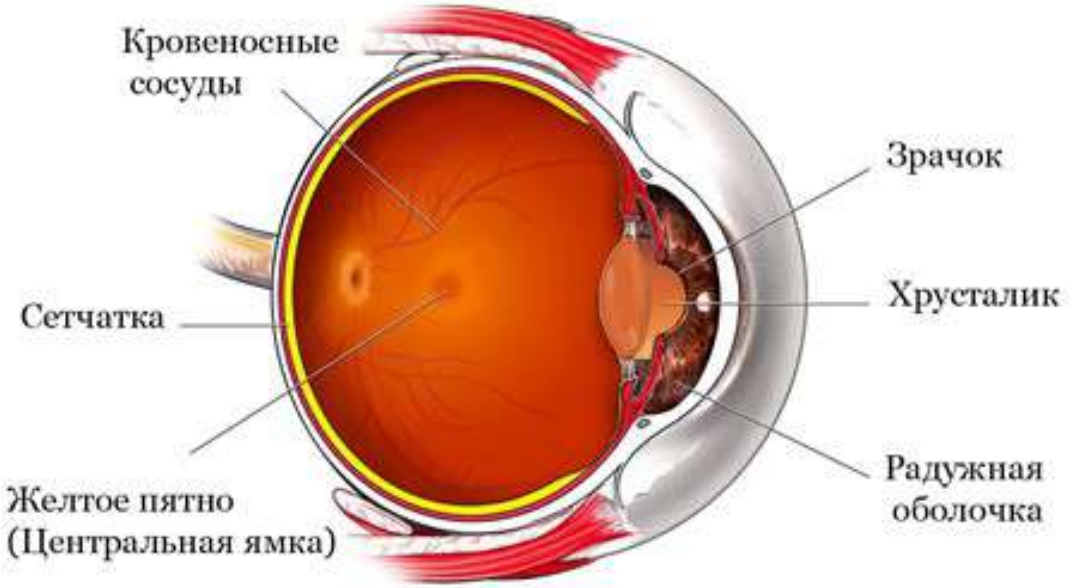
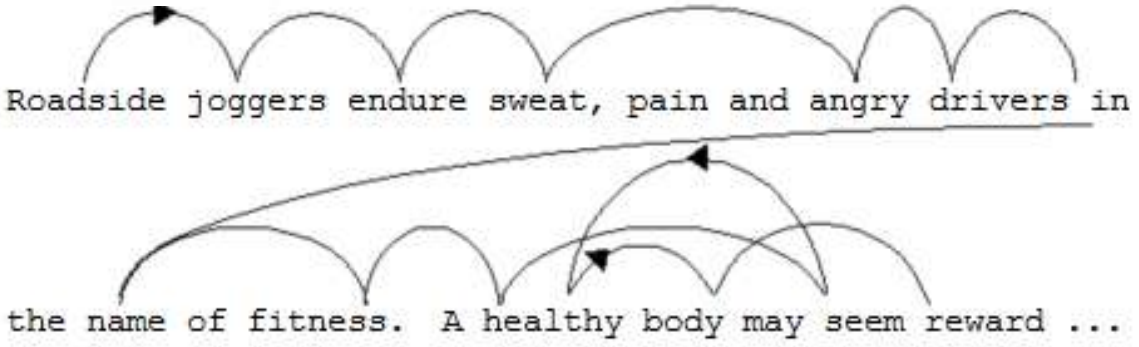
Using eye trackers for the oculographic recearch in GNU/Linux

А.В Дубицкий, Д.А. Костюк, А.А. Маркина

Как человек видит





Применение в UX-исследованиях

- Причины UX проблем

  (проблемы, связанными с заметностью элементов, точками фокуса внимания, ментальной нагрузкой и отвлечениями)
- Особенности поведения пользователей

  (стратегии визуального поиска, паттерны чтения и сканирования)
- Сравнение решений

Схема работы айтрекера

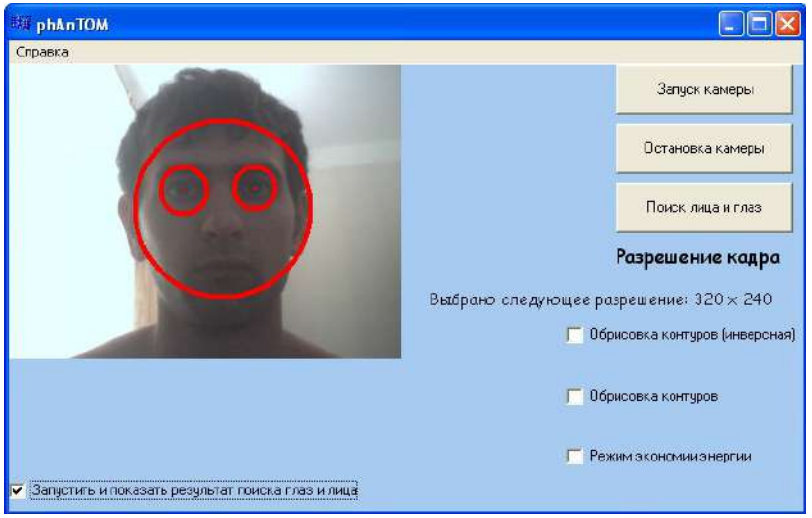
Сравнение решений — крайние случаи

- Профессиональные айтрекеры:

Особенности на примере профессиональных решений от tobii:

- Носимые либо дистанционные (монтируются под экраном в поле зрения пользователя
- Ценник от $ 10 000
- Часто совместимы с Linux и MacOS
  - например, отдают данные по протоколу TCP/IP

- Программные решения на основе веб-камеры:



Ряд свободных программных проектов нацелен на позиционирование взгляда с помощью веб-камеры:

- eviacam, TrackEye, WebGazer.js, ...
- Написаны на C#, Delphi, JavaScript
- Обычно распознают лицо в видео-потоке с помощью библиотеки OpenCV (иногда — с помощью собственных алгоритмов)
- Позиционируют взгляд с разрешением, вдвое ниже разрешения веб-камеры, не отслеживают поворот головы

Айтрекеры Tobii потребительского сегмента:

Два режима работы айтрекекра

1) в виде «сетевого адаптера с интерфейсом USB» (отдаёт координаты как сетевые пакеты по TCP/IP)
   - поддерживается в LINUX
2) в виде устройства с проприетарным USB-протоколом

Оба режима требуют проприетарный SDK

Два варианта SDK для Linux

1) старый gaze sdk
   - поддерживается больше айтрекеров
   - больше не распространяется, скачать трудно
2) новый Stream Engine SDK
   - заявлена поддержка всех, но по факту пока только Tobii 4c

Айтрекеры Tobii потребительского сегмента:

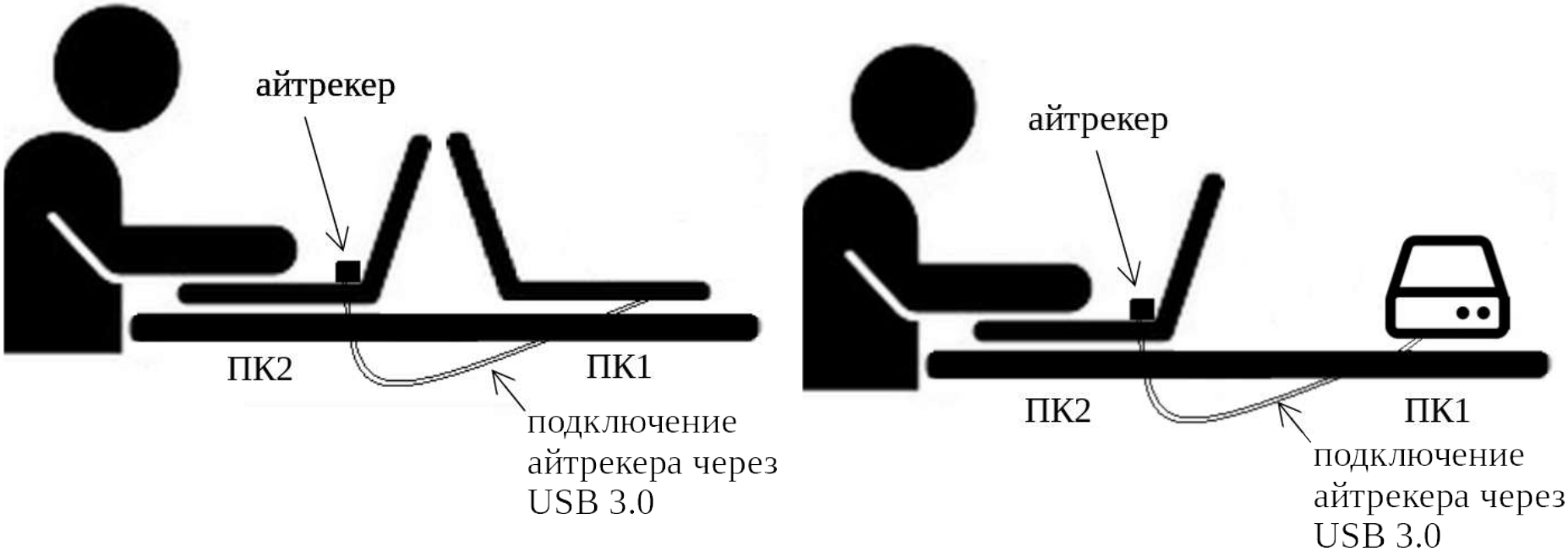
Tobii считает координаты фиксаций личными данными

– стандартная лицензия позволяет использовать SDK только если координаты не сохраняются в файл :)

– для всех остальных задач необходимо в индивидуальном порядке получать «продвинутую» лицензию, с дополнительными телодвижениями

– софт должен каждый раз рассказывать пользователю про сохранение его личных данных

Если с Linux-SDK не получилось:



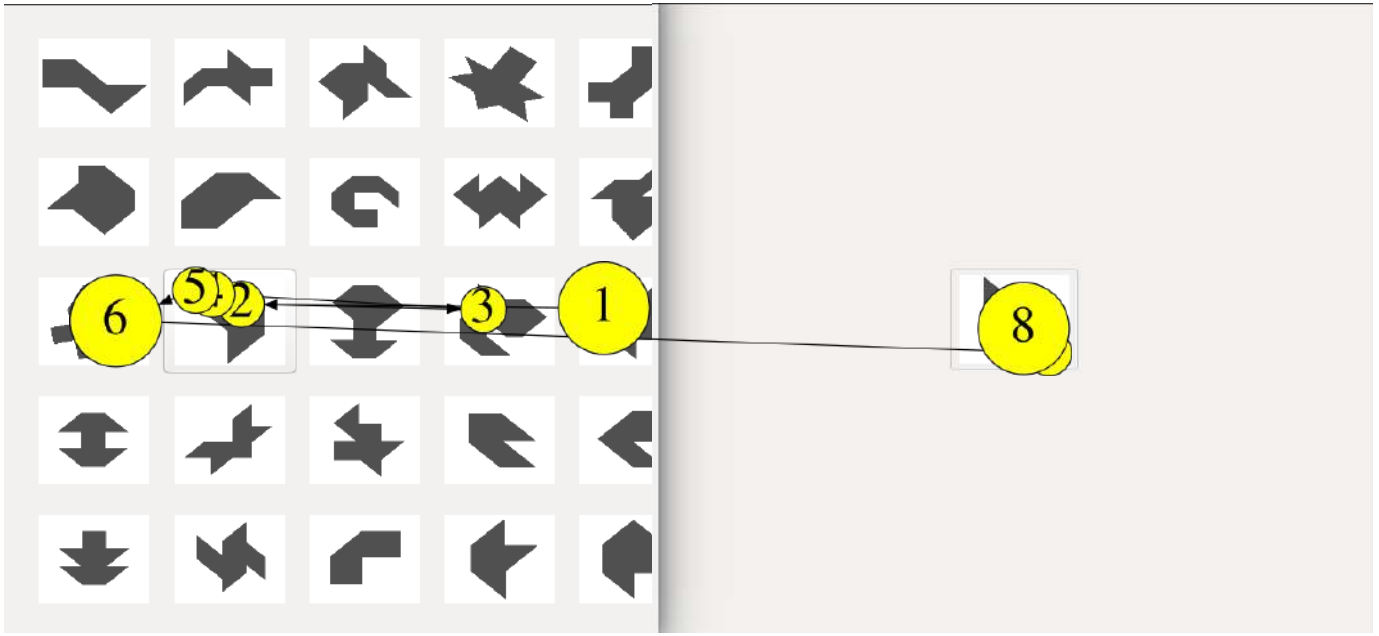
Получаемые данные

- Число фиксаций
- Длительность каждой фиксации
- Время до первой фиксации
- Длительность первой фиксации
- Общее число фиксаций
- Общее время фиксаций
- Скорость саккад
- Длина саккад
- Скорость просмотра
- Диаметр зрачка
- Сколько было фиксаций до посещения зоны интереса

Визуализация результатов эксперимента

- Gaze plot - граф, отражающий последовательность фиксаций взгляда
  - Метка узла — его номер в последовательности фиксаций
  - Размер узла — длительность фиксации
- Heat map или тепловая карта
  - Показывает разной температурой цвета распределение плотности фиксаций по изображению наблюдаемого объекта
  - С учётом длительности фиксации или без
  - Разновидность - fog map — наоборот, размывает части изображения, на которые меньше смотрели

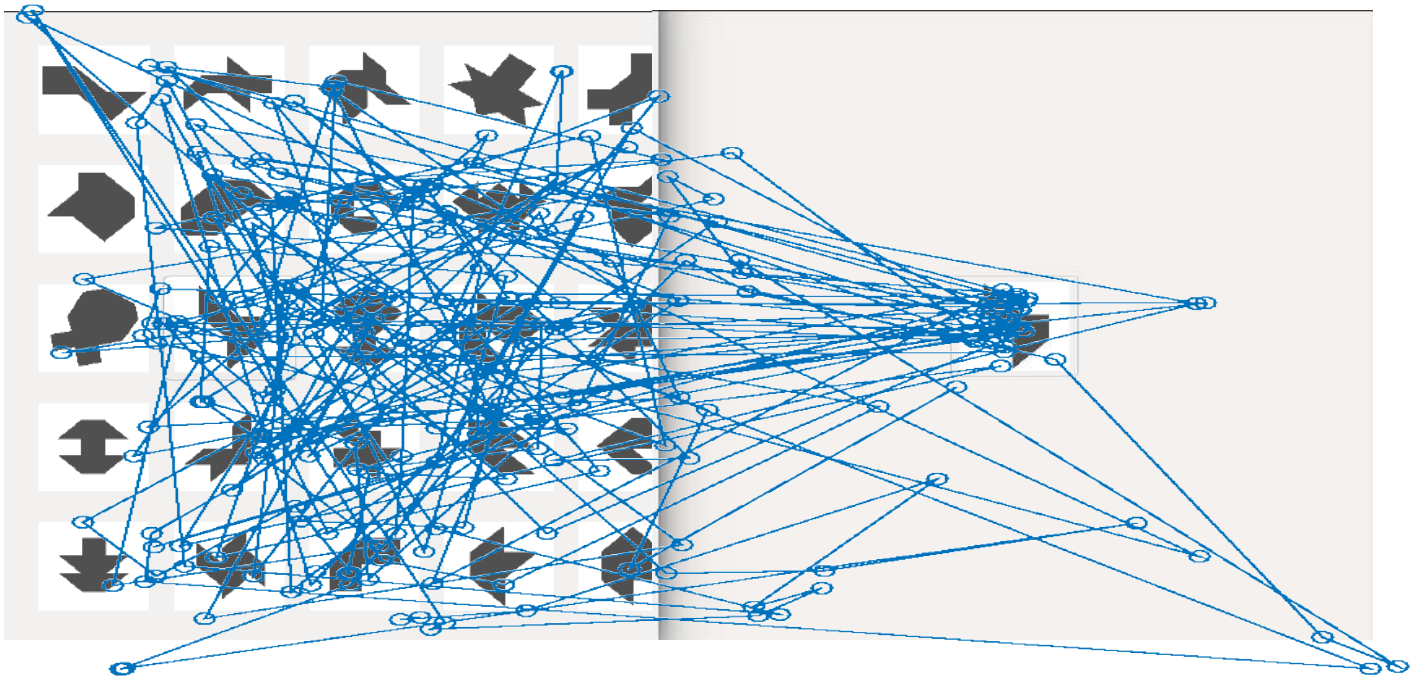
Gaze Plot с помощью graphviz




```
digraph G {
outputorder=edgesfirst
node [shape=circle style=filled
fillcolor=yellow fontsize=40
fixedsize=true width=0.5]
2 [pos = "137,375!"]
3 [pos = "328,371!"]
4 [pos = "114,383!"]
5 [pos = "101,386!"]
7 [pos = "774,337!"]
node [width=1]
1 [pos = "423,372!"]
6 [pos = "38,362!"]
8 [pos = "754,356!"]
1 -> 2
2 -> 3
```

- Применим для статического отображения при небольшом числе фиксаций
  - однако большой лог айтрекера различим только в пошаговом режиме
- Зато несложно сделать скрипт, автоматически превращающий отметки времени в длительность и переводящий наборы координат айтрекера в формат graphviz

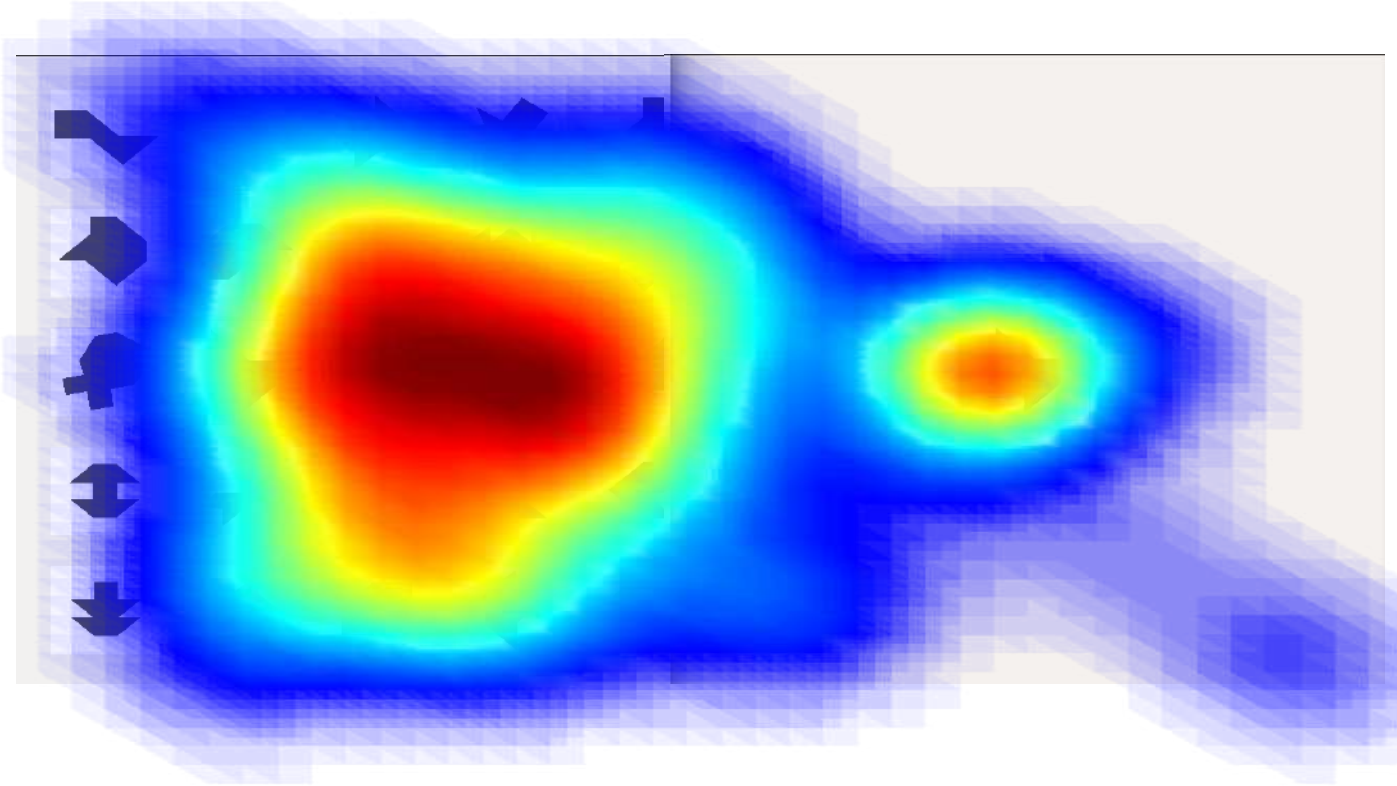
С помощью Octave (простая визуализация)



```
m=csvread('/tmp/sdf.txt');
x=m(:,1); y=m(:,2); z=m(:,3);
plot(x,y,"o-");
```

- большой лог айтрекера превращается вот в это...
- поэтому при анализе протяженных тестов обычно используют тепловые карты

Тепловая карта, построенная с помощью Octave



Генерация heat map:

```
m=csvread('sdf.csv');
x=m(:,1); y=m(:,2);
p= gkde2([x'; y']);
figure(1);
surf(p.x,p.y,p.pdf);
colormap(jet);
shading interp;
grid off; axis off;
view(2);
saveas (1,'sdf.png');
```

В примере использована черная магия, которая называется «бивариантная оценка плотности ядра» (или что-то в этом роде)

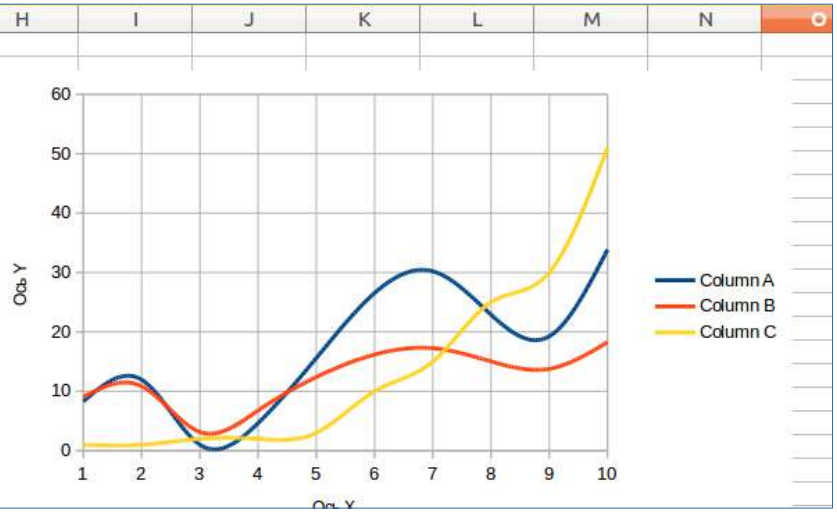
https://github.com/aulloa/Auto_Heat_Mapper — пример её использования, создававшийся для matlab — но в упрощённом виде оно работает в octave (см. код справа)

Интерпретация данных айтрекера на примере реальной задачи - тестировании эргономики табличных процессоров

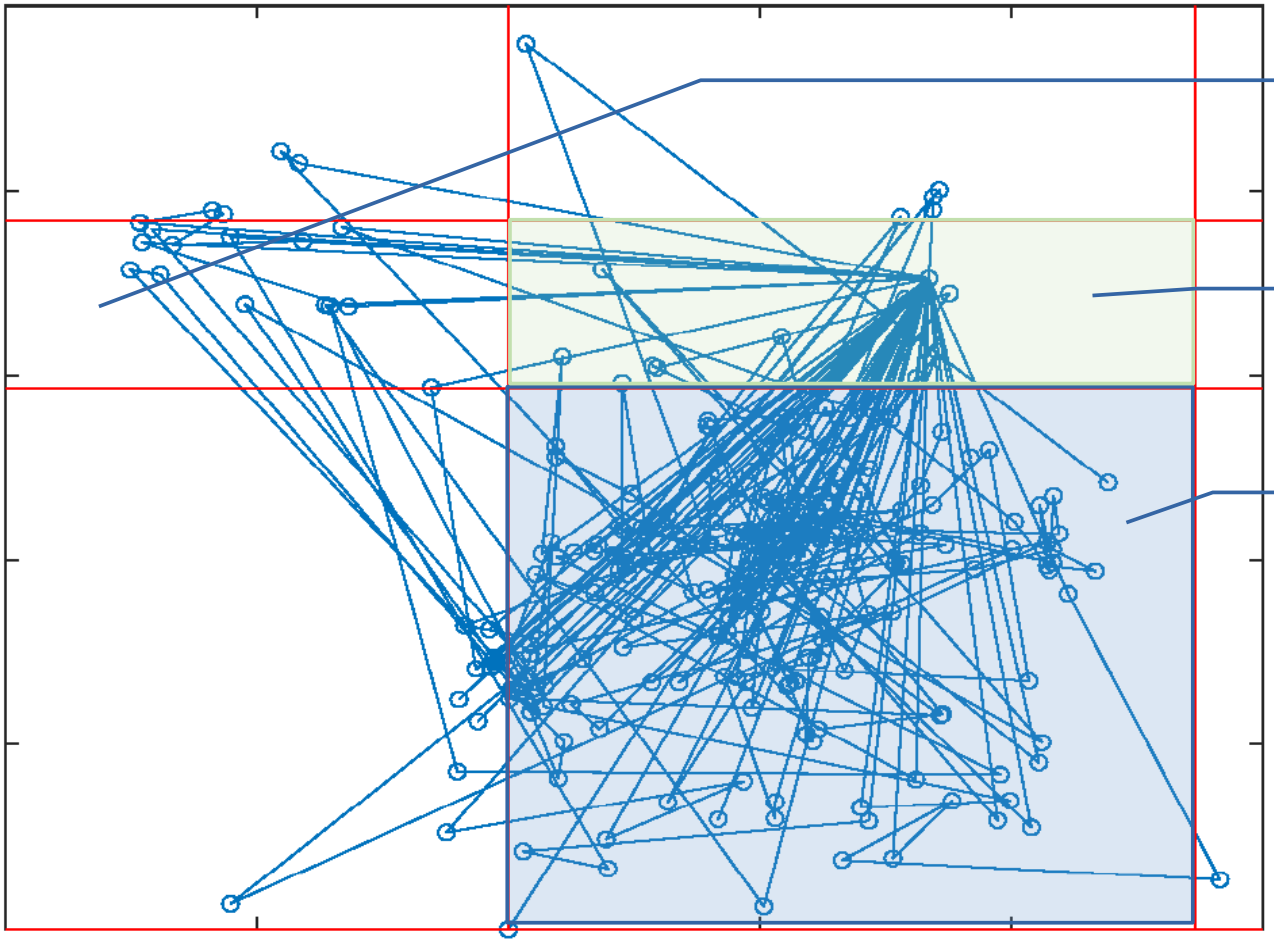
Сравнение табличных процессоров — тестовые задания

	A	B	C	D
1	Цена	Изменённая цена	Процентное изменение цены	
2	1	2	100%	
3	2	4	100%	
4	3	6	100%	
5	4	8	100%	
6	5	10	100%	
7	6	12	100%	
8	7	14	100%	
9	8	16	100%	
10	9	18	100%	
11	10	20	100%	
12	Сумма			
13	55	110		
14	Сред. Знач.			
15	100%			
16				
17				
18				

	A	B	C	D
1	8.3	9.05	1	
2	11.95	10.85	1	
3	1	3.14	2	средне знач
4	4.65	6.77	2	13.16065832
5	15.6	12.40	3	
6	26.55	16.18	10	
7	30.2	17.26	15	
8	22.9	15.03	25	
9	19.25	13.78	30	
10	33.85	18.27	51	
11				
12				
13				
14				
15		**Столбец 1**	**Столбец 2**	**Столбец 3**
16	Среднее	17.425	12.27205963	14
17	Среднеквадр	3.494608943	1.534784907	5.280993173
18	Мода	#VALUE!	#VALUE!	1
19	Медиана	17.425	13.08935019	6.5
20	Первый квар	9.2125	9.498337802	2



Сравнение табличных процессоров — зонирование области обзора



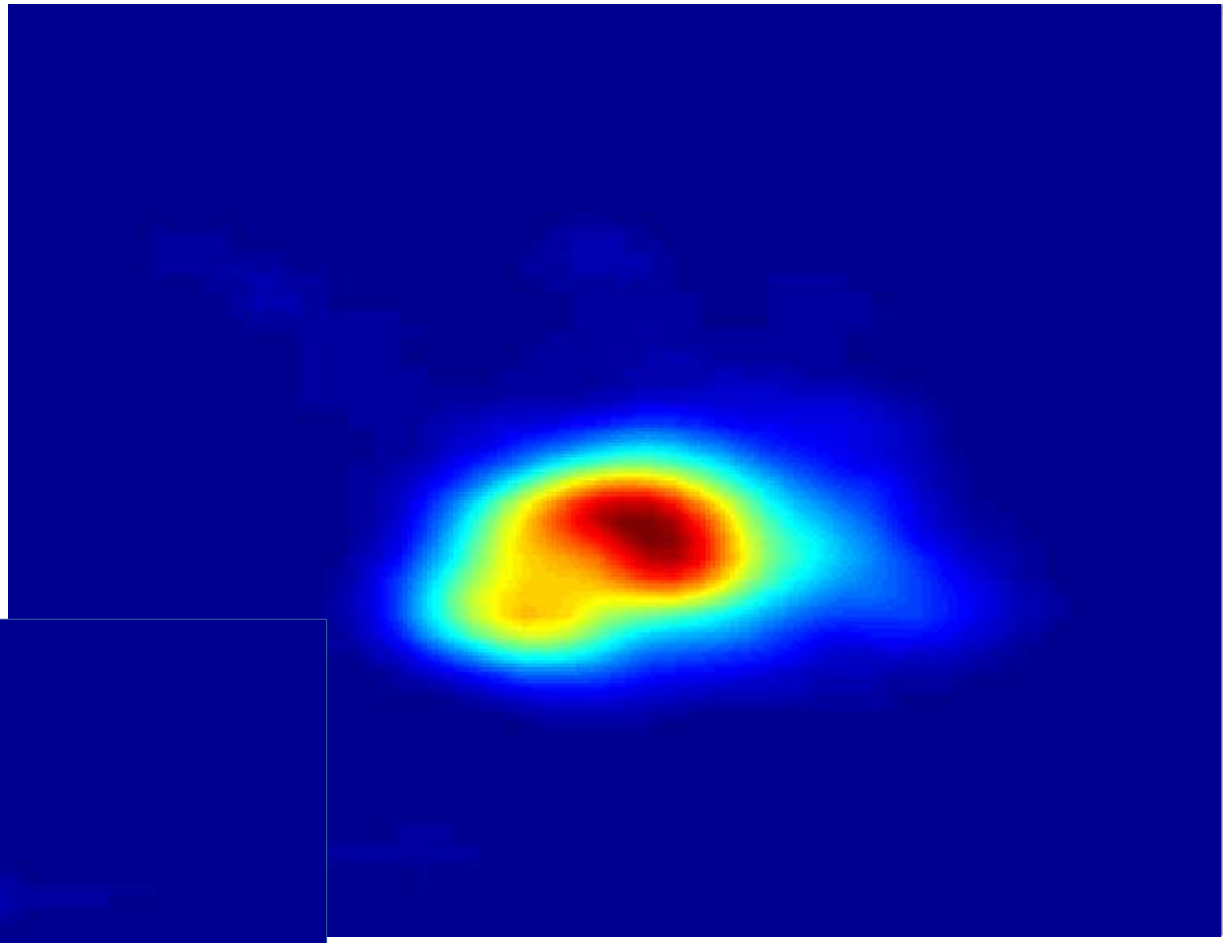
Зоны отвлечения внимания (внешней среды)

Область инструментальной панели

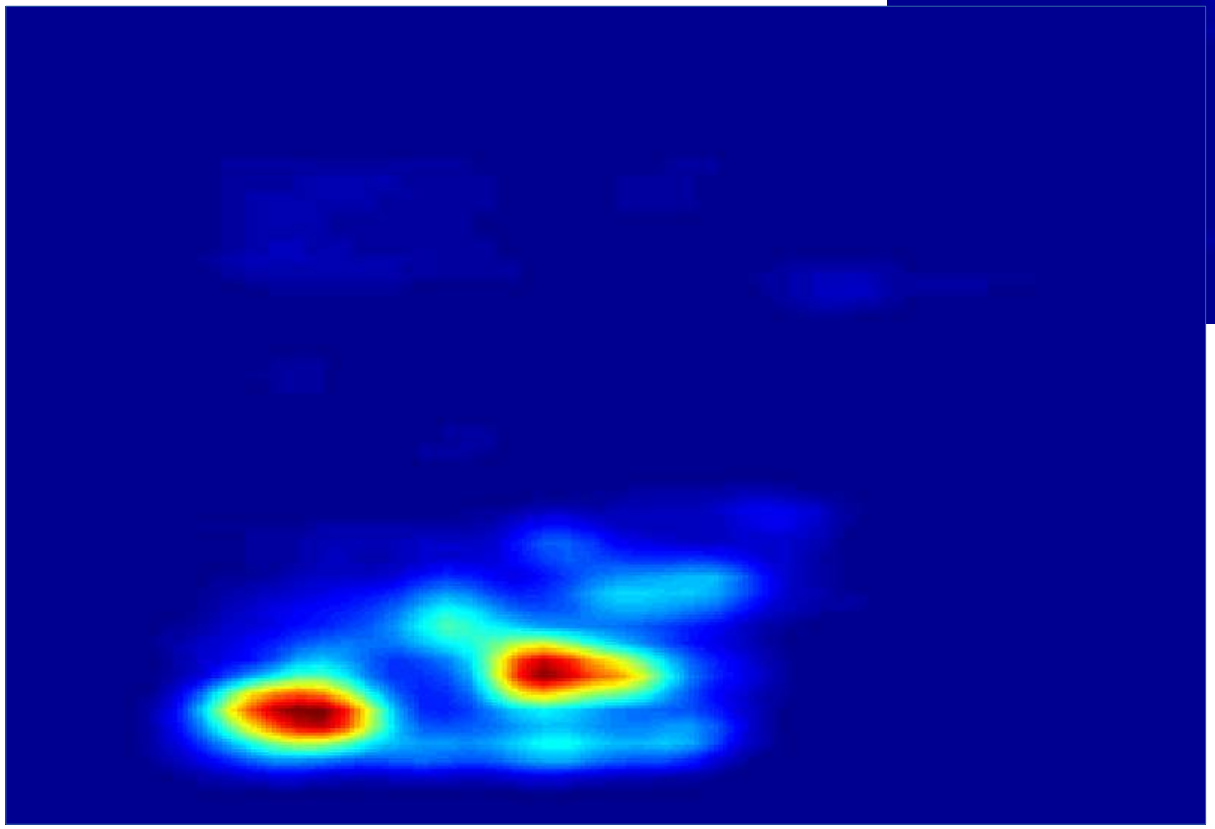
Рабочая область с электронным документом

Тепловые карты фиксации взгляда

Excel:

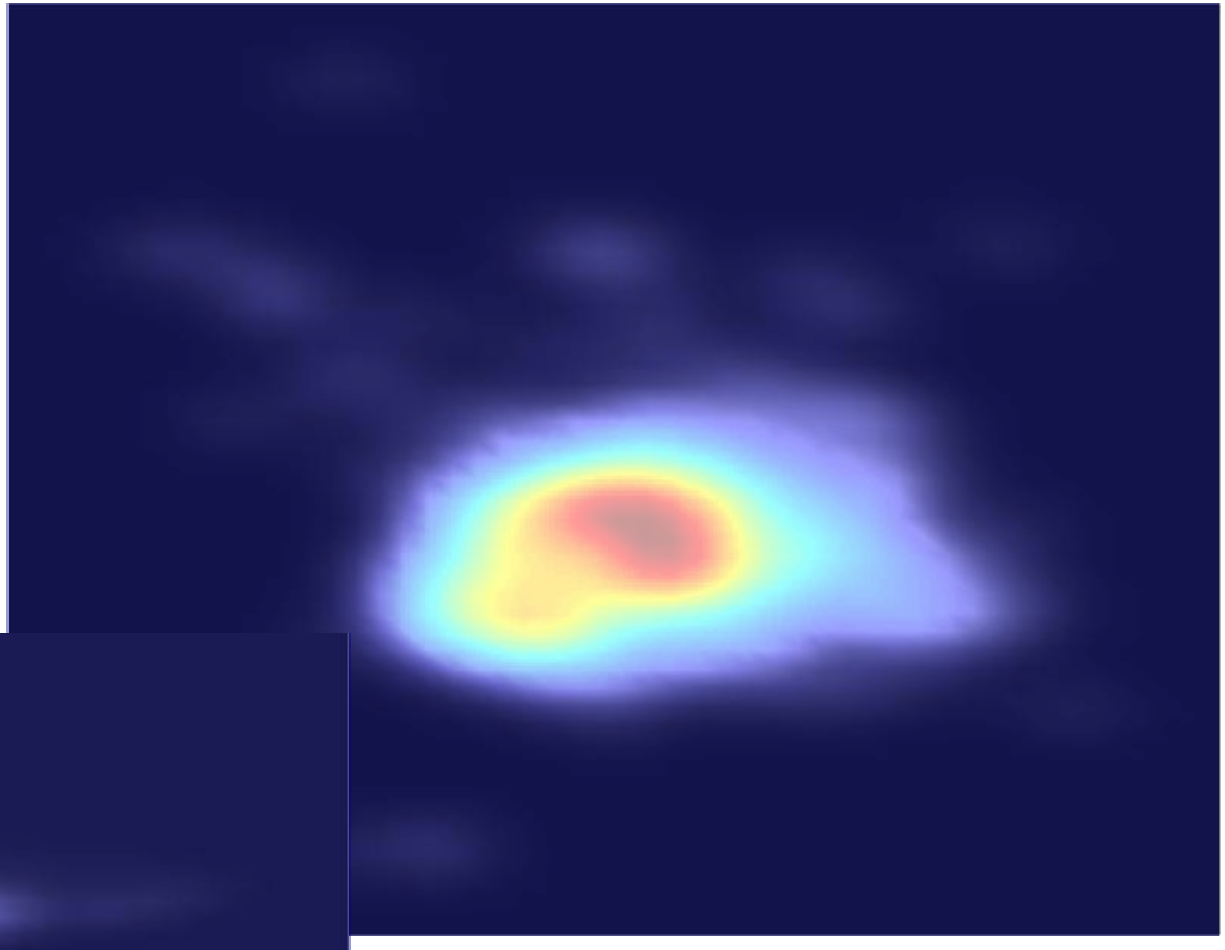


Calc:

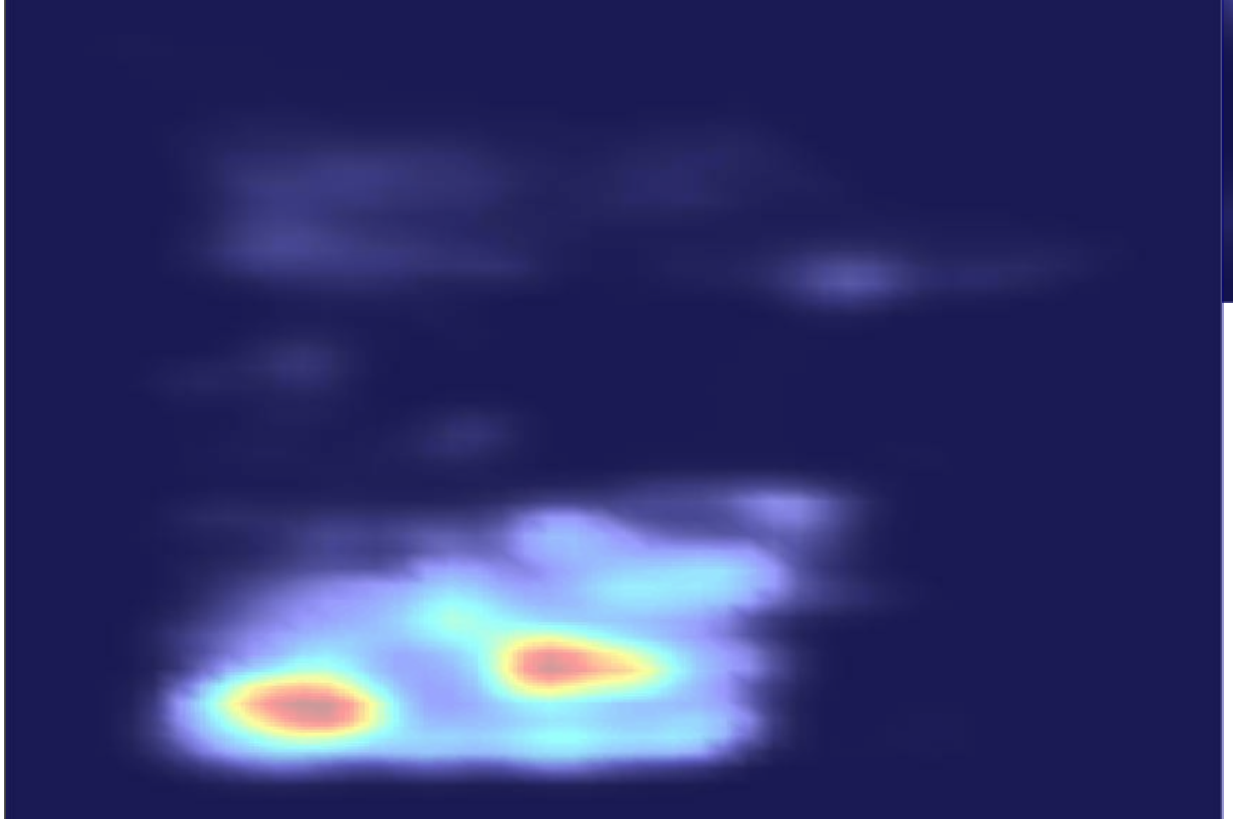


Тепловые карты фиксации взгляда

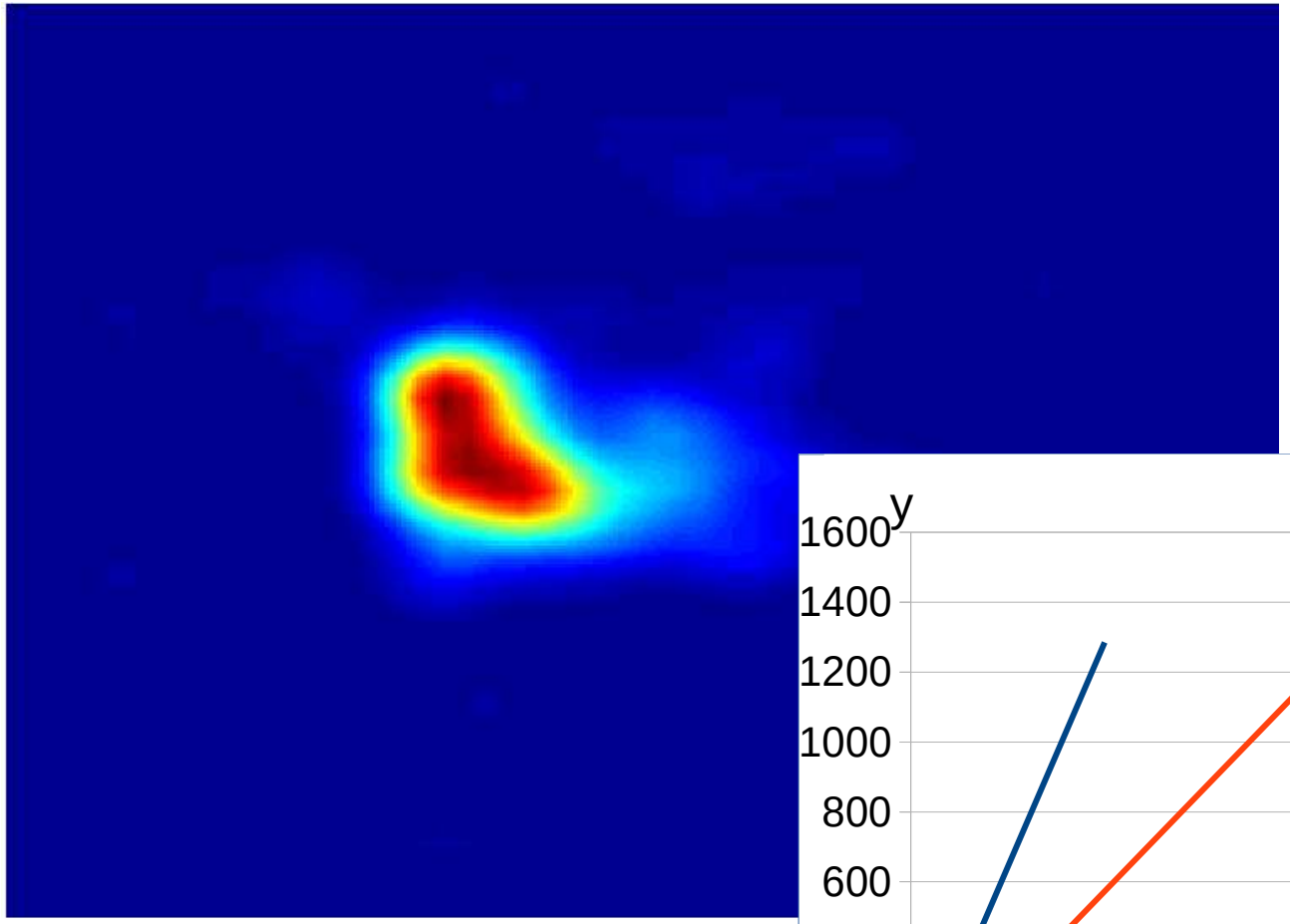
Excel:

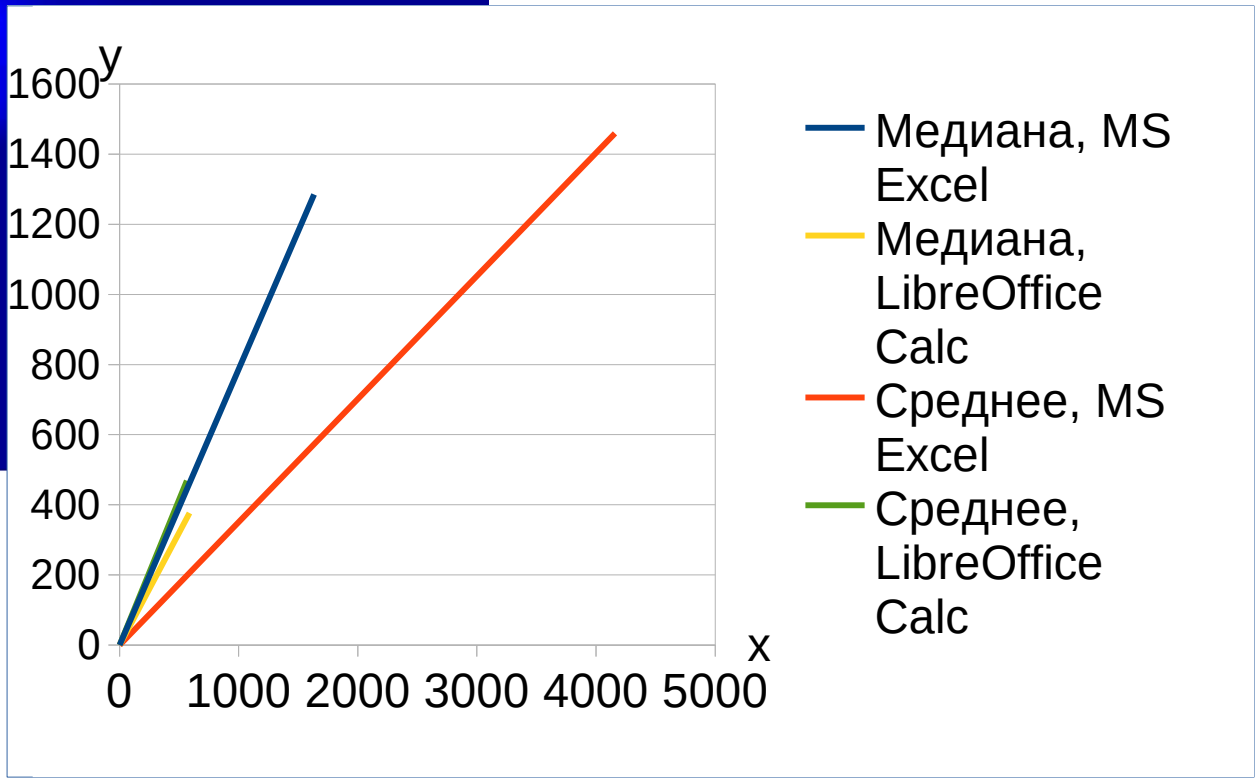


Calc:



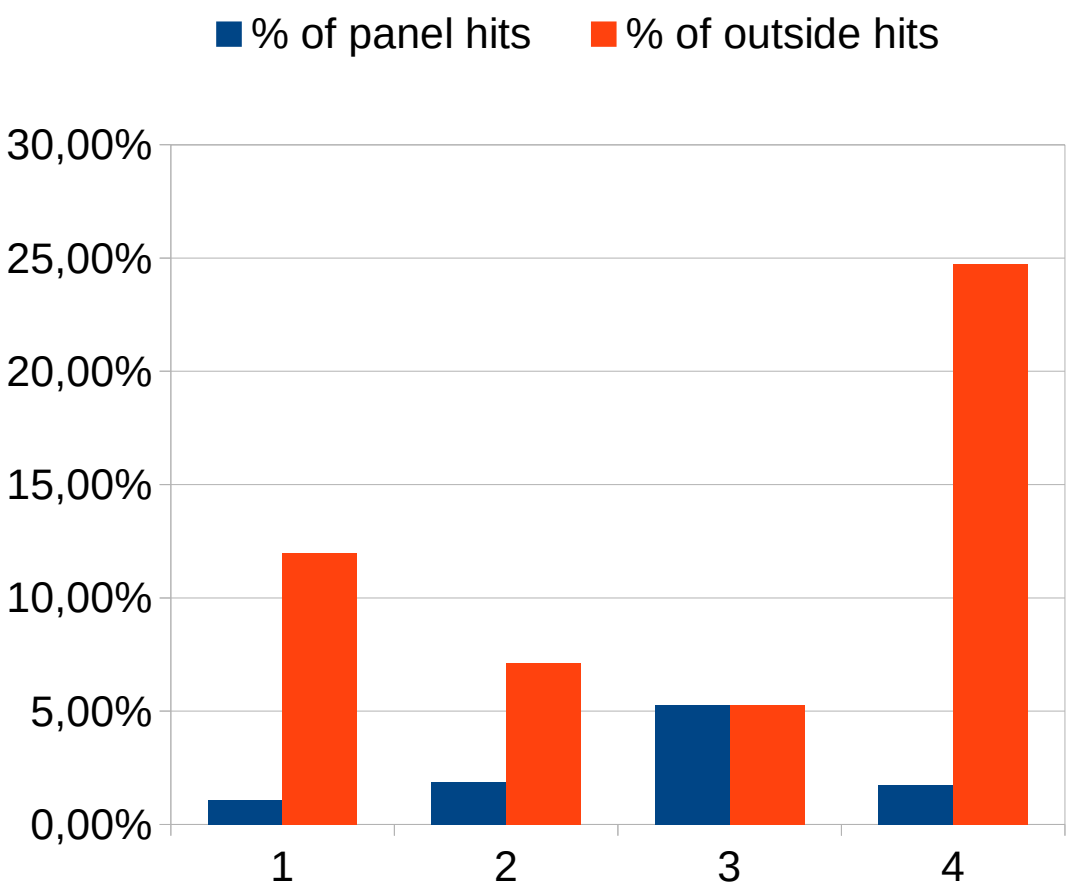
Смещение центра распределения выборки по теплокартам



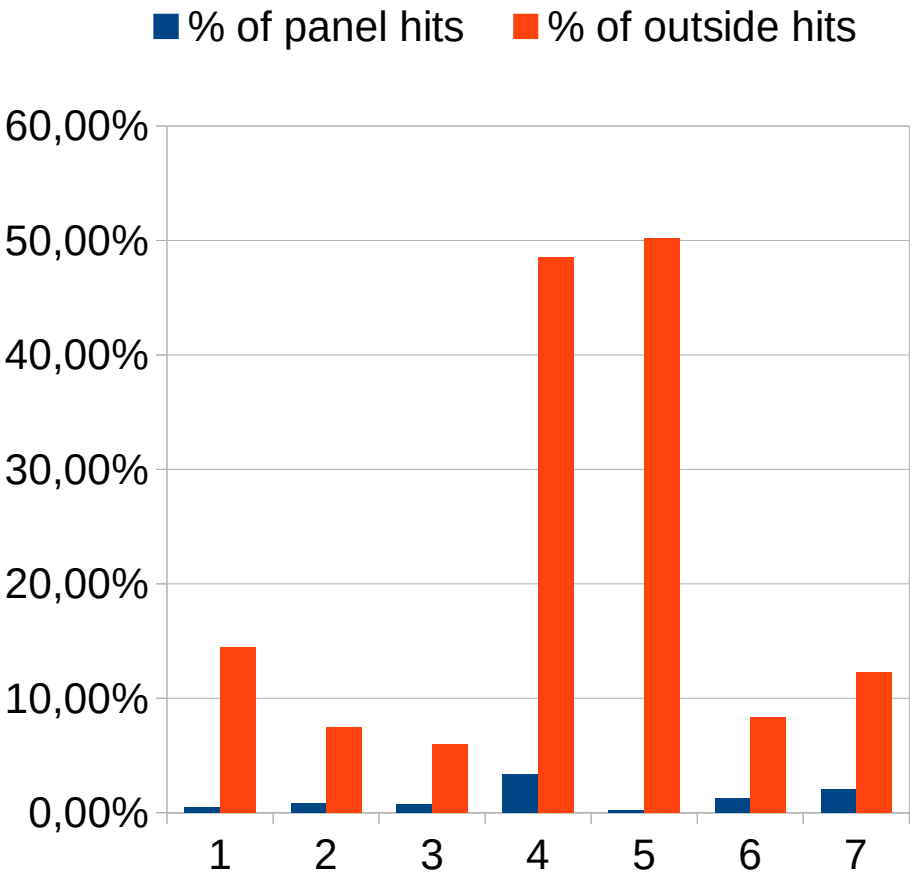


Соотношение фиксаций взгляда на инструментальной панели и области отвлечения внимания

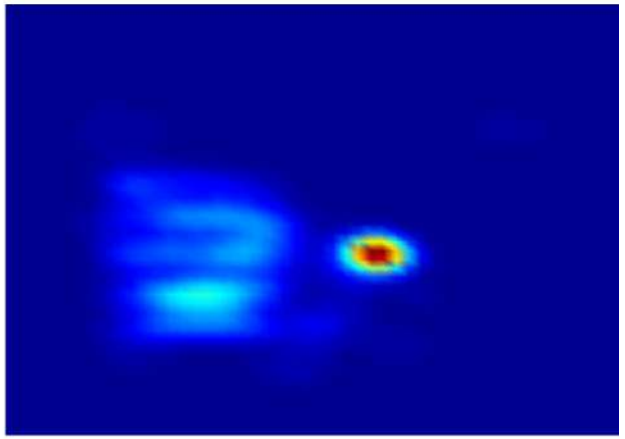
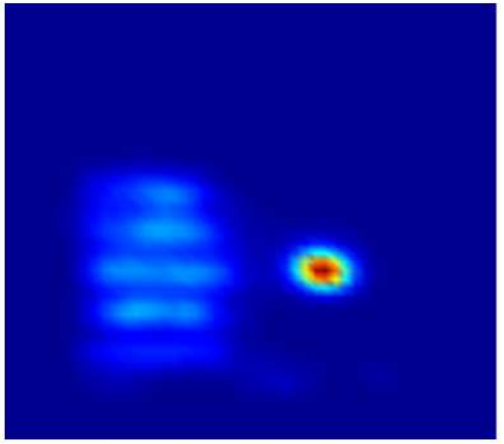
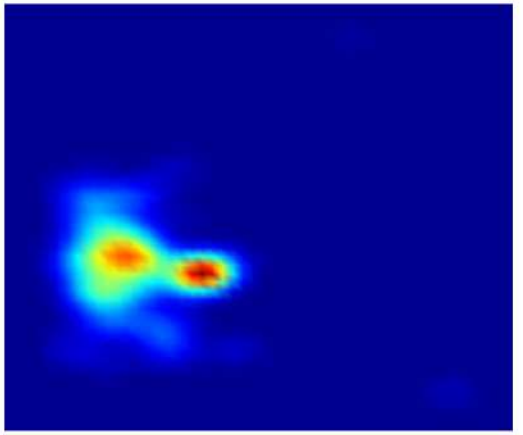
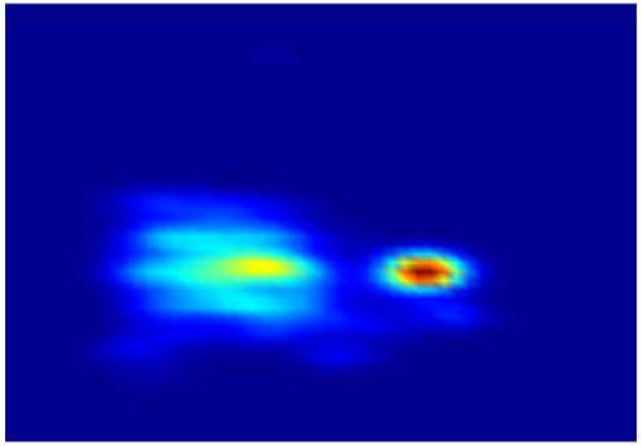
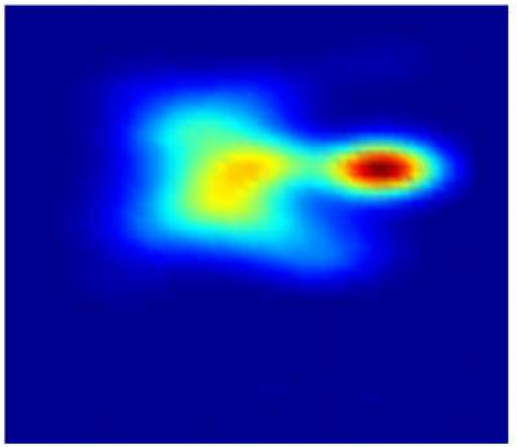
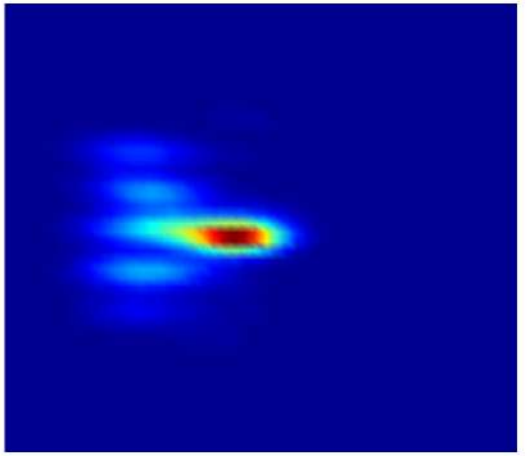
MS Excel



LibreOffice Calc



Еще немного красивых картинок :)

Using eye trackers for the oculographic research

Маркина Анастасия

e-mail: asyamarkina2@gmail.com

Telegram: @asyamarkina