Subject: Re: Fwd: Roadmap to Deviant programs
From: scott <oldmanwarren@gmail.com>
Date: 4/26/19, 12:02 PM
To: Ru-Yuan Zhang <zhan1217@umn.edu>

Im glad you found ImageHelper. Sorry I am not always more specific. In general you can go to my MATLAB folder and in BASH use commands such as

```
find ~sgwarren/matlab -name "ImageHelper.m"
```

to find a program I might carelessly reference. A lot of useful programs I downloaded, for example, are in sgwarren/Toolbox.

ImageHelper has lots of helper functions I use extensively.

Regarding deviantPosition versus stimulusIndex, you're right, my apologies. See below for a write up of all the data fields. First things first however.

The mapping from number to position in space is a quirk of the exceptionally well-documented program...

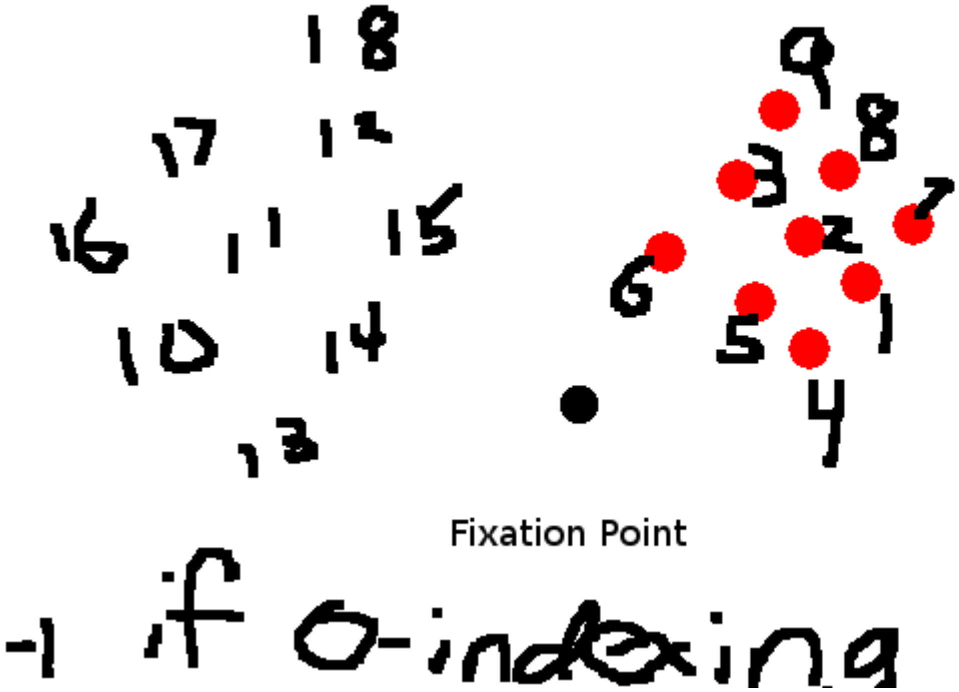
You're right, my apologies, deviantPosition is encoding the value. What the numbers 0-17 mean is probably the dumbest thing I've ever coded. Hang on, it will make sense I promise.

First: always remember that all the Deviant file data zero-indexes because it was written in C#. But all the MATLAB code 1-indexes.

So the stimulus is an array of 3x3 on either side of the screen, nine in total per side -- and with both sides it makes 18 in total. But which one is #0, #1, etc?

The program can't just say #0 is this stimulus element here, #1 is that one there, etc., because it has to compute the positions of the stimuli based on program parameters (the location of the stimuli, the size of the stimuli, etc.). As it computes the positions it assigns them values 0, 1, 2, 3, etc...

Whatever side the stimulus was programmed to be on (always right side in my programs, i.e. stimulus X position is positive), 0-8 are the nine on the programmed side, and 9-17 are mirror images on the opposite side. So it's just a matter of knowing what order the Deviant program computed stimulus positions. Long story short it computed along the MIDDLE row COUNTERCLOCKWISE first, then the INNER row, then the OUTER row. So you end up with the following numbers representing each stimulus in the two 3x3 arrays, assuming the program was given a stimulus position on the RIGHT FIELD:



(Sorry I 1-indexed in the picture, but you get the ordering.)

So for example, let's look at Blofeld's position tuning. In the F or File structure, the parameters for the stimulus position are in F.gaborSpace. We see the Azimuth was 3.25 and the Elevation was -2.50, so the stimulus was in the Right Lower quadrant. Stimulus 1 will be the middle and centered on (3.25,-2.5) degrees. Stimulus 0 will be more MEDIAL / CLOCKWISE, and stimulus 2 will be more LATERAL/COUNTERCLOCKWISE. Stimulis 3-5 will be the inner row, and 6-8 will be the outer row.

You probably noticed that there are many many more presentations of 0-8 than 9-17, and that is because I didn't want to map the other visual field because there's no imaging data from that field and its a waste of time. Those are all basically the same control state of "stimulus in other visual field". That's why

numel(find(deviantPosition == 1))

is about the same as

numel(find(deviantPosition > 8))

So for Blofeld the array looks like...

5

```
   4  2
3  1  8
0  7
6
```

TLDR: 1 is the middle, 0 and 2 flank 1, 3-5 are the inner row closer to fixation, and 6-8 are the outer row. **3 and 8 are the biased stimuli that occur most frequently.**

F = File Structure = Global Parameters that apply to Every Trial

F.dataGroupParam = Information about the data sources LabLib used to collect data from

F.dataGroupParam.singleData = Substructure same as above
F.dataGroupPatam.singleData.dataName = names of data inputs (imageData, eyeData, etc)
F.dataGroupPatam.singleData.deviceName = names of hardware data came from. PVCAM is the autofluroesence camera, iViewX is the eye tracker sending both eye position and pupil size, bTop is the switch that sent reward signals.
F.dataGroupPatam.singleData.channel = The 2's here just show that iViewX sends data packets in the formal (eyePositionX, eyePositionY, pupilSize1, pupilSize2)
F.dataGroupPatam.singleData.timing = milliseconds per sample. Camera ran at 5 Hz = 200 ms per image, eye tracker ran at 240 Hz = 4.1667 ms per sample
F.dataGroupPatam.singleData.type = I'm not sure what type of 4 signifies for reward data

F.dataGroupParam.count = I'm not sure

F.stimVarParam = IF a stimulus was being varied between trials, it was this. This says 'Position' for almost all data because Position is varied during Position Tuning. The param still says Position in task data, but task data showed all nine stimuli and this field is irrelevant.

F.testMode = 'Tuning' for Position and RadiusContrast trials, and 'Discrete' for Deviant task trials

F.gaborSpace = Spatial params for the Gabors in their BASE condition. Mostly sefl explanatory
F.gaborSpace.Orientationdeg = OFFSET added to stimuli IN ADDITION to fact that stim array was oriented along the line connecting fixation to the stim array (so a stimulus array located at (5,5) would be oriented to 45 degrees + offset in this field. Usually 0, and stimuli were aligned to the orientation bias of V1 to maximize stimulation.
F.gaborSpace.Contrast = base contrast of stimuli
F.gaborSpace.SFcycledeg = spacial frequency (cycles per degree)
F.gaborSpace.Sigmadeg = Sigma parameter for size of gabor
F.gaborSpace.Radiusdeg = hard stop size of gabor, also determines gabor spacing in the 3x3 array

---- So the gabor tapers in contrast according to sigma, and hard stops according to radius
F.gaborSpace.InnerRadiusdeg = Always 0. If greater than 0, the gabors would be donuts. Used in other programs to show donuts/rings.
F.gaborSpace.ArcStartdeg = Always 0 to show full circle
F.gaborSpace.ArcWidthdeg = Always 360 to show full circle. Changing these would let you show wedges (good for retinotopic mapping)
F.gaborSpace.Azimuthdeg = X position in degrees of visual space of the middle of the middle element. The array is centered on (X,Y)
F.gaborSpace.Elevationdeg = Y position as above
F.gaborSpace.SPhasedeg = Always 0, would let you vary spacial frequency phase if you wanted (to change location of white/black strips within gabor, useful if you have low spatial frequency Gabors that look weird but irrelevant for this program)
F.gaborSpace.Widthdeg = I think always 0? Does something useful for other programs but not used here.

F.gaborPulseSpace = Parameters for the Gabors that they Gabors CHANGE to when the animal detects the flash. "Pulse" is the term for the flash.
You will see that gaborPulseSpace has lots of differences between it and gaborSpace, but Deviant was hardcoded to ignore them. Only gaborPulseSpace.Contrast matters. This is the contrast the Deviant position increased to when the animal was supposed to detect it. For example Blofeld detected changes from 20% to 60% contrast during position tuning.

(Note, in genera, animals needed a higher step to detect changes during POSITION tuning when stimuli were presented independently, than during DISCRETE AKA DeviantOn or DeviantOff task modes when stimuli were in an array. Also Kananga detected smaller changes than Blofeld because Kananga is/was just a #1 boss monkey)

F.gaborColor = Color parameters for the Gabors. Always KDLThetadeg = 90 and KDLPhideg = 0, which means black and white.
F.gaborTime = Temporal parameters for the Gabors. Always TFHz = 4 hertz and TPhasedeg = 0

Like PulseSpace, PulseColor and PulseTime define color and temporal characteristics of the Pulse stimulus. These are not used in Devient

F.gaborSMod = 'Sine' for sine wave Gabor patterns. Lablib lets you use Square, etc. but this faculty not used here.
F.gaborTMod = 'Counter Ph' = Counterphasing gabors, 'Drift' = drifting Gabors. Some of Blofeld's later used Drift because he had fixation problems.

F.scaling = 'None' (Scaling let the program use larger Gabors for more peripheral stimuli in order to try to make each Fabor have an identical spatial footprint // representation within V1, but the stimuli ended up being so parafoveal that this didn't seem to matter in terms of data quality but it DID piss the monkeys off, so we didn't end up using scaling)

F.baseTiming = Timing parameters for the trials
F.baseTiming.Intertrialms = Gap between trials as a rest period between trials. Usually small because monkeys had a long time to acquire fixation and self-paced their breaks.

F.baseTiming.Acquirems = Time from when fix dot is presented to time they have the look at it
F.baseTiming.PreStimulus = Time from fixation to when stim array is presented
F.baseTiming.CueDuration = Not used I think.
F.baseTiming.Meanms = It says mean but this ends up being the MINIMUM time from Stimulus ONSET to the Pulse the monkey detects (see Jitter timing below)
F.baseTiming.Pulseframes = Number of frames the pulse lasts for. How many milliseconds this is must be computed from monitor frame rate
F.baseTiming.Interpulsemsfrm = Holdover from an older version of program. Not used.
F.baseTiming.MinResponsems = Min and Max of time window for monkey to response. A minimum is used because a response time of 0 is just a successful guess.
F.baseTiming.MaxResponsems
F.baseTiming.catch = Percent of catch trials. Also determines trial maximum duration (see below)

F.jitterTiming = Timing parameters for the variation / randomization of times between trials. Very mucF.ho important: POSITIVE values and NEGATIVE values mean very different things.

POSITIVE values = UNIFORM distribution between F.baseTiming.XXX and F.baseTiming.XXX+F.jitterTiming.XXX. So for example, if baseTiming has a value of 600 and jitterTiming has a corresponding value of 400, the monkey would see a uniform distribution of timings for this segment from 600 to 1000.

NEGATIVE values = EXPONENTIAL distribution of times. Geoff will explain this better than I can type it.

Using this, the corresponding values of jitterTiming are added randomly to the base valyes in baseTiming to randomize the intertrial, pulse timing, etc. durations of every trial. Of course, exact timing per trial is available in the Trial structure. These are just the timing parameters used to randomize these timings.

F.stimVar = Parameters that determine how the Position tuning randomizes Positions (and how RadiusContrast randomizes Radii and Contrasts). For position tuning...
F.stimVar.Levels = Number of different stimulus variations. Is "10" for Position tuning to represent the 9 stimuli IN the imaged visual field, and a 10th position that says to pick a random position outside the visual field. Not used for DeviantOn/Off data.
F.stimVar.Center = Not used
F.stimVar.Range = Not used

F.stimFlags = Noting really pertinent here. Image Stabilization was not used because it was insanely distracting to the animals. Autoinitiate means the program will keep running to bug Blofeld with wrong beeps because he had a tendency to sit there and zone out.

F.behaviorFlags = All self explanatory. Animals had to fixate (sometimes turned off for niche reasons), we didn't use a Lever, Sounds were enabled, if the animal was Wrong they had to wait all the way to the end of a full catch trial (a several second wait) to discourage Guessing. ProgressiveReward, if on, means win-streaks got larger and larger rewards. Kananga got this for some trials, but he would end up with such large rewards bTop would flood the room and make a mess. What a good monkey.

F.fixationParams.FixSpotdeg = size of fixation spot
F.fixationParams.FixWindowdeg = FULL WIDTH of fixation window. Animals had to be within FixWindowdeg/2 degrees of fixation in X and Y (not radial, X and Y independently checked. They can cheat out to sqrt(2(FixWindowdeg/2)^2) if they cheat along a 45 degree angle.

F.responseWindows.RespWinWidth = full width of the window Monkeys had to look INTO to signify they were looking at the stimulus. Generally I made this generous, because the eye tracker was so tightly calibrated to fixation positions to check the close fixation they had to maintain.

F.behaviorTrials = Parameters that describe block timing and stimulus geometry. Most are not used or associated with behaviorFlags not used (e.g. the progressive reward).
F.behaviorTrials.Tries = Not used
F.behaviorTrials.BlockSize = Number of trials before animal alternates attention to Left vs Right
F.behaviorTrials.BlockRep = We only show each block once then randomize a new block.
F.behaviorTrials.ArraySize = 3. The program can do 5x5, 8x8, you name it. But 3 = the 3x3 array we know and love.
F.behaviorTrials.ScaleFactor = The probability gradient. 800 = Stimuli 4 and 9 were 8x more likely. If 0 there was no probability bias (Kananga's DeviantOff data). if Negative the gradient was FLIPPED and biased toward positions 6 and 7 (Blofeld's DeviantOff data).
F.behaviorTrials.ExpCueInvalid = I'm not sure, had to do with moy many times they had to respond to very easy cues to show they understood left ve right bias
F.behaviorTrials.ExplicitCueReps = Number of cue trials
F.behaviorTrials.ProgressiveExpo: Not used (progressive ward parameter)
F.behaviorTrials.WaitReactTime = I think time they have to wait before reacting, but see min/max response windows timing above

F.fixSpot = Fairly self-explanatory params for the fixation spot
F.fixShape = Always a 'Circle'

F.stimWindowFlags = Noting pertinent, these are flags that determine behavior of the Deviant program used to monitor real time performance

F.eyeCalParams = Settings of the eye calibration. Somewhat importantly...
F.eyeCalParams.FixationAtimuthdeg and FixationElevationdeg: OFFSET to ENTIRE SCREEN such that (0,0) is actually at this position. I used non-zero values to move the stimulus over a bit because the eye tracker camera blocked a corner of the screen so monkeys couldn't see it.
F.eyeCalParams.CalibratingOffsetdeg = During eye tracker calibration, the spacing of calibration points. Geoff can explain.
F.eyeCalParams.NumberofOffsets = Always 6 because 6 is best.
F.eyeCalParams.CorrectionFactor = How aggressively program updates eye tracker calibration during program run time.

F.eyeCalParams.displayCIEx and y = Color calibration information, honestly forgot more than I remember. Long story short screens need calibration, Geoff will explain.
F.eyeCalParams.displayDistances = Size of monitor an distance from monkeys eyes to monitor center.

F.Synth*** = Parameters to make fake data for Lablib testing purposes. Not used in datasets for hopefully obvious reasons.

F.PVCAM = Camera settings
F.PVCAM.Exposurems = Exposure time
F.PVCAM.Intervalms = Spacing between images. Always longer than exposure time because this must ALSO include time for camera to send image data to computer.
F.PVCAM.GainMultiplier = Gain setting used. Generally the same gain day to day.
F.PVCAM.SpeedTable = Camera specific speed settings
F.PVCAM.GainIndex = Additive or Multiplicative gain, see camera manual
F.PVCAM.ImageXBin: 2 because we used 2x2 binning to downsample images from 512x512 to 256x256
F.PVCAM.ImageYBin: 2 as above
F.PVCAM.ClearFlag: Clears image data between images
F.PVCAM.PixelTimens: nanoseconds of read time (I think) per pixel?

F.bTopBits and F.iViewXParams and F.channelAssignments: Nothing pertinent. Eye data was not downsampled.

T = TRIAL STRUCTURE = TRIAL SPECIFIC INFORMATION

Everything in F either (1) applied to every corresponding Trial, or (2) varied in a non-significant way between Trials and was not used to group trials (e.g. small changes in trial timing or rewards to encourage continued good behavior). Trial structure contains trial specific info.

First thing to note is that while most Trial fields come from Lablib, many come from my preprocessing too to determine what preprocessing has been done to each trial.

Trial Data from LabLib

The following timing data, with rare exception, are present for all trials...

T.trialStartTime = Time of trial start, the very beginning of the trial
T.prestimulusTime = Time the animal fixated on the fixation spot and a prestimulus period began PRIOR to stimulus array or position tuning stimulus presentation
T.stimulusOnTime = Time the stimulus array or position tuning element appeared
T.stimulusOffTime = Time stimulus turned off **for any reason**
T.trialEndTime = Time trial ended.

while the following may or may not be present.

T.deviantOnTime = The start of the Pulse which the animal should detect
T.deviantOffTime = The time the Pulse stopped and the stimulus returned to normal (not the time the animal had to respond)
T.saccadeTime = The time the monkey made a saccade **anywhere and for any reason**

One needs to compute from these what happened. For example, if there is no data in deviantOn and Off time, and there IS data in saccade time, then the animal made a saccade but the stimulus didn't flash and the trial is a false positive (or a fixation break). Conversely if there is an On and Off time but no saccade time, the animal did not make a detectable saccade and the trial is a false negative.

T.trialDescription = Tells you what actual parameters the trial was randomized to have
T.trialDescription.stimulusIndex = This is he number that the trial was within the BLOCK. So this will vary from 0 to (F.behaviorTrials.BlockSize-1). Basically useless. You will note this isn't a perfectly uniform histogram because errors, program crashes, monkey misbehaving, and so on meant entire blocks were not always presented. This didn't significantly change the probabilities of different Deviant positions overall. **IMPORTANT: -1 means it was a Cue trial.**
T.trialDescription.explicitCue = Signifies where the animal was Cued to on this trial. 0 = in the same visual field as the gaborSpace was programmed to be in, and 1 = opposite field. This means 0 is Attend In and 1 is Attend Out of the imaged V1.
T.trialDescription.tuningValue = For Position tuning, tells you which stimulis was presented and always equals DeviantPosition unless tuningValue is 10, in which case the program presented a random stimulus on the other side (Deviant will be randomly from 9-17). Not used in DeviantOn/Off data.
T.trialDescription.responseAzimuth and responseElevation = Location of the correct array. Generally you can know this already from the deviantPosition.
T.trialDescription.deviantPosition = The deviant position that pulsed in the given trial, numbered as above. 3 and 8 are the biased spots in DeviantOn data.
T.trialDescription.devantTime = The time in milliseconds from the start of the stimulus period that the Pulse was SCHEDULED TO start, but there is no gaurantee that it was actually shown (monkey may break fixation, guess wrong too early, etc.)
T.trialDescription.arrayOrientation = The actual orientation of the array in addition to any offset and orthogonal bias programmed in as discussed above in the File gaborSpace data. Trial to Trial the Gabor changed between four cardinal orientations to avoid saturating / suppressing V1 neurons of certain orientation preferences.

T.behavior = How the trial was coded by LABLIB, not necessarily how it should be coded for data processing (see trial_performance.m)
0 = Correct, 1 = Wrong (false positive), 2 = Failed (false negative), 3 = Broke Fixation, 4 = Ignored (animal never fixated), 5 = Quit (I terminated trial)
T.radiusbasepulse = Used in a niche tuning condition.
T.eyeDataCalibration = The current eye calibration for this trial. Can be used to transform raw eye data into screen position data. See ImageHelper.convertEyeData.
T.saccadeEndpoint = Struct with x and y position of where the saccade ended up in DEGREES (not raw Eye Data). Useful for looking at animal responses without converting he raw eye data
T.trialCertify = Doesn't mean much.
T.sampleZeroTime = Self explanatory, the time that samples start coming in. Usually

similar to trialStartTime. Using this and the sample frequencies you could time each data point, but since the program sometimes drops images its better to use individual imageTimes
T.eyeData = Raw eye data from iViewX. Can be converted to degrees-on-screen with T.eyeDataCalibration and ImageHelper as above
T.pupilData = Raw pupil data from iViewX. Only available in some subsets of trials where this channel was saved.
T.imageData = OFFSET into corresponding I matrix in the data file where the trials first image lives
T.imageTime = Times of individual images. Useful to validate which trial epoch each image corresponds to.
T.spikeData = We did not record any spike data in this experiment, always empty
T.reward = Amount of reward given in arbitrary units (proportional to amount of juice but can't compute exact volume)
T.rewardTime = Parameters used to compute reward
T.trialNumber = Increments with each trial in a block. Lots of discontinuities in the counting represent trials thrown out and excluded from further analysis, quits and breaks in data collection, resets, etc. Not generally useful,

Trial Data from My Preprocessing

Note, I could and need to send a whole email about preprocessing. A LOT goes into making the Non-Raw / Final data sets for further analysis. The basic steps involve manipulating all the trials so they have the same structure format in MATLAB (cattrial), and registering all the images so they all align as well as possible to a single reference image per Monkey (so that pixel X,Y represents, as close as possible, the same brain tissue in all images)

T.fullDataPath = Where the trial came from once upon a time
T.cleaned_by_cattrial = 1 if the trial has been run through a function called cattrial used to combine trials from different original save files (different days, etc.). cattrial basically gets rid of structure discrepancies between different files' trials (e.g. some did and some didn't have spikeData fields, even through spikeData is always irrelevent). Should always be 1, this is just a flag to let you know you're not handing the purest raw data
T.invalidate = A flag for, if for some reason in preprocessing, I decided this trial was junk.
T.visualInspection = A flag for whether I personally visually reviewed the alignment of each image in the dataset. This is probably still 0 even though I did personally review them all, because I did the review as part of a giant extended fever dream while making a thesis.
T.intertrialRegistration = For each trial, a single Reference image from the trial is aligned to the Master Reference image for the given animal. This is an affine transformation matrix used to align these two images. An Affine transform is used because day to day there were variations in the distance and angle of the camera which made simple translation-based registration impossible.
T.intratrialRegistration = All of the images WITHIN a given trial were registered to each other via simple translation. This is an array of the X and Y offsets applied to each image.
T.referenceImage = The reference image used for this trial for intertrialRegistration as above