярный с указанием глаза, либо бинокулярный), частота дискретизации записи,
- б. расстояние до стимульного экрана (или калибровочное расстояние при использовании мобильного айтрекера),
- в. разрешение стимульного экрана (или разрешение видеокadra, в координатах которого представлена траектория взора, полученная в условиях мобильной регистрации),
- г. размеры стимульного экрана по ширине и высоте (или размеры калибровочной области при использовании мобильного айтрекера),
- д. способ аппроксимации формы зрачка, зависящий от алгоритмов детекции, используемых производителем айтрекера (обычно это либо окружность, либо эллипс);
- е. единица измерения времени (микросекунды, миллисекунды, секунды);
- ж. единицы измерения величины раскрытия зрачка (пиксели, миллиметры или сантиметры).

Таким образом, объект класса “Данные глаз” содержит всю необходимую информацию для предобработки траекторий (фильтрации пропусков), расчёта скоростей и ускорений (в т.ч. угловых), детекции окуломоторных событий, построения соответствующих графиков с выделением обнаруженных событий. При этом обеспечивается возможность задавать при необходимости уникальные настройки каждого испытания, что немаловажно для учёта особенностей проведения комплексных экспериментов.

В системе по умолчанию предусмотрены следующие виды событий: «Фиксация», «Саккада», «Глиссада», «Следящее движение», «Пропуск» и «Артефакт». Под глиссadou понимается конечный корректировочный участок саккады (см. Nyström & Holmqvist, 2010). Под артефактами понимаются как участки записи, которым соответствуют сверхвысокие скорости и ускорения движения взора, так и экстремально короткие события, на длительность которых в рамках классических алгоритмов устанавливаются эвристические ограничения.

Последовательное применение детектора событий и анализатора данных событий к объекту класса “Запись данных” позволяет обнаружить в анализируемой записи указанные окуломоторные события и рассчитать их информативные показатели. Результаты расчётов представляются в виде объекта класса «Данные факторов», хранящегося в составе объекта класса «Запись данных», к которому были применены операции обработки. Факторами в данном случае являются различные показатели обнаруженных событий.

Ряд классов системы предназначен для хранения данных, связанных с областями интереса. Так, для представления данных о последовательности посещённых областей

используется класс *«Последовательность областей интереса»*. Объект этого класса получается в результате работы детектора попаданий в области интереса, описанного выше, и содержит список имён областей, указанных в порядке, соответствующем времени «попадания». В данном списке идущие последовательно области с одинаковыми наименованиями группируются, т.е. число элементов последовательности соответствует не числу точек исходной траектории, а количеству переходов между областями. При обработке данных пользователь может указать, включать ли в результаты обработки метки о пребывании взора вне областей.

Для представления данных о характеристиках переходов между областями интереса используется класс *«Матрица переходов между областями интереса»*. Объект данного класса содержит матрицу, число строк и столбцов которой равно количеству областей интереса в заданном наборе. Эта матрица является результатом анализа последовательности посещённых областей, на пересечении её строк и столбцов располагаются вычисленные значения некоторой характеристики перехода между соответствующими областями. Такой характеристикой может быть, например, абсолютная частота перехода из одной области в другую. В данном объекте также сохраняются данные о названии используемого набора областей и сведения о том, учитывалось ли направление перехода при вычислениях.

Каждой из областей интереса можно поставить в соответствие некоторую статистику. Результат вычисления статистик для набора областей записывается в объект класса *«Вектор статистик областей интереса»*. Например, можно рассчитывать число точек траектории, попавших в данную область, долю таких попаданий, количество фиксаций взора в данной области, суммарное время, затраченное на изучение области, длительность первой фиксации в области. Этот список может быть дополнен другими параметрами, если исследователь определит функции, необходимые используемые для их вычисления. На данный момент в системе отсутствуют функции расчёта показателей областей интереса, но в этом направлении ведётся активная работа.

Основные методы и функции ядра системы

В данном разделе представлено описание основных методов и функций ядра системы, реализованных на момент написания статьи. Используя экземпляры операторов обработки данных, описанных в предыдущем разделе, методы ядра решают задачи загрузки данных айтрекинга в систему, их анализа (фильтрации, сглаживания, детекции событий, оценки показателей событий, оценки статистик данных комплексного наблюдения) и визуализации.

Методы загрузки данных

В системе по умолчанию используется *метод загрузки данных айтрекинга*, считывающий их из текстовых файлов: временные ряды, содержащие значения моментов времени, координат взора, величин раскрытия зрачка, и другие дополнительные каналы, а также дополнительную информацию, такую как код испытуемого, которому соответствует файл, размеры стимула, номера испытаний и т.п. Предполагается, что данные представлены в виде текстовых файлов с разделителями, позволяющих структурированно хранить табличные данные (каждый временной ряд – это столбец таблицы). Данный способ хранения данных является стандартным и поддерживается (в рамках функций экспорта) во всех системах регистрации и анализа ГДА.

При загрузке данных система позволяет установить соответствие между номерами столбцов таблицы и видами данных, которые они содержат. Это немаловажно, поскольку разные системы экспортируют данные в табличный формат, размещая временные ряды данных в порядке, который, как правило, не зависит от пользователя. Такой приём обеспечивает универсальность стандартного загрузчика. Дополнительно пользователь может указать, какой из временных рядов содержит метки испытаний, которые могут

быть представлены как порядковые номера испытаний (в данных, полученных с помощью решений компании SMI, это столбец с названием «Trial»), так и, например, название демонстрируемого стимула-изображения. В процессе загрузки таблица данных разбивается на подтаблицы, соответствующие отдельным испытаниям в рамках эксперимента, определяемыми различными уникальными значениями соответствующего столбца меток.

В случае если в «шапке» текстового файла содержится вспомогательная информация, которую целесообразно автоматически загрузить в систему (например, код испытуемого, размер стимула и т.п.), то она, как правило, также представляется в структурированном виде, а именно в виде строковых пар «ключ-значение» (key-value). Например, код испытуемого может храниться в шапке файла данных так: «Subject: Ivan Ivanov». В данном примере, подстрокой, определяющей ключ, является «Subject», разделитель – знак двоеточия, а значение ключа – «Ivan Ivanov». Подобный способ хранения обеспечивает возможность считывания значений известных пользователю ключей с помощью простейших регулярных выражений, реализованных в системе ETRAN. На данный момент в системе реализовано автоматическое определение лишь кода испытуемого для SMI-файлов, но в дальнейшем планируется реализовать возможность загружать больше данных: размер стимула, расстояние до экрана стимульного монитора, частоту дискретизации записи, название стимула и т.п. В идеале пользователь должен иметь возможность загружать из шапки значения произвольных ключей и осуществлять по ним поиск или какие-либо операции.

Помимо данных айтрекинга в систему с помощью *метода загрузки значений факторов* могут быть загружены таблицы значений факторов испытуемых, испытаний и стимулов. При загрузке этих данных пользователь указывает на расположение файла данных, указывает необходимые сведения о факторах (шкала измерения, наименования уровней для номинальных и порядковых факторов), а также указывает на расположение столбца идентификаторов (кодов, имён) испытуемых (испытаний или стимулов), которым соответствуют наборы значений факторов.

При загрузке в систему стимульного материала (с помощью специального *метода загрузки стимула*), который на данном этапе разработки может быть представлен набором изображений, пользователь указывает место расположения файла стимула и наименование стимула. Для загруженных стимулов в системе могут быть заданы области интереса.

Методы анализа данных

Перед использованием большинства методов анализа данных айтрекинга обычно осуществляют их предобработку. В системе реализованы основные необходимые для предобработки данных функции и методы, выполняющие фильтрацию артефактов и сглаживание траектории, а также выражающие координаты позиций взора в угловых градусах для корректного сопоставления результатов, полученных в различных экспериментах. Фильтры позволяют выявлять в записи точки с нулевыми координатами, а также точки, выходящие по координатам за пределы стимульного материала. Алгоритмы сглаживания, реализованные в системе – метод скользящего среднего, метод скользящей медианы и метод Савицкого-Голэя (Savitzky & Golay 1964) – описаны далее.

Скользящее среднее (или медиана) в точке вычисляется как среднее арифметическое (или медиана) значений анализируемого временного ряда в определённой временной окрестности вокруг точки, степень сглаживания зависит от величины окрестности. Расчёт производится последовательно во всех точках записи, а результат рассматривается как сглаженная версия исходного временного ряда. Преимуществами скользящего среднего и медианы является простота интерпретации результата, корректная работа при наличии тренда в данных и другие. Медиана же является более устойчивой к выбросам оценкой центральной тенденции, поэтому её рекомендуется использовать в

случае большого количества одиночных артефактов в записи. Недостатками скользящего среднего и медианы является эффект запаздывания оценок, а также краевые эффекты (меньшая надёжность результата сглаживания по краям записи, обусловленная меньшим числом усредняемых координат).

Метод Савицкого-Голэ, он же цифровой фильтр Савицкого-Голэ, позволяет извлекать полезную информацию из зашумленных экспериментальных данных, при правильном подборе параметров, не особо размывая пики, фронты и спады, в отличие от многих методов сглаживания. Суть метода заключается в том, что по анализируемому ряду, как и в методе скользящего среднего, «пробегает» временное окно, внутри которого эмпирическая зависимость аппроксимируется с помощью полинома заданного невысокого порядка, начиная с квадратичной функции (параболы). Результат аппроксимации – гладкая дифференцируемая функция, из которой путём дискретизации выбираются значения, соответствующие моментам времени, в которые фиксировались отсчёты исходного сигнала (окулограммы). Таким образом, данный метод является аналогом локальной полиномиальной регрессии. Основным преимуществом метода является его способность сохранять особенности исходного сигнала, такие как положение и ширина локальных максимумов и минимумов, обычно загромождаемые при использовании других техник скользящего сглаживания (например, скользящего среднего). Это очень важное преимущество в плане точности детекции окуломоторных событий, поскольку пики угловой скорости, рассчитываемой по сглаженным исходным данным, определяют положение саккад. Недостатком метода является необходимость экспериментального подбора параметров для оптимальной фильтрации конкретного вида данных.

Рассмотрим теперь реализованные подходы к оценке скоростей и ускорений движения взора по поверхности стимула (в т.ч. угловых, рассчитываемых по последовательностям позиций взора, выраженных в угловых градусах относительно начала координат – левого верхнего угла монитора или выбранной референтной точки). Соответствующие графики скоростей или ускорений необходимы, например, для визуальной оценки корректности работы детектора событий. В системе реализовано два подхода: конечно-разностная аппроксимация и оценка скорости и ускорения как первой и второй производной аппроксимирующего полинома (с использованием возможностей метода Савицкого-Голэ). При конечно-разностной аппроксимации скорость (моментальная) в точке траектории оценивается просто как отношение пути между этой точкой и следующей за ней к времени, прошедшему с момента регистрации первой точки до момента регистрации второй точки. Ускорение оценивается при этом как отношение прироста моментальных скоростей к соответствующему приросту времени. Такие оценки наиболее просты, но не являются самыми надёжными и точными в силу высокой степени влияния на них возможных шумов. При использовании оценок скорости и ускорения по методу Савицкого-Голэ выполняется сначала аппроксимация данных, как было описано выше, а затем аналитическое (не численное) взятие производной от аппроксимирующих полиномов. В результате дифференцирования, например, полинома второй степени, получается линейная функция, выражающую скорость движения взора, а повторное дифференцирование даёт константу ускорения. Таким образом, оценки скорости и ускорения этим методом являются более надёжными, поскольку при аппроксимации учитываются несколько точек в окрестности, определяемой длиной скользящего окна.

В системе ETRAN реализован абстрактный *метод детекции событий*, позволяющий обнаруживать окуломоторные события в траекториях движения взора по поверхности стимульного материала. На момент написания статьи, метод позволяет использовать для обнаружения событий несколько алгоритмов детекции: алгоритм идентификации событий по порогу скорости IVT (Salvucci & Goldberg, 2000), IVT с пост-обработкой (Tobii, 2010), адаптивный IVT (Nystrom & Holmqvist, 2010), алгоритм идентификации событий по порогу дисперсии IDT (Salvucci & Goldberg, 2000). В результате работы каждого алгоритма к данным соответствующей траектории

добавляются специальные метки, указывающие номер и вид события, к которому относится та или иная точка траектории (сэмпл). Указанные метки используются в дальнейшем как для визуализации событий на графиках, так и для разбиения траектории на участки, по которым рассчитываются показатели событий. Теперь опишем особенности реализованных алгоритмов.

Алгоритм IVT – классический алгоритм, основанный на пороговой скорости. Используется, как правило, при обработке данных, записанных с высокой частотой регистрации (500 Гц и более). Для каждой двух соседних точек данных рассчитывается мгновенная угловая скорость. Участки данных, для которых мгновенная скорость превышает заданный порог (как правило, больший 30 °/с), рассматриваются как саккады, остальные участки – как фиксации. Более сложный вариант алгоритма предполагает также расчет мгновенного ускорения, в таком случае к саккадам относятся участки данных, для которых как скорость, так и ускорение превышают пороговые значения. Недостатком такой реализации является порождение сверхкоротких артефактных фиксаций, связанных с частичной окклюзией зрачка (Барабанчиков и Жегалло, 2013) и другими факторами (например, на рис. 1 видно, что последовательно идущие саккада и глиссда распознаны как две саккады с очень короткой фиксацией в промежутке). Также существуют реализации алгоритма с детальной постобработкой результатов детекции (Olsen, 2012): заполнением пропусков путём линейной интерполяции пропущенных точек траектории, объединением близких во времени и пространстве фиксаций, удалением коротких фиксаций и прочими эвристическими приёмами.

Адаптивный IVT – относительно новый алгоритм детекции, отличающийся от ставшего классическим IVT тем, что позволяет избежать ручного выбора ряда параметров детекции, среди которых основным является порог угловой скорости движения взора, отделяющий саккадические движения от фиксаций. То есть параметры детекции выбираются исходя из специфики обрабатываемых данных. Данный алгоритм позволяет помимо фиксаций и саккад выделять также и глиссы – низкоамплитудные корректирующие саккады (см. рис 1). Моргания относятся алгоритмом к артефактам записи, то есть не выделяются отдельно от различных «выбросов» и пропусков.

Адаптивный алгоритм IVT в качестве объекта анализа использует временной ряд моментальных скоростей, оцененных методом Савицкого-Голэя. Помимо сглаживания, из анализа исключаются точки траектории со слишком высокой моментальной скоростью (>1000 °/с) и ускорением ($>100\,000$ °/с), а также точки взора с нулевыми координатами. Работа алгоритма разделяется на пять основных последовательных этапов: фильтрация и удаление артефактов, определение пиков саккад, обнаружение начала и конца каждой саккады, обнаружение глисс, обнаружение фиксаций. Алгоритм сначала выполняет итеративную оценку пороговой угловой скорости, используемой при детекции саккад. Начальное значение пороговой скорости устанавливается в диапазоне 100–300°/с. По всем интервалам данных, для которых скорость выше пороговой, вычисляются среднее значение скорости μ и стандартное отклонение σ . Новое значение пороговой скорости вычисляется как $\mu + 6\sigma$. Алгоритм завершается, когда вновь вычисленное значение пороговой скорости отличается от предыдущего не более чем на 1%. На следующем этапе выполняется выделение саккад. От каждого интервала данных, на котором скорость выше порогового значения, выполняется просмотр вперед и назад. Началом саккады считается первый интервал, на котором угловая скорость превышает значение $\mu + 3\sigma$. Предельное значение угловой скорости, при котором оканчивается саккада, вычисляется как взвешенная комбинация начальной угловой скорости и факторов, характеризующих локальный уровень шума. При выделении саккад также учитывается их минимально допустимая продолжительность. В качестве фиксаций алгоритм выделяет участки данных, которые не являются шумом, саккадами или глиссадами. Дополнительно используется критерий минимально допустимой продолжительности фиксаций (Барабанчиков и Жегалло, 2013).

В отличие от алгоритмов, основанных на анализе скорости перемещения взора, алгоритм *IDT* учитывает тот факт, что точки фиксации находятся достаточно близко друг к другу. Алгоритм имеет два настроечных параметра: минимальную продолжительность фиксации и пороговую дисперсию. Фиксациями считаются фрагменты данных, продолжительность которых не меньше минимальной заданной, а дисперсия не больше максимального порога. Остальные данные относятся к саккадам. При относительно малом значении пороговой дисперсии выделение фиксаций алгоритмом *IDT* оказывается практически невозможным в силу воздействия шумов. При увеличении значения пороговой дисперсии число выделенных фиксаций сначала возрастает, а затем начинает сокращаться, так как соседние фиксации сливаются в одну большую. Типичные значения параметров детекции: порог дисперсии – $0,5\text{--}1^\circ$, минимальная продолжительность фиксации – 50 мс. Результатом работы алгоритма является последовательность меток, обозначающих фиксации и саккады. Алгоритм применяется, как правило, при обработке данных, записанных с низкой частотой (Барабанчиков и Жегалло, 2013).

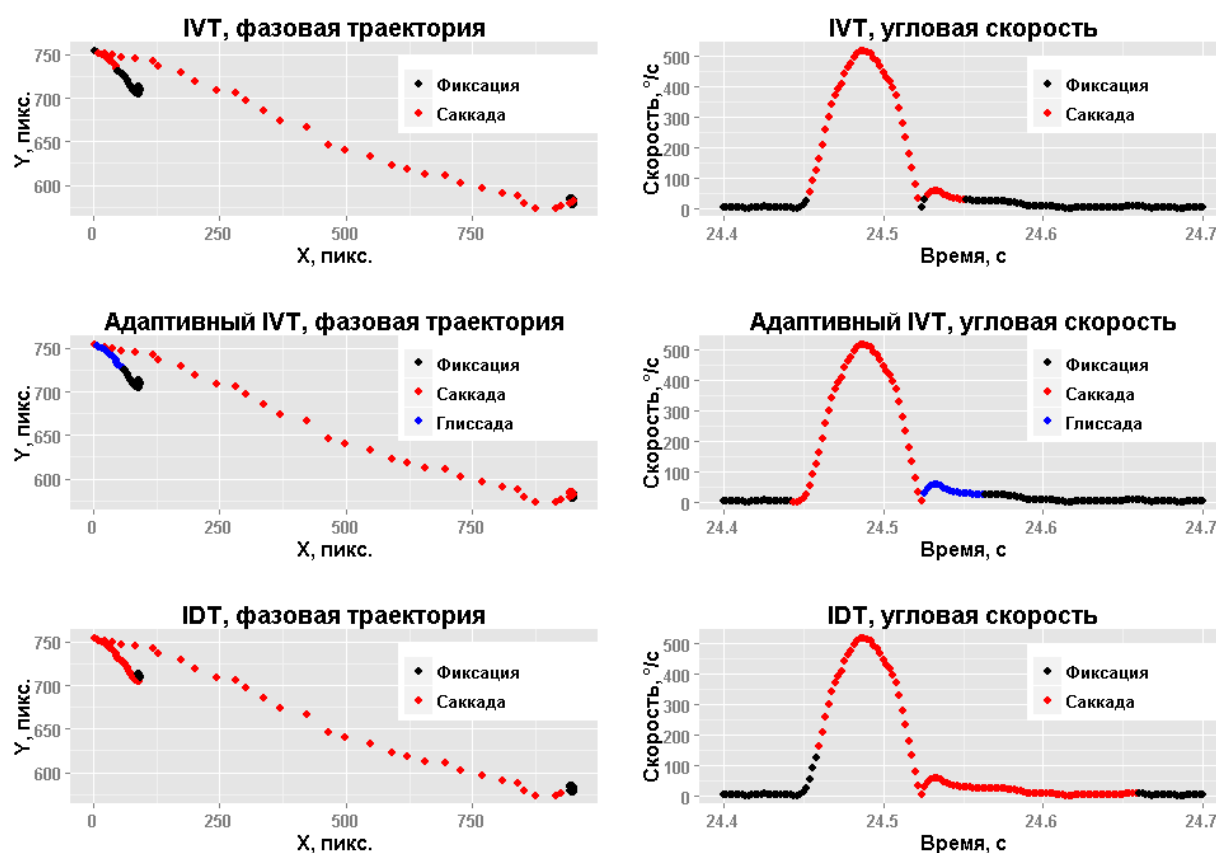


Рис. 1. Пример результатов детекции окуломоторных событий, полученных с помощью алгоритмов IVT, адаптивного IVT и IDT при рекомендованных в литературе настройках.

По участкам траектории взора, рассматриваемых в результате детекции в качестве тех или иных окуломоторных событий, с помощью специального *метода анализа событий*, опирающегося на анализатор событий и набор соответствующих подфункций, могут быть рассчитаны различные качественные и количественные показатели отдельных событий. Такие пространственные показатели, как длина, амплитуда, разброс и т.п., могут быть рассчитаны как в пикселях, так и в угловых градусах. Скоростные показатели аналогично рассчитываются с учётом единиц измерения показателей длины пути. В системе на данный момент доступны подфункции для расчёта следующих показателей:

- 1) метка валидности события, указывающая, что хотя соответствующий участок траектории и был классифицирован как то или иное событие, но при этом включает некорректные данные (пропуски или артефакты);

- 2) моменты времени начала и конца события относительно начала испытания, а также длительность события;
- 3) горизонтальные и вертикальные координаты точки взора на момент начала и конца события – актуально для саккад и глиссад;
- 4) горизонтальная и вертикальная «амплитуды» события, рассчитываемые как абсолютные величины разницы соответствующих координат на момент начала и конца события, а также пространственная «амплитуда», рассчитываемая как евклидово расстояние от точки взора на момент начала до точки взора на момент конца события – актуально для саккад и глиссад;
- 5) горизонтальная и вертикальная координаты центра масс, рассчитываемые как средние значения координат точек взора, относящихся к событию;
- 6) горизонтальный и вертикальный разбросы, рассчитываемые как стандартные отклонения координат точек взора, относящихся к событию, а также радиус, рассчитываемый как среднее евклидово расстояние от всех точек взора до центра их масс;
- 7) длина пути участка траектории, относящегося к событию, рассчитываемая как сумма длин всех отрезков участка, образуемых смежными точками взора;
- 8) безразмерная кривизна участка траектории, рассчитываемая как длина пути, делённая на пространственную амплитуду события;
- 9) пиковые скорость, ускорение и торможение на участке траектории – актуально для саккад и глиссад;
- 10) асимметрия саккады, характеризующая соотношение продолжительности фаз ускорения и торможения во время саккады;
- 11) ориентация саккады, определяемая углом между вектором саккады (вектор, началом которого является позиция первой точки саккады, а концом – позиция последней) и горизонтальной осью системы координат стимула;
- 12) среднее значение и стандартное отклонение величины раскрытия зрачка.

По каждому из объектов, входящих в состав комплексного наблюдения, с помощью специального *метода анализа объектов*, опирающегося на анализатор объектов и набор соответствующих подфункций, можно рассчитывать статистические показатели, характеризующие наблюдение в целом. В системе на данный момент доступны подфункции для расчёта следующих показателей:

- 1) общая длительность траектории (в испытании);
- 2) длина траектории, рассчитываемая как сумма длин всех отрезков, образуемых смежными точками взора;
- 3) количества событий разного вида.

Добавление в систему подфункций расчёта других показателей не составляет труда: пользователю необходимо лишь дополнить соответствующий файл определением функции на языке R. В дальнейшем будет реализована возможность подключения подфункций посредством графического интерфейса.

Методы визуализации

Визуальный анализ данных окулографии является мощным средством разведочного анализа. В том числе аккуратная инспекция графиков исходных данных позволяет обнаружить наличие шумов и артефактов в записи. Просмотр, например, графика изменения угловой скорости движения взора, в котором различные участки размечены с учётом обнаруженных окуломоторных событий, позволяет оценить корректность работы алгоритма детекции или заданных настроек.

В разрабатываемой системе будет реализовано порядка 6 видов графиков (как для отображения данных айтрекинга, так и для визуализации распределений результирующих параметров и статистических зависимостей), часть из которых будет поддерживать опции динамического воспроизведения (последовательного отображения графиков, построенных

по локальным во времени участкам траектории) и группового отображения, когда на график или диаграмму наносятся данные нескольких наблюдений: например, несколько фазовых траекторий поверх одного стимула.

На данный момент в системе реализованы методы для построения таких видов графиков, как *фазовая траектория* (суть точечная диаграмма координат позиций взора, см. левую диаграмму на рис. 2) и *временной ряд* (график, отражающий изменение некоторой одномерной характеристики во времени; см. правый график на рис. 2 и рис. 3). Фазовая траектория может быть отрисована поверх стимульного материала, а её точки могут быть раскрашены в соответствии с окуломоторным событием, которое они образуют. График вида «временной ряд» позволяет отображать как динамику позиции взора, скорости движения взора, ускорения движения, величину раскрытия зрачка, так и любые другие данные, загруженные в систему извне в качестве дополнительных (например, динамика позиции центра зрачка в кадрах, регистрируемых айтрекером и т.п.).

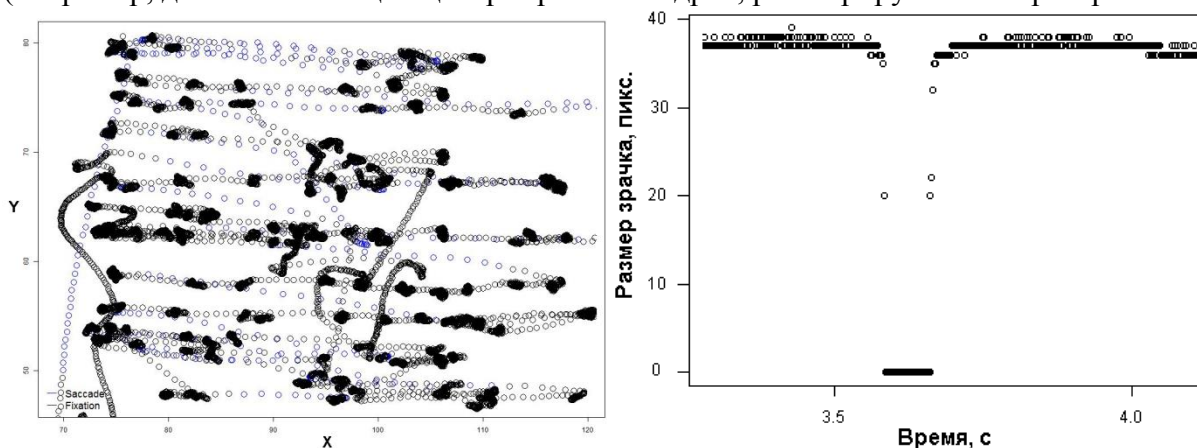


Рис. 2. Фазовая траектория взора в координатах стимульного материала (слева) и участок временного ряда величины раскрытия зрачка (справа).

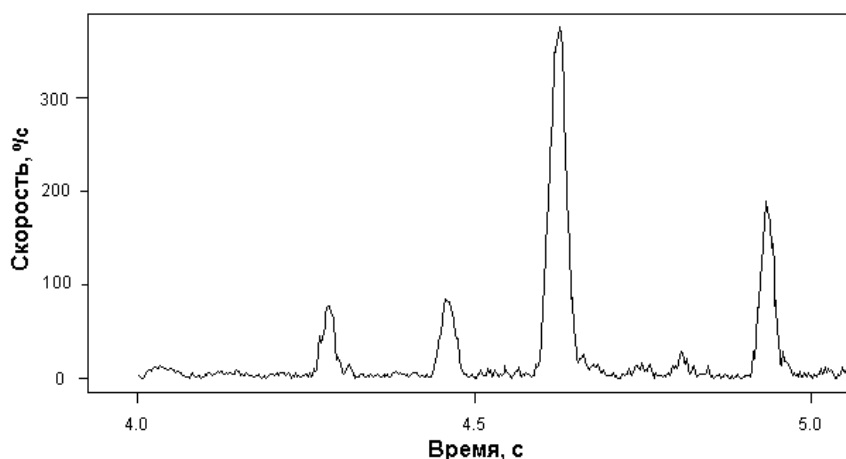


Рис. 3. Угловая скорость движения взора по плоскости стимула.

Пользовательский интерфейс системы

Язык R предоставляет пользователю лишь интерфейс командной строки, что делает проблематичным ее использование неспециалистами в области программирования. Понимая важность обеспечения таких пользователей возможностями R, а также в педагогических целях, представителями R-сообщества были разработаны пакеты для R (Tcl/Tk, RGtk, gWidgets2 и другие), позволяющие создавать графические пользовательские интерфейсы (GUI, graphical user interface). К сожалению, настройка существующих GUI для нужд данного проекта не представляется возможной, поскольку все они рассчитаны на работу лишь с данными, представленными в табличном виде, а не со сложными объектами вроде наборов окуломоторных событий, графических стимулов и т.п.

Для создания собственного GUI системы используется пакет Shiny, что позволит любому экспериментальному психологу проводить анализ собственных данных, опираясь на интуитивно понятный графический интерфейс с элементами диалога, не теряя при этом возможности использовать интерфейс командной строки (Мармалюк и др., 2015).

Shiny – средство для создания приложений с веб-интерфейсом. Это средство доступно в виде пакета-расширения для языка программирования R. Оно позволяет создавать интерактивные веб-приложения без знаний в области веб-разработки (например, нет необходимости программировать на PHP и JavaScript). Для создания приложения достаточно знания языка R, элементов интерфейса Shiny, а также базовых принципов взаимодействия компонентов интерфейса с программным кодом, однако, возможно и расширение стандартного набора элементов интерфейса собственными компонентами (в этом случае требуется их программная реализация с использованием средств HTML, CSS и JavaScript).

Преимуществами Shiny являются кросс-платформенность (Shiny-приложения работают в большинстве стандартных браузеров), возможность создания собственных элементов интерфейса, обширный набор настраиваемых компонент, позволяющих осуществлять вывод таблиц и графиков, а также возможность работы с локальной версией приложения как на локальном компьютере без подключения к сети Интернет, так и с его веб-версией, обеспечивающей облачное хранение данных.

Примеры интерфейсов, созданных с помощью Shiny, можно ознакомиться на сайте создателя пакета (RStudio Inc., 2015), который также представил миру RStudio – среду разработки для языка R.

В рамках текущего проекта было решено реализовывать локальную версию приложения, взаимодействующего с ядром системы, поскольку объёмы обрабатываемых данных видеоокулографии слишком велики для комфортной в плане быстродействия удалённой работы с ними.

Общий вид интерфейса (первая тестовая версия) системы представлен на рис. 4

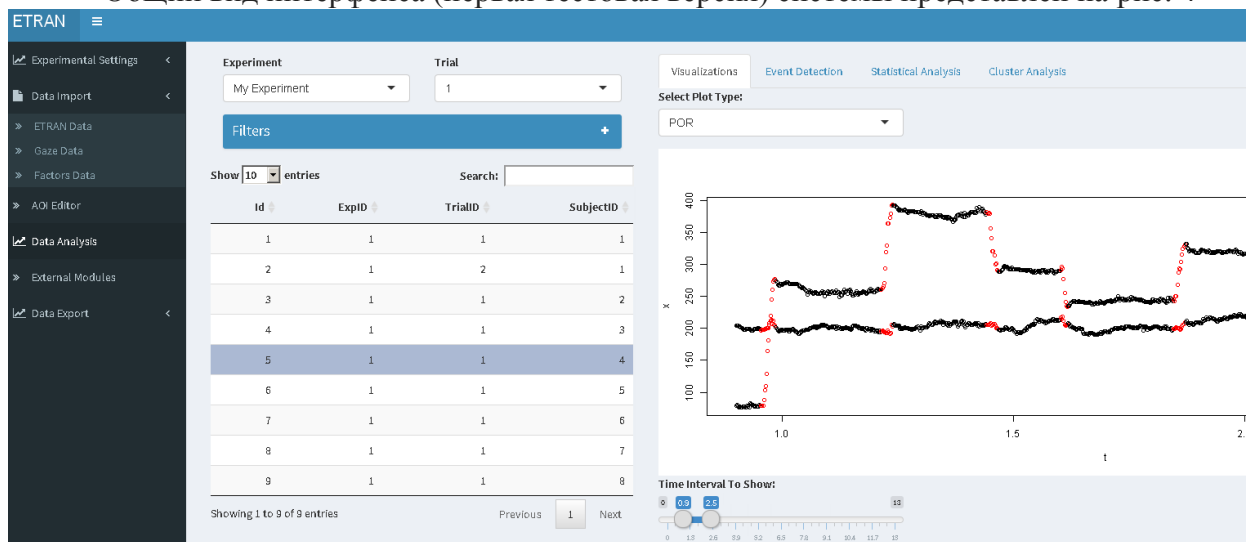


Рис. 4. Прототип графического пользовательского интерфейса системы ETRAN. Левая панель предназначена для перемещения между закладками «Настройки эксперимента», «Импорт данных», «Редактор областей интереса», «Анализ данных», «Подключение внешних модулей» и «Экспорт данных». Центральная панель закладки «Анализ данных» содержит таблицу комплексных наблюдений с возможностью применения фильтров.

Правая панель закладки «Анализ данных» содержит, в свою очередь, подразделы «Визуализации», «Детекция событий», «Статистический анализ» и «Кластерный анализ». Выбор одной или нескольких строк в таблице наблюдений позволяет либо получить их визуализацию (в том числе групповую), либо указать оператору обработки, какие наблюдения необходимо анализировать.

Планируемые доработки

Описанные ранее методы и функции обработки и визуализации данных являются самыми основными, без которых использование системы не представляло бы смысла. Однако и этих функций совершенно недостаточно для полноценной научной работы, поскольку таковая подразумевает не только детекцию событий и проведение разведочного визуального анализа, но и применение строгих статистических критериев и методов анализа, позволяющих проверять различные исследовательские гипотезы. В данном разделе описываются необходимые с точки зрения авторов функции, находящиеся в разработке.

Очевидна *востребованность функции подключения к системе внешних пользовательских модулей*, поскольку зачастую научные работники в состоянии программное реализовать тот или иной алгоритм, автоматизирующий, например, расчёт нестандартных характеристик окулomotorных событий.

При работе с большим числом наблюдений (от 30) немаловажна *функция фильтрации выборки*, позволяющая как отыскать тот или иной результат эксперимента, так и выполнить анализ только конкретной подвыборки данных, определённой некоторым критерием. Таким критерием может быть ограничение на возраст или пол испытуемых, на ту или иную характеристику ГДА, принадлежность к некоторому кластеру, и, вообще говоря, любое осмысленное в контексте исследования составное условие. Фильтры данных вкупе с возможностями групповой визуализации откроют ещё больше возможностей для разведочного визуального анализа данных айтрекинга.

В современных системах обработки айтрекинговых данных стандартными являются *функции построения специализированных графиков*, как «путь сканирования» (scanpath) и «тепловая карта» (heatmap), которые необходимы и в разрабатываемой системе.

На момент написания статьи ядро системы включало методы создания различных областей интереса и объединения их в наборы, однако, это внутренние методы, позволяющие создать соответствующие объекты посредством исполнения команд в среде разработки. Реализованы методы задания и обработки динамических областей интереса, которые, по сути, отличаются от статических добавлением к набору областей дополнительного «измерения», т.е. для каждой динамической области набора определяется не один список детерминирующих её параметры характеристик, а множество таких наборов, соответствующих разным моментам времени. Такой подход позволяет отслеживать объединённые единым содержательным смыслом области на меняющемся во времени стимульном материале, например наборе кадров видеозаписи.

Для того чтобы рядовой пользователь мог создавать нужные области интереса, необходим *графический редактор областей интереса*, позволяющий буквально рисовать области поверх стимульного материала, пользуясь набором специализированных инструментов (прямоугольник, квадрат, эллипс, круг, многогранник, нечёткая область, регулярная сетка и т.п.). Разработка такого редактора потребует создания специализированного компонента, написанного на JavaScript и подключаемого к пакету Shiny.

Разрабатываемая система рассматривается, в том числе, и как средство выборочного статистического анализа данных айтрекинга. Статистический анализ подразумевает как анализ распределений одномерных показателей ГДА, так и анализ их взаимосвязей с внешними факторами. Учитывая, что в системе реализована функция загрузки значений внешних факторов испытуемых, стимулов и испытаний, а возможности языка R для статистических вычислений без сомнений огромны, реализация средств статистического анализа указанных данных заключается, фактически, в том, чтобы связать имеющиеся функции языка R с пользовательским интерфейсом разрабатываемой системы ETRAN. Данная задача, однако, не является тривиальной, поскольку требуется

предусмотреть возможность ввода многочисленных настроек методов статистического анализа и учесть множество вариантов выходных результатов применения статистических методов.

Помимо «классических» методов статистики планируется реализовать относительно новые в области айтрекинга алгоритмы кластерного анализа данных комплексного наблюдения, когда в качестве меры подобия траекторий могут выступать различные показатели:

- «расстояния», рассчитываемые по интегральным показателям ГДА, внешним факторам испытуемых, показателям степени следования пространственным стратегиям переходов между областями интереса (Хохлова, 2011), элементам матрицы переходных вероятностей или матрицы представления преемника (Мармалюк, Поляков, 2015) и т.п.;
- степени различия двумерных вероятностных распределений координат взора или точек фиксации (например, дивергенция Кульбака-Лейблера).

Для разведочного анализа эффективны также различные подходы к выявлению пространственных паттернов в траекториях взора или последовательностях фиксаций (Хохлова, 2011; Мармалюк, Звонкина, 2012; Мармалюк, Поляков, 2015). Эти подходы также планируется внедрить в систему.

Указанные подходы позволяют обнаруживать закономерности, которые либо сложно предугадать или сформулировать в виде гипотез на этапе планирования эксперимента, либо не укладывающиеся в рамки текущей парадигмы научных представлений в области и, следовательно, являющихся маловероятными кандидатами.

Наконец, в системе совершенно необходимы *гибкие средства экспорта данных*: как внутренних данных системы для работы с ними на другом компьютере, так и результатов обработки данных комплексного наблюдения для проведения дополнительного анализа в сторонних системах.

Основные результаты и выводы

1. Основными результатами первого этапа разработки системы являются:
 - ✓ реализовано ядро системы, включающее базовые классы сущностей, возникающих в рамках видеоокулографического эксперимента, а также абстрактные методы анализа соответствующих объектов, конкретизация которых выполняется посредством задания программных функций, автоматизирующих непосредственные расчёты;
 - ✓ создан рабочий макет пользовательского интерфейса системы;
 - ✓ определены пути дальнейшего развития системы, дополнительные компоненты системы находятся в разработке.
2. Обеспечена возможность расширения функций системы путём подключения к ней внешних модулей, реализованных на языке R.
3. Результаты тестирования системы, подробное рассмотрение которых выходит за рамки тематики журнала, позволяют говорить о высоких требованиях системы к оперативной памяти при работе даже с малыми выборками траекторий, полученных при высокой частоте дискретизации и течение длительного (от 2 минут) времени, а также о необходимости оптимизации скорости вычислений различными алгоритмами, что связано с особенностями языка программирования R. Так, можно сказать, что обнаружены узкие места используемых технологий и ищутся пути их устранения.

Поддержка

Публикация подготовлена в рамках поддержанного РГНФ научного проекта №14-06-12012 «Программное обеспечение с открытым исходным кодом для анализа результатов окулографических исследований».

Литература

1. Барабанщиков В. А., Жегалло А. В. (2013). *Регистрация и анализ направленности взгляда человека*. М.: Институт психологии РАН. 316 с.
2. Мармалюк П.А., Жегалло А.В., Юрьев Г.А., Панфилова А.С. (2015). *Принципы построения программного обеспечения с открытым исходным кодом для анализа результатов окулографических исследований*. Экспериментальная психология. Т.8. №1. С.127–144
3. Мармалюк П.А., Звонкина О.М. (2012). *Опорные показатели глазодвигательной активности при прохождении теста Равена и автоматизация их расчёта*. Молодые ученые - нашей новой школе. Матер. XI межвуз. науч.-практ. конф. с межд. участием. М.: МГППУ. С.350–352.
4. Мармалюк П.А., Поляков Б.Ю. (2015). *Анализ данных видеоокулографии с применением матриц представления преемника и деревьев решений* [Электронный ресурс]. Молодые ученые – столичному образованию. Материалы XIV городской научно-практической конференции. М.: МГППУ. CD-ROM. Загл. с этикетки диска.
5. Мармалюк П.А., Поляков Б.Ю. (2015). *Выявление информативных характеристик глазодвигательной активности с применением метода главных компонент и обучаемых моделей*. Молодые ученые - нашей новой школе. Матер. Всероссийской науч. конф. М.: Московский институт психоанализа. (В печати).
6. Хохлова А. А. (2011). *Исследование глазодвигательной активности при прохождении матричного теста интеллекта Равена*. Молодые ученые - нашей новой школе. Матер. X науч.-практ. межвуз. конф. Москва: МГППУ. С.343–345.
7. Юрьев Г. А., Панфилова А. С., Мармалюк П. А. (2015). *Архитектура программного обеспечения для анализа результатов окулографических исследований*. Программная инженерия. №1. С.24–33
8. Hayes T.R., Petrov A.A., Sederberg P.B. (2011). *A novel method for analyzing sequential eye movements reveals strategic influence on Raven's Advanced Progressive Matrices*. Journal of Vision. №10. P.1–11.
9. Nyström M., Holmqvist K. (2010). *An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data*. Behavior Research Methods. V. 42. №1. P.188–204.
10. RStudio Inc. [Electronic resource] *Shiny by RStudio. A web application framework for R*. 2014. URL: <http://shiny.rstudio.com/> (accessed 24.09.2015).
11. Salvucci D. D., Goldberg J. H. (2000). *Identifying fixations and saccades in eye-tracking protocols. Proceedings of the 2000 Symposium on Eye Tracking Research and Applications* (November 6-8, Palm Beach Gardens, Florida, USA). ACM Press, vol. 1, pp. 71–78.
12. Savitzky A., Golay M.J.E. (1964). *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*. Analytical Chemistry. V.36. №8. P.1627–39. doi:10.1021/ac60214a047.
13. Olsen A. Tobii I-VT Fixation Filter [Electronic resource]: Algorithm Description, 2012. URL: <http://www.tobii.com/en/eye-tracking-research/global/library/white-papers/the-tobii-i-vt-fixation-filter> (дата обращения 02.10.2015).
14. Zhegallo A.V. and Marmalyuk P.A. (2015). *ETRAN — R Extension Package for Eye Tracking Results Analysis*. Perception 0301006615594944, first published on August 19 as doi:10.1177/0301006615594944. URL: <http://pec.sagepub.com/content/early/2015/08/14/0301006615594944.full>

References Transliteration

1. Marmalyuk P.A., Zhegallo A.V., Yur'ev G.A., Panfilova A.S. (2015). *Printsipy postroeniya programmnoho obespecheniya s otkryтым iskhodnym kodom dlya analiza rezul'tatov okulograficheskikh issledovaniy*. Eksperimental'naya psikhologiya. T.8. №1. S.127–144

2. Marmalyuk P.A., Zvonkina O.M. (2012). *Opornye pokazateli glazodvigatel'noi aktivnosti pri prokhozhenii testa Ravena i avtomatizatsiya ikh rascheta*. Molodye uchenye - nashei novoi shkole. Mater. XI mezhvuz. nauch.-prakt. konf. s mezhd. uchastiem. M.: MGPPU. S.350–352.
3. Marmalyuk P.A., Polyakov B.Yu. (2015). *Analiz dannykh videookulografii s primeneniem matrits predstavleniya preemnika i derev'ev reshenii* [Elektronnyi resurs]. Molodye uchenye – stolichnomu obrazovaniyu. Materialy XIV gorodskoi nauchno-prakticheskoi konferentsii. M.: MGPPU. CD-ROM. Zagl. s etiketki diska.
4. Marmalyuk P.A., Polyakov B.Yu. (2015). *Vyyavlenie informativnykh kharakteristik glazodvigatel'noi aktivnosti s primeneniem metoda glavnykh komponent i obuchaemykh modelei*. Molodye uchenye - nashei novoi shkole. Mater. Vserossiiskoi nauch. konf. M.: Moskovskii institut psikhoanaliza. (V pechati).
5. Khokhlova A. A. (2011). *Issledovanie glazodvigatel'noi aktivnosti pri prokhozhenii matrichnogo testa intellekta Ravena*. Molodye uchenye - nashei novoi shkole. Mater. X nauch.-prakt. mezhvuz. konf. Moskva: MGPPU. S.343–345.
6. Yur'ev G. A., Panfilova A. S., Marmalyuk P. A. (2015). *Arkhitektura programmnogo obespecheniya dlya analiza rezul'tatov okulograficheskikh issledovaniy*. Programmnaya inzheneriya. №1. S.24–33

Support

This work was supported by the Russian Foundation for Humanities (project №14-06-12012 «Open-source software for oculography data analysis»)