

NSN 7540-01-280-5500

Analyzing Data from the dual Purkinje image (DPI) Eyetracker

Eye Movement Laboratory *Michigan State University*

This documents explains how to analyze eyetracker data in the Michigan State University Eye Movement Laboratory (MSU Eyselab)¹.

These instructions assume:

1. Your data was collected on the dual-Purkinje eyetracker.
2. You were presenting your stimuli in graphics mode.
3. You were running your experiment under the MSU version of the PCEXPT system.
4. You did not do anything idiosyncratic, such as change the format of the data files.
5. You intend to analyze the data using the PICDRY analysis package.

For technical information concerning these data analysis, see the Technical Details section at the end of this document.

Note that if you ran your experiment in text mode, then you must use the old EYEWASH and EYEDRY analyses, which are not documented here. (And if you use them, then you get what you deserve.)

DATA COLLECTION

- 1. Compile your Program:** Compile your experiment running program on the computer that is interfaced with the dual-Purkinje image (DPI) eyetracker using the instructions posted there.
- 2. Choose a Pixel Criterion:** When you run your experiment, you must decide what range of pixel values you want to treat as equivalent during data collection. This is the "pixel criterion" value that you type into the program when you run your subjects. The pixel criterion value determines how many pixels the eye can "slide" and still be taken to be part of the same fixation. Generally, you want to set this value at *4 pixels* (as in Kronos), which gives a range of +/- 4

¹This document was prepared by John M. Henderson and Gary Schrock. The analysis programs documented here were created with support from the Army Research Office (DAAH04-94-G-0404) to John M. Henderson, and the National Science Foundation (SBR-9319272) to Fernanda Ferreira. For further information, contact John M. Henderson, Department of Psychology, 129 Psychology Research Building, Michigan State University, East Lansing, MI 48824-1117, or email john@eyelab.msu.edu.

pixels from the first pixel value for the current fixation (i.e., a range of 9 pixels vertically and horizontally). This translates into about 10' of arc at the current viewing distance.

DATA ANALYSIS

There are two main steps to analyzing eyetracker data.

1. Run your raw data files through EYEDROP to remove the time the eye is rapidly moving from one position to another (i.e., the saccades).
2. Run PICDRY. The input for PICDRY is the data files created by EYEDROP. PICDRY is the program that actually analyzes the fixation data.

EYEDROP

The purpose of eyedrop is to remove the saccades from the data files. The program goes through the data file and removes any sample that is under some specified value on the assumption that such short samples are parts of a saccade. Generally, you want to set this value to 15 ms. Using this value along with a value of 4 as the pixel criterion in the running program means that the eyes have to travel at least 5 pixels in 15 ms to be dropped. This does a good job of removing saccades and some of the overshoot seen at the end of saccades.

The 15 ms value is appropriate when the display monitor is at the far position in the DPI room. For changes in visual angle (if the monitor is moved), this value may need to be modified. Since each pixel is about 1.18 minute of arc at that distance, this translates into a movement of about 5.92' in 15 ms = .0987° in 15 ms = 6.58°/sec. Note that if the monitor were moved closer, then the eyes would have to be moving faster to travel 5 pixels in 15 ms. Thus, the eyedrop value should be changed (decreased) on the command line when the monitor is closer.

Note: You should always run EYEDROP before PICDRY (except in the case of the Ante analysis). If you don't, then PICDRY is likely to believe that each trial has just one very long fixation.

To Run Eyedrop:

1. **Create data.lst:** Create a file that contains the names of all of the files you want to analyze. The easiest way to do that is to type the following at a dos prompt:

```
dir /b fname*.dat >drop.lst
```

This will copy the names of all the files in the current directory with the name fname*.dat into a new file called drop.lst. Otherwise, you can just make a list in an ascii editor like dos edit or notepad in windows.

A good practice is to call this .lst file the same as the experiment name plus drop, e.g., for the praha experiment, *prahadrp.lst*²

An example prahadrp.lst file for four subjects in the experiment praha would be:

Praha1.dat
Praha2.dat
Praha3.dat
Praha4.dat

2. Run Eyedrop: To run eyedrop, type the following:

Eyedropl drop.lst 15 da1 800 600

The general version is:

Eyedropl drop.lst cutoff new_extension max_x_value max_y_value

The parameters are:

drop.lst: The name of the file that lists the names of the data files you want to eyedrop.
See Step 1 above.

Cutoff: time in milliseconds equal to or below which sample times will be dropped.
Typically 15.

New_extension: the extension for the created files. This is typically da1. Note: don't include the dot-- it'll be added automatically.

Max_x_value: the maximum allowable x screen value in pixels (typically 800).

Max_y_value: the maximum allowable y screen value in pixels (typically 600).

If the drop.lst file used as the example above were used as input, the output would be four new files each of which now had any sample under 15 ms excluded. These files would be called:

² Praha is the laboratory name of an eyetracking experiment that used many of these analyses. See: Henderson, J. M., McClure, K., Pierce, S., & Schrock, G. (in press). Object identification without foveal vision: Evidence from an artificial scotoma paradigm. *Perception & Psychophysics*.

Praha1.da1
Praha2.da1
Praha3.da1
Praha4.da1

PICDRY

PICDRY is the program that actually analyzes the eyetracker data by scoring regions. The possible analyses include first fixation duration, gaze duration, and total fixation time, along with a number of other analyses that are described below. This program is extremely flexible and allows you to examine many different aspects of the eye movement record. You are encouraged to use as many of these analyses as seem relevant for your data set. PICDRY currently has the capability of analyzing the data over four different kinds of scoring regions:

1. Rectangular regions that are constant across condition and item.
2. Rectangular Regions that are different for each condition/item combination.
3. Disk/Circular shaped regions that are constant across condition and item.
4. Disk/Circular shaped regions that are different for each condition/item combination.

To Run Picdry:

1. Create fname.lst. Create a new subject list file that contains a list of the names of the data files that you want to analyze. These files are typically the da1 files you created with eyedrop. So, you can use the same method as described above to create this file:

```
dir /b*.da1 >praha.lst
```

Or, again, you can use an ascii editor to create the file.

It is a good practice to call this by your *expt.lst*, where *expt* is the name of your experiment. For example, *praha.lst*.

An example praha.lst file for the praha experiment with four subjects would be:

Praha1.da1
Praha2.da1
Praha3.da1
Praha4.da1

2. Create the region file. The region file defines regions that you're interested in analyzing with PICDRY. These regions are specified by pixel coordinates in the two dimensional space of the display monitor. A good practice is to call this file *expt.reg*, where *expt* is the name of your experiment (e.g., *praha.reg*). The type of region you want to analyze (rectangle or circle/disk) is

specified when you run PICDRY (see below), and must match the nature of your region file. The region file can be one of four different formats, depending on the type of analysis you want to do.

Format 1: Regions are rectangular in shape and are constant across all items and conditions.

The region file for this type of analysis contains a single number on the first line indicating how many regions there are. All other lines contain the coordinates for each of the rectangular regions. One region should be specified on each line, with the values (in pixels) of the upper left x, upper left y, lower right x, and lower right y. The pixel values should be separated by spaces.

Note that the scoring region includes the boundary of the region.

An example from the Praha series of experiments, with a linear array of line drawings, is:

```
5
134 200 160 400
260 200 286 400
387 200 413 400
513 200 539 400
639 200 665 400
```

Format 2: Regions are rectangular in shape, but differ in position and/or shape by condition/item.

The region file for this type of analysis contains a zero (0) on the first line to flag that regions differ by condition/item. The next line will be item, condition, number of regions for that condition/item, the Upper Left x, Upper Left y, Lower Right x, and Lower Right y, all separated by spaces and in pixel values, for each region in that condition/item combination. The scoring region includes the boundary of the region.

There are several things to keep in mind with this type of analysis:

A. You must independently specify the regions for every condition/item pair in the experiment, even if some repeat.

B. If you define more than one region per condition/item pair, the first region specified will be analyzed as similar across conditions and items, as will the second, etc., so similar regions must be aligned ordinally in the file. In the subject data file output, you get condition by region output, formatted as:

```
condition 1 region 1, condition 1 region 2, condition 1 region 3 ... condition 1
region n, then condition 2 region 1, condition 2 region 2, condition 2 region 3 ...
condition m region n.
```

C. As with all of the analysis programs, you specify the condition number and then the item number. For example, 1 2 means condition 1, item 2. These should be indicated in the first two columns.

The following example shows a part of the region file from the Kronos5 scene viewing experiment. This is the first two items. Note that the regions in conditions 1 and 2 are the same, but still must be specified independently:

```
0
1 1 1 521 282 578 375
1 2 1 303 272 364 391
2 1 1 521 282 578 375
2 2 1 303 272 364 391
```

Format 3: Regions are either a circle or disk and are constant across all items and conditions.

The region file for this type of analysis contains a single number on the first line indicating how many regions there are. All other lines contain the coordinates for each of those regions. One region should be specified on each line, with the values (in pixels) of the center x, center y, inner radius of disk, and outer radius of disk. The pixel values should be separated by spaces. To specify a circle, make the inner radius 0. Note that you can specify concentric disks using this format. (This was done in the landing position analysis of Praha.) The scoring region includes the boundary of the region.

The following example gives two circular regions at screen coordinates 250, 300 and 650, 300 each with a radius of 50 pixels:

```
2
250 300 0 50
650 300 0 50
```

The following example gives two concentric scoring regions, with an inner circle and an outer disk, at the center of a 600 x 800 pixel display:

```
2
400 300 0 50
400 300 50 100
```

Note that the outer boundary of the inner circle and the inner boundary of the outer disk have the same value. This setup is needed to ensure that fixations don't "slip through the cracks" between the regions. However, it should be noted that there is an extremely small chance that a single fixation will fall exactly on the boundary and so will be included in both regions.

Format 4: Regions are a circle or disk, but differ in position and/or shape by condition/item.

The region file for this type of analysis contains a zero (0) on the first line to flag that regions differ by condition/item. The next line will be item, condition, number of regions for that condition/item, the center x, center y, inner radius of disk, and outer radius of disk, for each region in that condition/item combination. The pixel values should be separated by spaces. To specify a circle, make the inner radius 0. Note that you can specify concentric disks using this format. The scoring region includes the boundary of the region.

Note the caveats described under Format 2.

The following example gives two circular scoring regions for each condition/item combination:

```
0
1 1 2 250 300 0 50 650 300 0 50
1 2 2 300 300 0 50 700 300 0 50
2 1 2 250 300 0 50 650 300 0 50
2 2 2 300 300 0 50 700 300 0 50
```

3. Run PICDRY

To run PICDRY, go to a dos prompt and type:

picdry

You will be prompted to answer a number of questions concerning your analysis. The following example goes through these questions and provides potential answers. In this example, the questions asked by the program are underlined, and the answers you type in are *italicized*. Notes are in normal text.

Load config file?

y(or n)

This question is asking if you want to load a file that contains the answers to all of the questions that PICDRY needs to ask. If you answer *y*, you will be prompted for the name of this file (typically a file with a .cfg or .crn extension). You will only have one of these files if you have previously run *picdry* and stored your answers.

Smallest experimental item number:

1

What is the smallest item number you want to analyze? Typically, you would type *1*, but if you want to analyze a subset of the items, you can start at some other item number.

Largest experimental item number:

1000

What is the largest item number you want to analyze? Typically, you would type the largest item number in your stimulus file, but you can analyze a subset of items by modifying this value.

Smallest condition number:

1

What is the smallest condition number you want to analyze? As with item number, you would typically use 1, but you can analyze a subset of conditions by choosing a higher value.

Largest condition number:

1000

You should be getting the hang of this by now.

Collapse across any conditions?

y (or n)

This question is asking whether some conditions that have different condition numbers should be treated as a single condition. For example, perhaps you coded yes/no trials as separate condition numbers for the sake of analyzing the answers, but you want to combine across those conditions when analyzing the eyetracker data. This lets you do that.

If you say yes, you will be prompted with:

Conditions are mapped as follows:

1	2
1	2

Condition number to change (0 to finish):

1

This output shows that condition 1 is mapped to condition 1, and condition 2 is mapped to condition 2. In other words, the mapping is shown vertically. Type the condition number you want to remap, and then you'll be prompted:

Condition number to remap to:

1

Type the condition number you want to remap your specified condition to. In this example, I've remapped condition 1 to condition 1, so nothing will change.

You will then be shown the current mapping. Type 0 when you want to stop remapping.

Smallest fixation time to keep:

90

This question is asking whether you want to impose a lower cutoff on the data. Any fixations shorter than the value specified here will be dropped (i.e., won't be included in computing fixation durations or any other analysis value). Typically you will use 90 ms to toss out fixations that are unlikely to be under cognitive control. If you specify 0 no data will be excluded based on the low cutoff.

Largest fixation time to keep:

1000

This question is asking for an upper bound on fixation duration. Any fixations larger than the value specified here will be dropped. Typically you want to use a value of 1000 ms or lower. The best way to determine this value is to run an histogram analysis (see below), and base your high cutoff on the observed distribution of fixation durations.

Region type (1: rectangle; 2: circle):

1

Your answer to this question determines whether your regions will be a rectangle or a circle/disk. Your answer to this question must match your region file.

Filename of region template:

expt.reg

Type the name of the region file that you created above.

Filename of the list file:

expt.lst

This is the name of the file with the list of data files in it that you created above. Note that this should be the file that contains the list of eyedropped file names (the da1 files). The raw data files must always be run through EYEDROP to remove saccades before the data is subjected to PICDRY.

Bin size (for hist anal):

20

This question is asking whether you want a histogram of fixation durations, and if so, how big of a bin (in ms) you want to use to generate the histogram. The output is an absolute count of the number of fixations in each bin.

If you don't want to do a histogram analysis, type *1*. Any other answer will give a histogram with bins of that size provided that you specify the histogram analysis later. I don't know what would happen if you type *0*, but it probably wouldn't be good

It is recommended that you run the histogram analysis in it's own separate run of PICDRY because it is very memory intensive.

Toss incorrect trials (y/n)?

y (or n)

This question is asking whether you have some indication of bad (incorrect) trials in your data file. For example, if you had a question appear after each trial, you may want to exclude data from trials in which the answer was incorrect.

Note that the program assumed that if you answer *y* here, then your trial structure is such that the answer appears on the next line in the data file.

If you answer *n*, nothing else happens. If you say *y*, then you are given the following message:

Note: Questions are assumed to be the next line in the data file!

Position of response:

7

Provide the column position of the correct/incorrect response. This will almost always be column *7* unless you modified the data acquisition module.

Correct response:

1

What value in column *7* indicates the trials to retain? This value is typically *1*.

After you answer the last question, you will get a view of all of your answers, and you can change any mistakes. Type the number associated with the answer you want to change, then hit enter. You're prompted to enter the new value.

Note: There are four values that will appear in the summary answer list that you didn't specify. These are condition number column position, item number column position, column position of the number of fixations, and column position of the start of the eyetracker data. These have

default values, and as long as you use standard PCXPT eyetracking data files, all is well. The only time you might need to change something is if you accidentally coded the condition and item numbers backwards. (Normal coding in stimulus file: condition number, item number).

When you're finished changing values: type 0, then enter to move on to specify your analyses.

You will be prompted with a list of the analyses you can do. Possible analyses include:

1. First Fixation
2. Gaze Duration
3. Total Time
4. Second Pass
5. Fixation Count
6. Entry Analysis
7. Gaze Fixation Count
8. Saccade Matrix
9. Percent Entered
10. Histogram Analysis
11. Percent Correct
12. Ante Analysis
13. Kronos Analysis

The meaning of each of these analyses is as follows:

1. First Fixation: Duration of initial fixation in a region.
2. Gaze Duration: Total fixation time in a region from first entry to first exit.
3. Total Time: Total fixation time in a region across all entries.
4. Second Pass: Total fixation duration from second entry to second exit.
5. Fixation Count: Total number of fixations in a region (i.e., the number of fixations the durations of which are added together to give the total time measure).
6. Entry Analysis: Total number of entries into a region.
7. Gaze Fixation Count: Number of fixations that are being summed to produce the gaze duration in a region.
8. Saccade Matrix: Matrix of where the eyes moved from/to. Not currently implemented.
9. Percent Entered: Percentage of trials on which the region was entered.
- 10 Histogram Analysis: Histogram of fixation durations in regions. It is recommended that this analysis be run separately from the others because it is memory intensive. Histograms by both subject and item can be generated. Subject file will be set up as: Subject 1, Condition 1, Region 1; Subject 1, Condition 1, Region 2; Subject 1, Condition 1, Region n; Subject 1, Condition 2, Region 1; Subject 1, Condition 2, Region 2; etc. Item file will be set up the same way, with item replacing subject.
11. Percent Correct: Percentage of correct trials based on your "correct" trial value.
12. Ante Analysis: This option conducts analyses that are specific to the antisaccade experiments

- (see Karen Butler for details). These analyses will be explained in another document.
13. Kronos Analysis: Number of fixations until a region is entered. The value includes the first fixation in the region. Note that the first fixation on the scene is counted as fixation 1 (i.e., if the first fixation on the scene when it is displayed is in a scoring region, that fixation will be given a value of 1. So to ask what percentage of trials were fixated following the first saccade in the scene, you want to use ordinal fixation 2 as the critical fixation number (the “correct” response in MRT3.). This analysis generates a new set of data files, which can then be analyzed by MRT3 to determine mean number of fixations until region fixated, or percentage of trials fixated by a certain number of fixations (e.g., On what percentage of trials was a region fixated on the first fixation following the initial fixation on the scene?) Note: the initial fixation position is coded as fixation 1,

To do one of the above analysis, type the number. For example, type *1* for a first fixation duration analysis. For each analysis type, you will be prompted for the following information:

File to store analysis:

exptffd.sub

Type the filename for the data by subjects, collapsed over items. Typically, you want to use a file name that includes your experiment name, the analysis type, and the fact that it is a subject analysis. For example, for praha, the file is *prahaffd.sub*.

Item by item analysis?

y

Do you want to analyze the data by items collapsed across subjects? If so, answer *y*, otherwise *n*. If you answer yes, you will be prompted to provide the file name for the items analysis. Note: You cannot currently do an items analysis without also specifying a file name for the subject analysis. In other words, if you want an items analysis, you must also do a subjects analysis.

File to store item by item analysis:

exptffd.itm

Type the filename for the data by items, collapsed over subjects. Typically, you want to use a filename that includes your experiment name, the analysis type, and the fact that it is a items analysis. For example, for praha, the file is *prahaffd.itm*.

Output number of samples per value:

y (or *n*)

This question is asking whether you want an output of the number of trials that went into producing the means for the analysis you are currently specifying. In other words, this

analysis outputs the number of trials that are included in the current analysis for each subject as a function of condition and region. In the current example, this would be the number of trials used to compute the mean first fixation duration. If you answer yes, you are prompted to provide the name of the file for these values:

File to store sample count:

exptffd.cnt

Type the filename for the count data for this analysis. Typically, you want to use a file name that includes your experiment name, the analysis type, and the fact that it is a count analysis. For example, for praha, the file is *prahaffd.cnt*.

You will now be returned to the screen indicating the analyses possible, with your answers shown next to the types of analyses you have chosen. Repeat as necessary for additional analyses. Note that you can remove an analysis you have previously chosen by typing the number of that analysis.

****Note:** There is currently a bug that if you specify you don't want a samples file, you won't be able to get an items file.

Finally, when you're ready to run the analyses, type 0 to run. (This is currently undocumented.)

TECHNICAL DETAILS

Data Acquisition

The analog values produced as output by the DPI eyetracker are digitized during data collection using the Data Translations DT2802 A-D board running in a Pentium-90 based computer. Sampling rate is based on the polling mode of the A-D board, and is greater than 1000 Hz under current implementation. Nominal output values from the eyetracker should be greater than ± 5 millivolts vertically and horizontally over the calibration range.

The digital x,y values are converted into pixel values on-line by the MSU PCEXPT data acquisition software. Under the current implementation of the acquisition software, the possible screen coordinates fall within the 800 x 600 pixels available in SVGA mode, though the analysis software is resolution-independent.

During data acquisition, the x,y screen coordinate of the current sample is compared to the x,y screen coordinate of the last sample. If the samples differ by less than the pixel criterion (specified at run-time) in city block units, then the current sample is taken to be part of the same acquisition unit, and no new data point is generated. This procedure continues until a sample falls beyond the pixel criterion. At that time, a new data point is generated. The use of a pixel criterion of about $\pm 5'$ of arc (4 pixels at a viewing distance of 1 meter) helps to smooth over system noise while losing relatively little spatial or temporal resolution. The pixel criterion also reduces the size of the data files considerably because multiple samples can be considered part of the same acquisition unit and so can be coded as a single data point. Note that a pixel criterion of 0 would provide no smoothing, would lose no resolution, but would lead to enormous data files and would probably cause the program to run out of memory.

Data files produced by the data acquisition software includes trial structure information (e.g., condition and item number, ordinal presentation number of trial), as well as eye movement data. The eye movement data includes the following information for each trial: total number of data points for that trial, x position of current data point, y position of current data point, start time of current data point, end time of current data point. For example, a single data point might look like this:

423 322 12335 12423

which would indicate an eye position at 423,322 on the display (± 4 pixels) that lasted 88 ms.

Thus, aside from the smoothing provided by the pixel criterion, all spatial and temporal information about the eye's behavior during each trial, including movement, is stored.

Data Analysis

Fixation locations and durations are computed from the raw data files in the following manner:

EYEDROP deletes any data point that has an elapsed time of less than the specified value. Using a 15 ms cutoff value in EYEDROP deletes any data point where the eye moved through the *pixel_criterion* number of pixels in less than *cutoff* time. Using typical values, this means that any data point indicating that the eyes moved through 5 pixels in less than 15 ms will be dropped.

Using the following values leads to treating any horizontal eye movement of greater than 6.58°/sec as a saccade and any movement less than that value as part of a fixation:

1. Eye to monitor distance (for the NEC XE 15 monitor) = 1 meter.
2. Pixel criterion = 4 pixels.
3. EYEDROP cutoff = 15 ms.

The estimated velocity comes from the following values: Each pixel subtends about 1.18 minute of arc at 1 meter, so 5 pixels in 15 ms translates into a movement of about 5.92' in 15 ms = .0987° in 15 ms = 6.58°/sec. Observation of raw data files suggests that this velocity removes the beginnings of saccades, as well as the "overshoot" at the ends of saccades observed in the record generated by the DPI, but does not remove time that should be included at the beginning and end of the saccade.

Each data file produced by EYEDROP contains *chunks* of temporally contiguous position samples (i.e., within 1 ms) that are separated by temporal holes. The holes are created by the missing samples that were removed by EYEDROP and are assumed to be saccades.

The chunks that remain in the eyedropped files are periods of time when the eye was moving relatively slowly, and are assumed to reflect fixations. The duration of a fixation is computed as the duration of the chunk. This is equivalent to the time that elapsed between two consecutive saccades.

Because "fixations" are actually made up of slow drifts, microsaccades, and nystagmus, the position values tend to change within each fixation chunk. There is, therefore, a question of how to determine, for the sake of data analysis, where the eye was during that fixation. The solution taken in PICDRY is to use the average position in that chunk, weighted by the duration at each specific x,y sample position. More precisely,

$$xpos_{fix} = \frac{\sum (xpos_{datapoint} \times duration_{datapoint})}{\sum duration_{datapoint}}$$

$$ypos_{fix} = \frac{\sum (ypos_{datapoint} \times duration_{datapoint})}{\sum duration_{datapoint}}$$

where $xpos_{fix}$ and $ypos_{fix}$ are the x and y positions of the fixation (in pixel values), $xpos_{datapoint}$ and $ypos_{datapoint}$ are the x and y positions of the current data point (in pixel values), and $duration_{datapoint}$ is the duration of the current data point.

To summarize: Samples are generated at >1000Hz by the eyetracker. Multiple samples are clumped into data points at runtime based on the *pixel criterion*. Data points less than *cutoff* are assumed to be saccades and are removed by EYEDROP. Remaining data points form temporally contiguous chunks, which are assumed to be fixations. PICDRY computes the duration of each individual fixation as the duration of each chunk, and the position of each fixation as the average position of each data point in that chunk weighted by the duration of each of those data points.

Following generation of fixation position and duration data, PICDRY drops any fixation that is below the *low_cutoff* or above the *high_cutoff* that is specified by the user. This allows the user to exclude from consideration short fixations that are unlikely to reflect cognitive processes, as well as long fixations that may reflect lapses of attention and/or track losses.

Each fixation is assigned to a scoring region (or not) based on (1) the position of that fixation as computed above, and (2) the boundary of the scoring region as specified in the region file that is used as input to PICDRY.

Finally, the position and duration of each individual fixation is used by PICDRY to compute average values, by subject (averaged over item), or item (averaged over subject), for each condition and region.