Data for Umeton et al "Pattern and speed interact to hide moving prey", Current Biology 2019

The following is a list of all the files in this archive and a description of how to use them.

CodedData3.xlsx (Excel file) and CodedData3.csv (same as comma-separated value text file)

Contains the results of the video coding. Each row represents a single trial, i.e. one presentation of the prey. The columns are as follows:

individual: identifies the mantis doing the experiment, e.g. S10Feb18

direction: +1/-1, specifies whether the prey was moving left or right

bug: specifies the prey type, as follows: 0 = black, 1 = grey; 2 = background-matching; 4 = wide-striped; 5 = narrow-striped.

speed: 37, 74, 145 = slow, medium, fast (average speed in deg/sec)

LiveCoder: specifies who coded the mantid's response live, during the experiment, via the webcam: DIANA = Diana Umeton, BLOCK1,2 = Eugenia Fezza

LiveResponse: 0 or 1: specifies whether (1) or not (0) the live coder thought that the mantis responded to the stimulus, either by making saccades, striking, or tracking the target.

LiveTrack: 0 or 1: specifies whether (1) or not (0) the live coder thought that the mantis tracked the stimulus. NB these two columns are almost always identical.

Hanna, Giorgia: These two columns record the judgements made by our two lay coders. 0 = confident that mantis did not respond; 0.2 = unsure but probably mantis did not respond; 0.8 = unsure but probably mantis did respond; 1 = confident that mantis did respond (in any way).

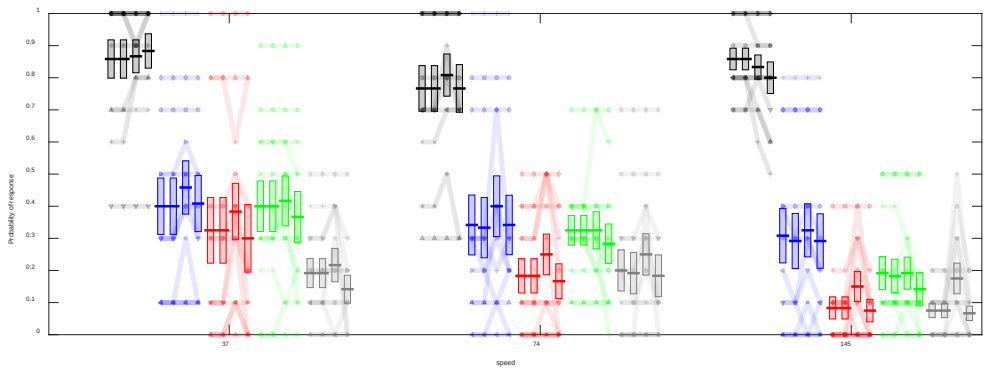
trialnumber: records the number of this presentation in the series of 150 presentations (1 to 150).

CompareCoders6.m

Matlab file which reads in CodedData3.xlsx and produces various plots comparing the judgments made by the different coders. The confidence judgments made by the offline coders are discarded, i.e. their judgments are converted to binary ones.

This file uses the file CohensKappaBinary.m and BinomConf_Score.m .

Example output:



This figure shows the judgments of the live coders and the two video coders. The different colours show the bug types (black = black, blue = wide-striped, red = narrow-striped, grey = grey, green = background matching. Within each colour block, from left to right are: live coders including any response; live coders counting tracks only (note this makes virtually no difference); Video coder 1 – Hannah; Video scorer 2 – Georgia. The horizontal lines show the grand mean across all mantids for that coder, speed and bug type, while the vertical bars above and below show +/- 1 SEM. The pale lines show data from individual mantids; data from the same mantid is linked. We see that there is fairly large variability between mantids (modelled as random factor in our statistical analysis) but rather little variability between the human judgments of their behaviour (hence the high Cohen's kappas between the live coding and those made later from videos).

mantisSpeedyBugBM.zip

This contains the data. When unzipped, it contains sub-folders containing data from a single experimental session. For example, "S10Feb18 23-03-2018 09.32 (DIANA)" contains data from an experiment that Diana ran on 23rd March 2018, starting at 9.32am, on mantis S10Feb18. Where a mantis did not complete 150 trials when first tested ("BLOCK1"), it was retested in a new session on a separate occasion ("BLOCK2", "BLOCK3").

Each folder contains four files written at the time of the experiment:

The parameters for each session are stored in params.mat (containing the variable paramSet) and the results scored by the original experimenter are stored in results.mat (containing resultsSet), while hardware_info.mat records data about the hardware. The toolbox…zip file contains information enabling the exact experiment to be re-run using GT's M3toolbox, available at https://github.com/m3project .

paramSet is an array with one row per trial (so, 150 in our experiments).

The first column records direction (1=left, -1=right).

The second column records bug type: (0=black, 1=grey, 2=background matching, 4=2W or wide stripes, 5=4W or narrow stripes).

The third column records speed (37, 74, 145).

resultSet is also an array with one row per trial.

The first column records the number of saccades.

The second column records whether or not any tracks were made.

The third column records whether or not the mantis made peering movements. We did not include these as a response to the stimulus, as mantids often made small peering/quivering movements throughout the entire trial and they did not seem to be a response to the stimulus (see https://www.jennyreadresearch.com/research/m3/faq/guide-to-scoring-mantis-videos/)

The fourth column records whether or not the mantis struck. This occurred only rarely.

Each folder also contains video files recorded at the time of the experiment, named trial1.mp4 etc. Note that these are not what were used by the experimenter to code the responses at the time; the experimenter saw a different webcam which did not view the screen, so they coded blind to stimulus.

Video files named Masked_trial1.mp4, etc, are the same videos, but after the addition of a black mask blanking out the part of the monitor where the moving target appeared. This enabled our video coders (Hannah and Georgia) to code responses blind to the stimuli.

The judgments made by the offline coders are stored in resultsGiorgia.mat and resultsHannah.mat. We did not ask our video coders to score those sessions that were incomplete and not used in the paper.

SelectDataWithCoders3.m

This is a Matlab file which reads in the data from the folder mantisSpeedyBugBM, screens out sessions where the mantis did not respond to the black control bug or did not complete all 150 trials, and saves the results to the Excel file CodedData described above.

Coding the videos

The instructions we provided to our video coders are available here:

https://www.jennyreadresearch.com/research/m3/faq/guide-to-scoring-mantis-videos/

These contain examples of mantis responses and non-responses.

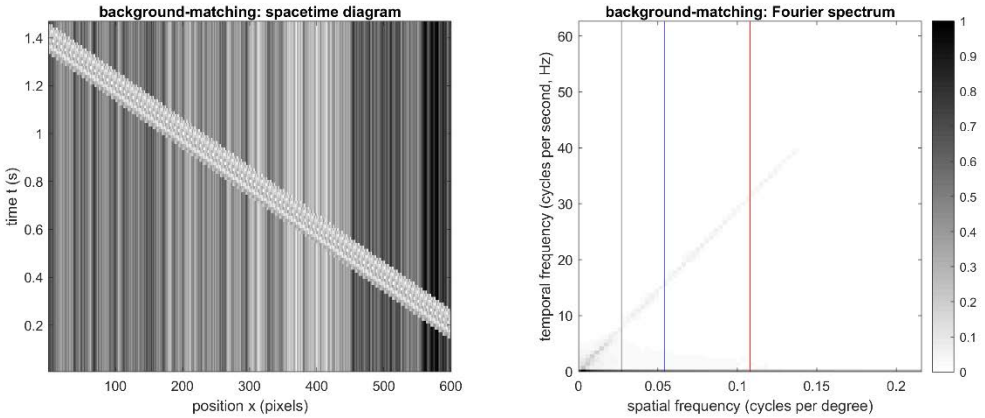
We wrote Matlab software to display and code the masked videos. These are provided in the files VideoCoder.fig and VideoCoder.m. Place these in the same folder; move to that folder in Matlab and type VideoCoder at the command line. This should then run as described in the above website. Data will be written to file results[Name].mat in each folder of mantisSpeedyBugBM. In this way, anyone who has the stamina to code all 1800 videos can check how well they agree with our coding. You can run SelectDataWithCoders3.m (qv) to generate a new Excel file containing your coding to compare with ours. To do this, you'll need to modify the list of coders to include your new coder, at line 14 of SelectDataWithCoders3.m.

Statistical analysis

The statistical analysis is provided in the R markdown document AnalyseData.Rmd and the resulting document AnalyseData.pdf. To run this, you will need the R libraries readxl, ggplot2, lme4, dplyr and tidyr.

Fourier spectra of stimuli

The file FourierSpectrumVideo2.m generates Fourier spectra of the stimuli. It calls genNaturalPattern.m (to generate the background texture) and jfft2.m (to get the frequencies of the Fourier spectrum). It generates files like background-matching.tif:



The left panel shows a horizontal slice through the stimulus and how that changes with time. The prey is visible as a diagonal stripe. The right panel shows the Fourier spectrum. The vertical lines in the right hand plot show the frequencies corresponding to the horizontal size of the prey (prey width = half a period, black line) and to the wide and narrow stripes (blue and red respectively).

The Fourier spectrum is computed by generating a stimulus as a function of (x,y,t), taking the 3D Fourier spectrum of this, then averaging over the horizontal frequencies (corresponding to the variation as a function of y) to get a 2D Fourier spectrum as a function of vertical spatial frequency and temporal frequency. This is then averaged over multiple videos to give the results shown.

Note that for this analysis, “0” in the stimulus levels corresponds to gray, with positive numbers representing bright features and negative dark, i.e. we work in contrast rather than luminance. This only affects the value of the DC.