Gaze Contingent Eye Tracking and Timing

Precision in Timing

- Do we really need precise timing?

- NO
  - Variance in our measured variables is added equally across all conditions
- Yes
  - Mental chronometry is all about measuring the precise timing of cognitive processes

Random noise

- Both points are true
- But what is the real cost of random noise in the experiment?
- Power is 1- β
  - β = Chance of incorrectly rejecting the null (making a false negative)

Mean condition 1	250 ms
Mean condition 2	260 ms
Desired power	.8
Expected sd	10 ms
Required n	16

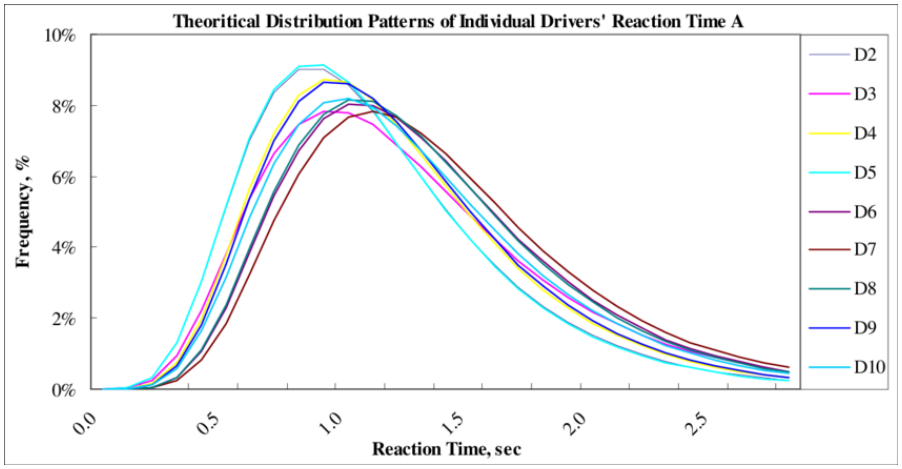
Mean condition 1	250 ms
Mean condition 2	260 ms
Desired power	.8
Expected sd	**15 ms**
Required n	36

Mean condition 1	250 ms
Mean condition 2	260 ms
Desired power	.8
Expected sd	**20 ms**
Required n	64

G-power or
http://powerandsamplesize.com

Sources of noise

- Normal variance
  - This is data, not noise
- Between subject differences
- Within subject differences
  - Eg. Alpha?
- Block/session differences
- Equipment
  - Gamepad/mouse/kb
  - Monitor refresh?
  - Eye tracker latency?
  - Operating system multitasking



Statistical consideration:
LME random effects

Minimize with good code and fast equipment

Minimize noise


Gamepad/mouse/kb


Monitor refresh?


Eye tracker latency?


Operating system multitasking

Minimize noise

Gamepad/mouse/kb

Monitor refresh?

Eye tracker latency?

Operating system multitasking

Monitor choices

- ‘Gaming’ monitor
- Dual-DVI, HDMI or display port

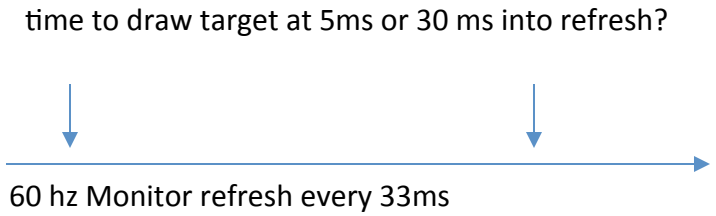
- 144+ hz
  - Frames per second

- 1ms response time
  - pixel black to white

- 2-4ms latency lag
  - Signal to image

- IPS?? (for colour maybe) but in general, these monitors will not be acceptable for proper colour space experiments. (CRT or graphic professional)

- Mostly you can only control how you send the images



Minimize noise

Gamepad/mouse/kb

Monitor refresh?

Eye tracker latency?

Operating system multitasking

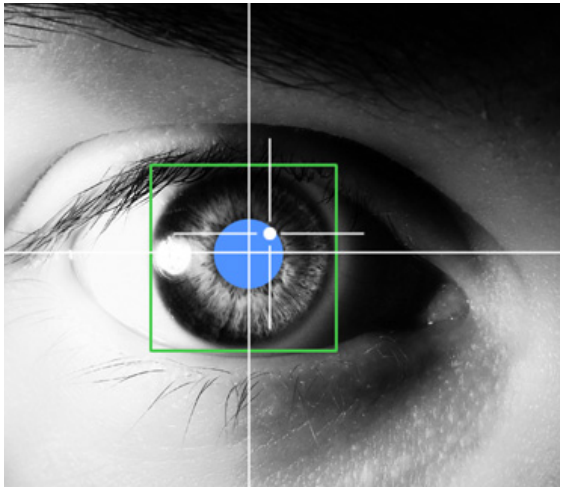
Specifications

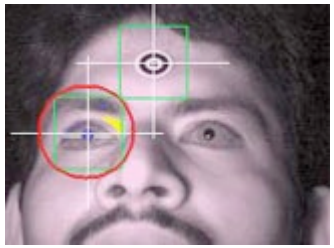
	SMI tower	Eyelink 1000
Temporal resolution	1250 hz	1000 hz
Spatial resolution*	< .1 degree	< .1 degree
Set up	1 computer (two possible)	2 computers
Expt software	SMI expt creator, Matlab, Presentation, Python, C++, E-Prime, …. Open Sesame	Eyelink Expt builder, Matlab, Presentation, Python, C++, E-Prime, …. Open Sesame
Offline data	Gaze, saccades, blinks, messages, AOI	Gaze, saccades, blinks, messages, AOI
Online data	Gaze, AOI	Gaze, saccades, blinks, messages, AOI
Online latency	1ms ?	2-4 ms for samples, longer for events (4ms plus 10 samples for smoothing?)

*with a mechanical eye. Real accuracy depends on calibration

Gaze location: Technical challenges

- Input image -> output gaze XY
- Still not a trivial task
- Pupil occlusion by upper lid
- Threshold problems
  - Are sclera and pupil greyscale values universal? Across individuals? Eye colour? Race? Medical conditions? Makeup?
- One to one mapping of gaze location and pupil image?
  - Only if you control for head position
  - Head tracking, chin rest
- Lighting
  - Dark pupil algorithms, but these require additional infrared source
- Accuracy limitations of camera?
  - Spatial accuracy improved by corneal reflection (.1 deg with artificial eye?)
  - Temporal accuracy improved with camera/cable technology (1000 hz +)
- Computational limitations?
  - Image processing
  - Triangulation
  - Saccade detection (Eye link only)
  - Various solutions

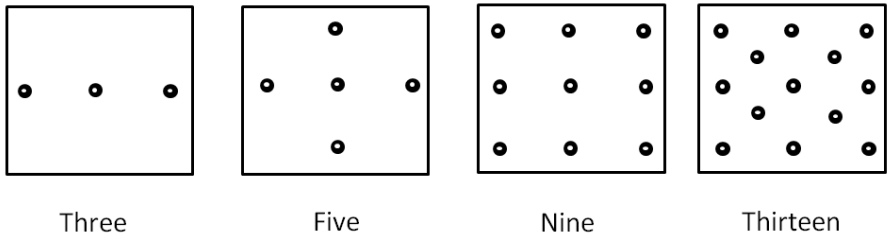


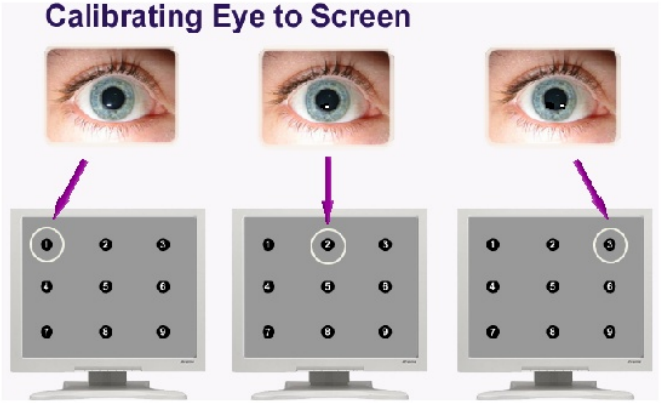


Still only compensates for small (20cm) head movements

Calibration

- Individual differences
  - Pupil size and shape
  - Pupil and iris shade (grey scale thrshold)
- Setup differences
  - These should be minimized!
  - Luminance (pupil size)
    - Control ambient lighting across participants
  - Camera/headrest/monitor placement
    - Use tape on desk to remember location
- **Validation** follows calibration and compares first vs second run.
  - Are the results consistent? Error in visual degrees
- **Drift correction** is a third single point comparison at fixation for the start of each trial
  - Only Eyelink forces drift correction

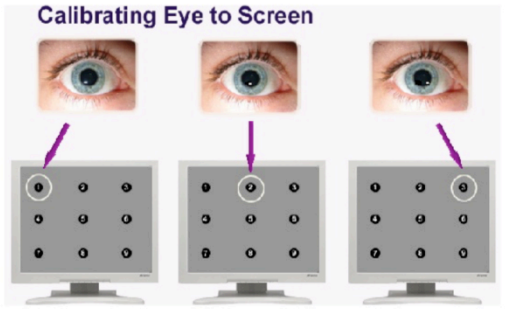


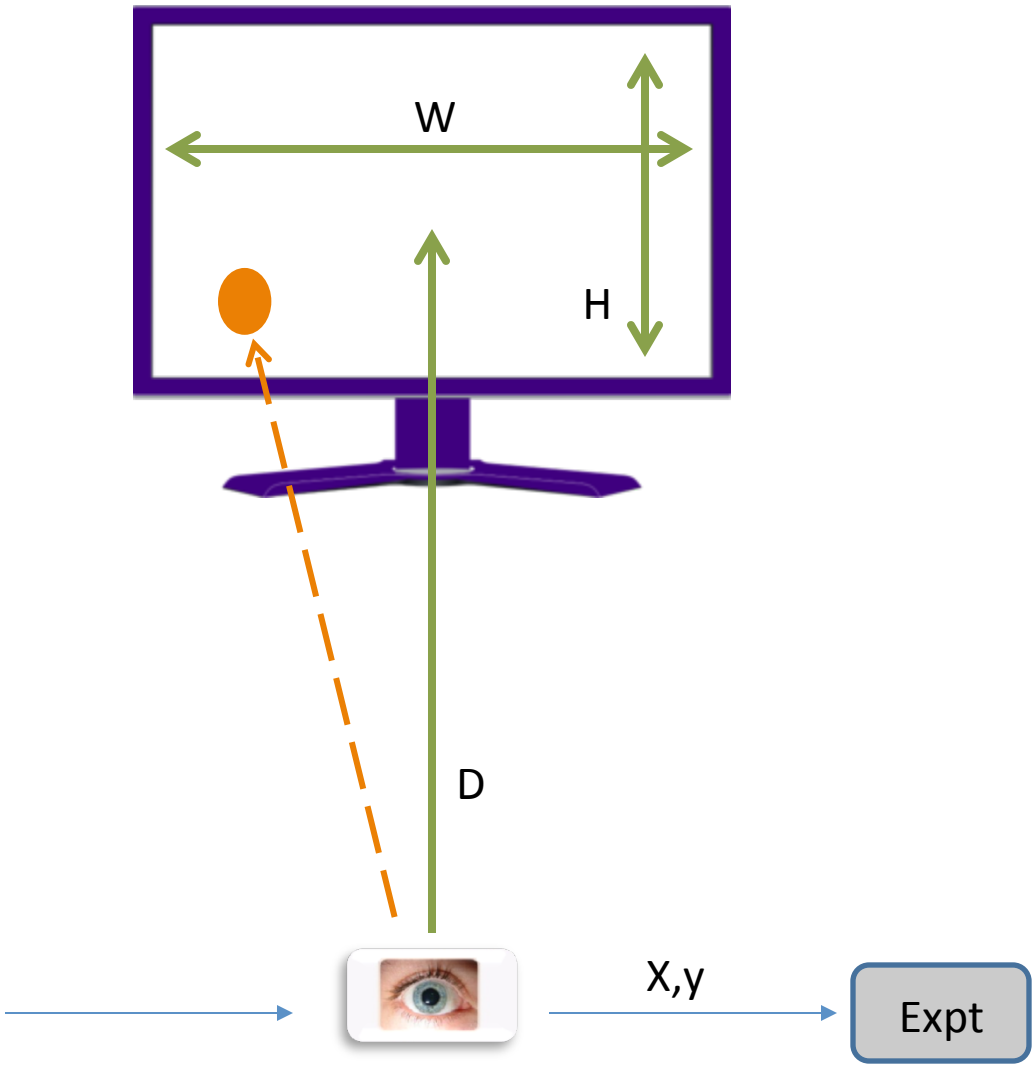


The pupil changes are very subtle, but still detectable via computer. Pupil vs corneal reflexion is much easier to spot

calculation

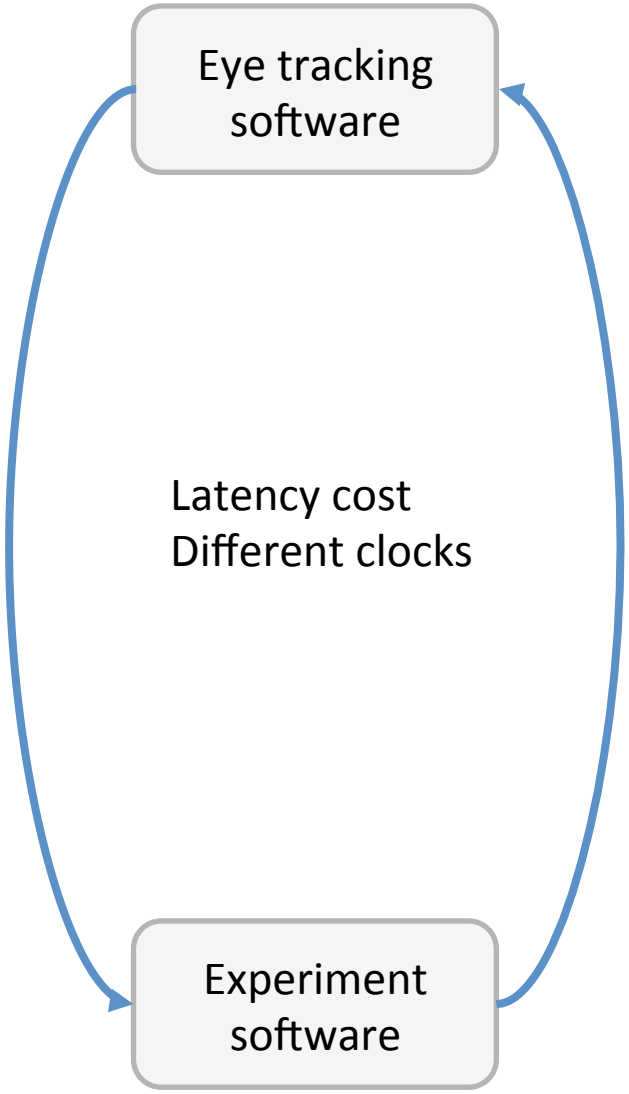
- Given
  - Current eye image
  - Monitor distance, height and width
  - Previous calibration image/points
- Calculate
  - Current X,Y
- In less than 1ms, every millisecond





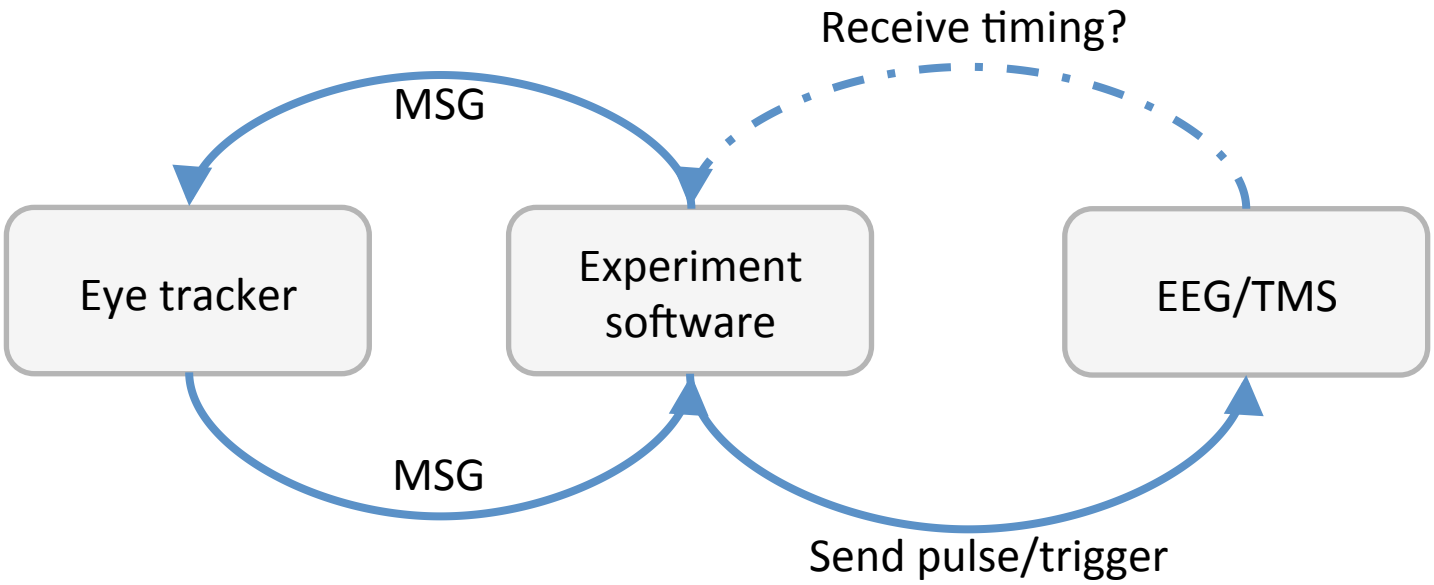
Solution? Dual computers/processes

- Eye tracking software is proprietary, so it is run as a separate program (SMI) or computer (SMI,Eyelink)
  - Dedicated computer has more resources free for online saccade detection
  - Single computer has lower latency between processes
- In both cases, this means that your experiment has to be run separately
  - What your subject sees, hears and interacts with uses its own program (ie matlab)
  - **Your program also uses a different millisecond clock!!!**
- **Communication is bi directional!!!**
  - Request current eye information to use in your experiment
  - Send messages about expt events to insert in the eye tracking time stream
  - **These two methods are the ONLY way you can compare times of eye movements and expt events**



Nuroimaging

- This can get more complicated when you want to integrate eye tracking with neuroimaging
- EEG/TMS cannot connect directly to eye tracker
- Experiment software must be the link between eye tracker and EEG/TMS
- Matlab, presentaion and E-Prime can send signals through triggerbox
  - Others might be possible



Minimize noise


Gamepad/mouse/kb


Monitor refresh?


Eye tracker latency?


Operating system multitasking

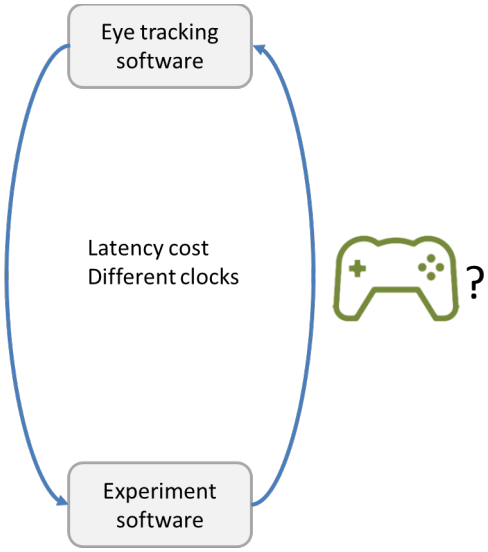
USB (and ps2???)

- ‘old’ computer geeks will tell you that only PS2 gives realistic timing
  - Ps2 is interrupt driven
  - USB is polling
- USB polling rate for a keyboard is ‘typically’ 125 hz
- You can customize the ‘polling lag’ of any USB device in the registry
  - 1000 hz is possible
  - Corsiar rapidfire K70 allows you to change this via hard switch
- Online tests suggest
  - Cheap PS2 beats cheap Usb by a few milliseconds
  - High end USB beats both by a wide margin
  - The corsair will likely beat most psych button boxes

Corsair rapidfire k70

Responses where?

- Old debate
  - Gamepad to display PC (Joe)
  - Gamepad to eyetracker PC (Eyelink)
- Remember, it's *difference* in times we care about
  - If you are comparing the button press to a saccade then use an eyelink response pad
  - If you are comparing the button press to a visual onset, then use a PC response pad
  - The second is simply more common



Minimize noise

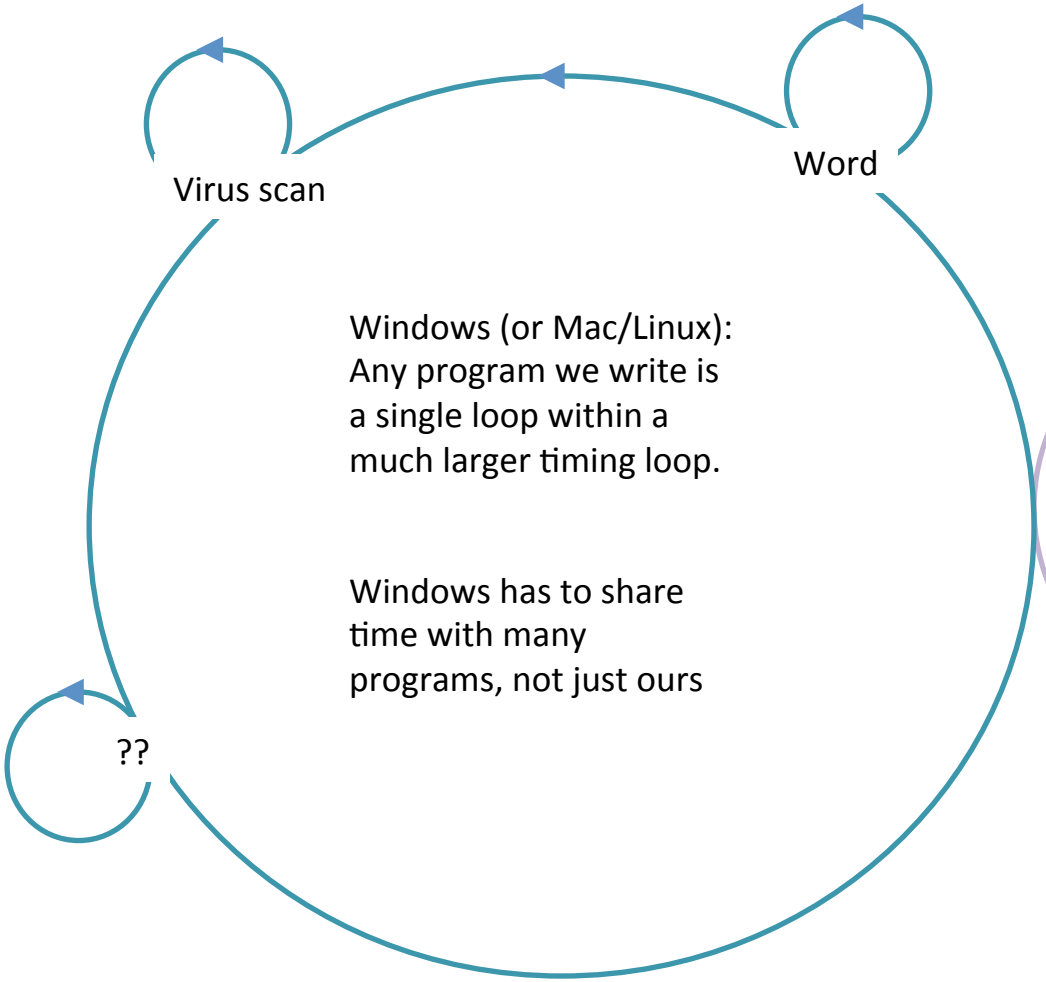
Gamepad/mouse/kb

Monitor refresh?

Eye tracker latency?

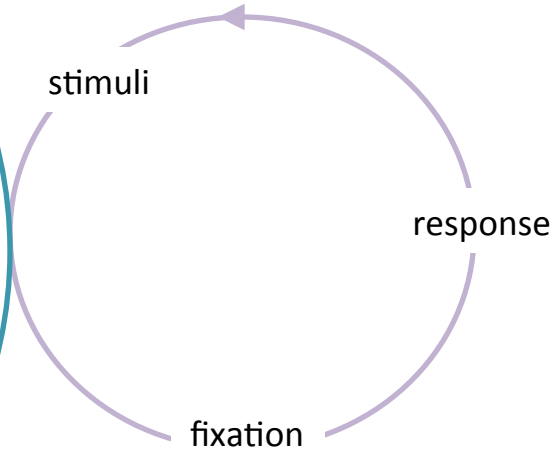
Operating system multitasking

Messaging loop



Windows (or Mac/Linux):
Any program we write is a single loop within a much larger timing loop.

Windows has to share time with many programs, not just ours



We would like to think of our experiments, as a nicely timed loop.

Packages like Matlab and experiment toolbox treat it this way

Timing in a modern OS

- You need to work with your OS to get maximum timing
- Millisecond timing is still possible
  - 'Millisecond timing on PCs and Macs'; MacInnes and Taylor, 2000

1. Don't install timing hogs on your experiment computer.
   - Anything that runs as a background task
   - Anything that 'runs on start'
2. Turn off automatic tasks
   - Disable windows update
   - Windows 10 is NOT appropriate for experiments
3. Disconnect from internet completely
   - Firewalls and virus scanners are the worst offenders
4. No remote access or group/network install
   - IT may try to tell you differently, but we have permission for all experiment machines to be separate from the HSE network.
   - Every two years they come in and try to change this
5. Update software versions and install new programs *between* experiments
   - Make sure no one else is running
   - 'shared' software libraries (Java, python) may be changed that influence the way an experiment runs

Timing in a modern experiment

- C/C++ vs Java/Matlab/Python
  - Java, Matlab and Python don't run directly on the computer
  - They use an interpreter or virtual machine
  - 'Garbage collection' might steal timing
  - For the most part, they seem good enough

- Quality programming is more important than choice of language
  - How do we interact with the operating system?
  - How do we separate slow access parts from timing critical parts?
  - How do we ensure precise timing for drawing, responses and eye tracking?

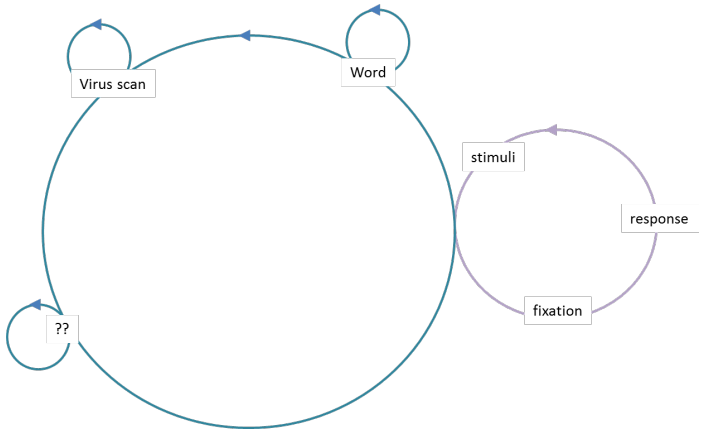
Breaking the loop

1. Use the messaging loop as if you were a normal program
   - Not ideal, it relies on pure speed of computer and a lot of assumptions
   - Eyelink sample code uses this method, so I assume so does their experiment builder

2. Pretend the windows messaging loop doesn't exist
   - Write your code as if it has 100% of the computer
   - As long as other timing hogs aren't running, this isn't a bad strategy
   - Used by psychtoolbox, and most others

3. Use a peek message loop
   - Optimal, but uncommon
   - 'bending' the loop, but not breaking it
   - Used for optimal timing in some computer games
   - This is beyond the advanced eye tracking course

- Unfortunately, many experiment packages are proprietary and we don't know which they choose.




```
while (msg != WM_QUIT) // {

if(PeekMessage(&msg,0,0,0,PM_REMOVE)
{ // You choose how to deal with critical windows events here }

else
{ // Run experiment here.}

 }
```

Those we know and suspect

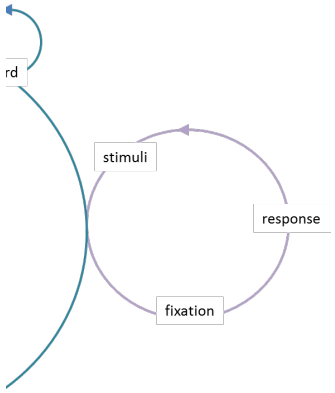
- Matlab/Psychtoolbox ignores the messaging loop
- Eyelink sample code uses the windows messaging loop
  - I would not use this code for timing critical expts
  - I suspect that experiment builder does the same
- OpenSesame??
  - Don't know but it is built with gaze contingent in mind

Other tricks

- Double buffer
  - Graphics technique to draw in the background first
  - Then, only when complete, swap visible screen for background screen

- Use asynchronous contact with OS
  - We still need OS messages for anything outside of our program
  - keyboard, display and possibly contact with eye tracker
- Synchronous calls wait until request is finished and then continue in your experiment
  - Your program is effectively paused while it waits
  - No display, gaze feedback, button detection during this time
- Asyncronous sends request then comes back immediately to your code without delay
  - OS is then going to deal with the request while your program does other things
  - Async versions of: Detect keyboard, detect mouse, draw to monitor, get eye position, …

programming




```
Experiment preparation{
  %Do slow hard drive access here
  %like loading in stimuli
}

While more trials{
  %or for loop of course
  %prep individual trials
  %anything you can pre-calculate

        Single trial timing loop{
          %This is critical MILLISECOND timing

          % if statements
          % stim display
          % get gaze position
          % detect response
        }
}
```

Example of display expt

- Trial starts
  - Send message to eye tracker of new trial, trial number

Inside Timing loop

- Check time: Draw Fixation
  - Send message to eye tracker that fixation was drawn
  - Request gaze position from eye tracker and give error to subject if fixation not maintained
- Check Time: Draw target
  - Send message to eye tracker
- Check Button press (reaction time)
  - Send message to eye tracker,
  - Calculate Button Reaction time (local button press time – local target display time)
- Check saccade (reaction time)
  - Request gaze/saccade information
  - Calculate SRT (local time gaze information arrives – local time target presented)

- Send plenty of event messages to the eye tracker to let it know what is happening in your experiment
- The eyetracker saves a massive file of all gaze, event and messages that you can look through later in case you want to analyze something you didn't anticipate
- Warning though, don't send or receive messages every ms, since message buffer can overflow.
- In general, anytime the display changes, or the subject does something, let the eye tracker know

Message types

- Samples
- Events
  - Eye link events (receive)
  - Message events (send)
  - AOI (may be either eye or message)
- Show slide of information
- Show asci file of embedded information

Eye tracking output

samples

- X,y gaze position
  - Usually in screen pixel coordinates
- Time stamp in ms
- Pupil size
- Error codes

High end eye trackers give both
Some only give samples

events

- Saccade (start, end)
  - Time (ms)
  - X1,y1,x2,y2
  - Peak velocity
  - Duration
  - latency
- Fixation (start, end)
  - Time(ms)
  - X,Y
  - Pupil area
  - Duration
- Blinks
  - Time (ms)
  - Duration
  - These may be conflated with saccades
  - (saccade start, blink start, blink end, saccade end)
- User messages
  - Time (ms)
  - String

MSG 253669 -2 DISPLAY_FIXATION
MSG 253669 -2 !V DRAW_LIST ../../runtime/dataviewer/sub01/graphics/VC_2.vcl
MSG 254672 0 TIMEOUT_FIXATION

MSG 254678 SOUND_CUE
MSG 255063 0 TIMEOUT_SOUND_CUE
MSG 255075 -8 DISPLAY_CUE
MSG 255168 0 TIMEOUT_CUE
MSG 255175 -8 DISPLAY_TARGET
MSG 255175 -8 !V IAREA FILE ../../runtime/dataviewer/sub01/aoi/IA_2.ias

EFIX R 253621	255196	1576	1002.5	577.1	1272			
SSACC R 255197								
ESACC R 255197	255241	45	1012.7	566.8	1364.7	600.1	5.98	200
SFIX R 255242								
EFIX R 255242	255443	202	1356.9	604.6	1293			
SSACC R 255444								
ESACC R 255444	255481	38	1355.4	607.6	1006.9	579	5.91	280
SFIX R 255482								
EFIX R 255482	256939	1458	1020.6	577.2	1285			
SSACC R 256940								
SBLINK R 256958								
EBLINK R 256958	257022	65						
ESACC R 256940	257075	136	1027	585.3	1021.9	572.5	0.24	500
SFIX R 257076								
EFIX R 257076	258443	1368	1017	583.8	1273			
SSACC R 258444								
SBLINK R 258463								
EBLINK R 258463	258528	66						
ESACC R 258444	258570	127	1020.4	587.5	1014.4	620	0.57	446
SFIX R 258571								
EFIX R 258571	258732	162	1002.2	582.3	1282			

Experiment types

Why does timing matter (part 2)
When does timing matter

Experiment types 1: simple

- Offline analyses
- Send messages only
  - No timing problems (or at least experiment and eye tracking timing problems are independent of each other)
  - Send message on time and only once
- Analyze fixation location after experiment
- Visual search, memory strategies, visual world

Awareness

- Does fixation lead to awareness?
- Evidence from change blindness

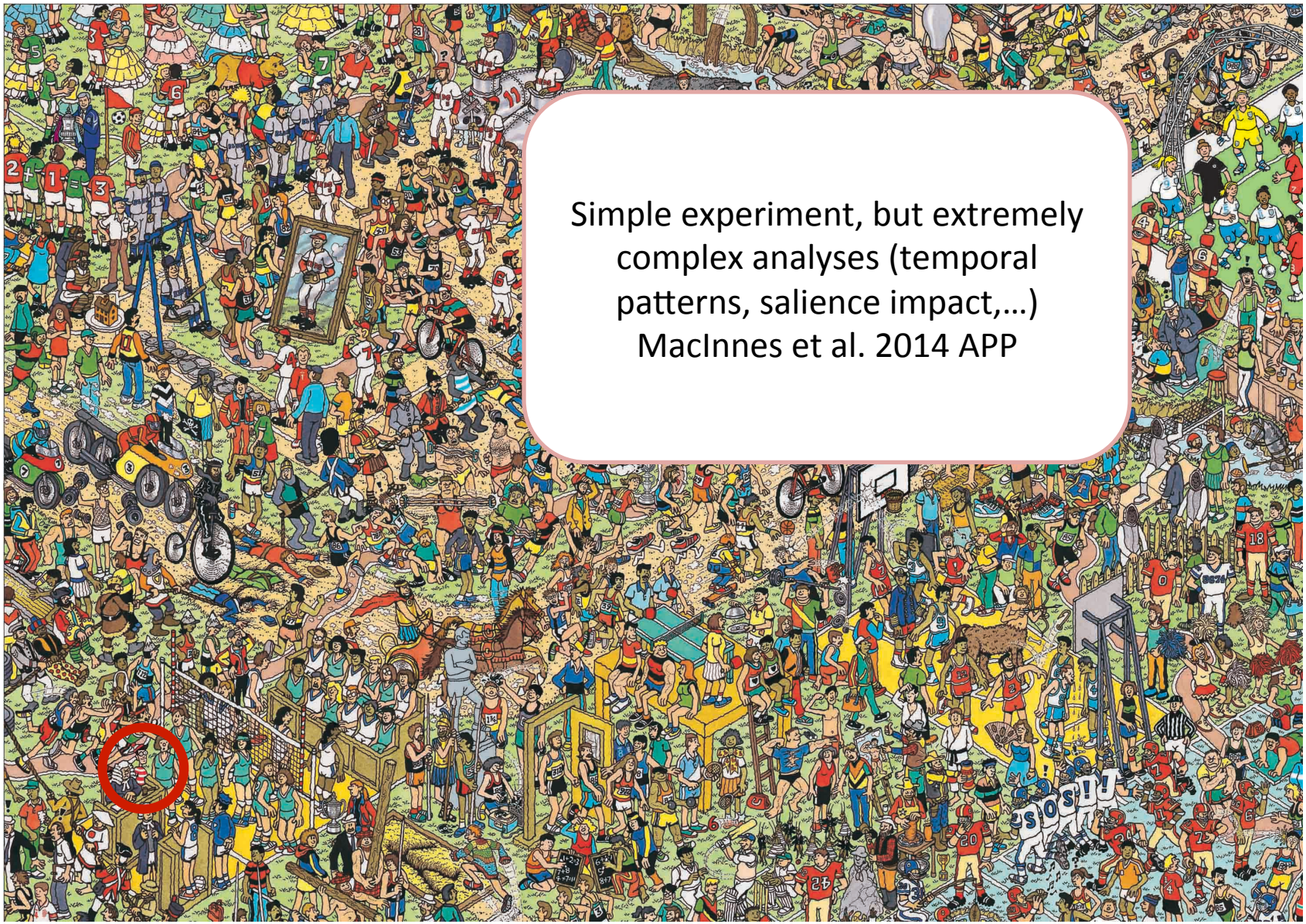
- Unnoticed changed are still processed to a degree
  - (Rensink, 2004: forced choice)

- Overt attention improves change detection compared to eyes restricted
  - (Hollingworth et al, 2001)

- Fixations near the change are good predictors of finding the change
  - (Henderson & Hollingworth, 2001)

- But direct fixations do not guarantee detection
  - (Treisch et al, 2003; Caplovitz et al, 2008)
  - ‘attentive blank stares’

Visual Search



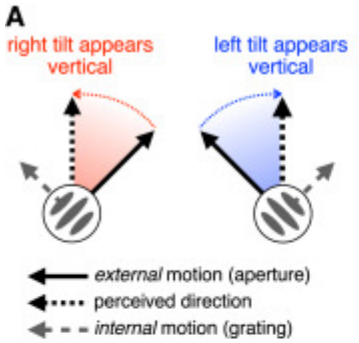
Simple experiment, but extremely complex analyses (temporal patterns, salience impact,…)
MacInnes et al. 2014 APP

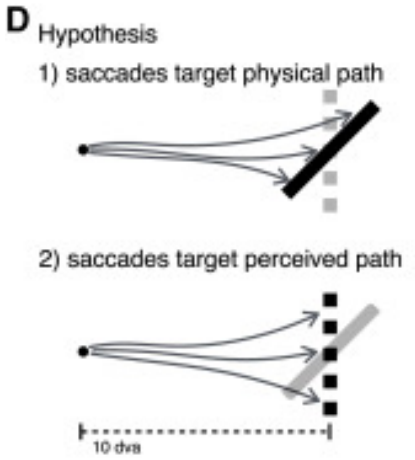
Experiment types 2: gaze aware

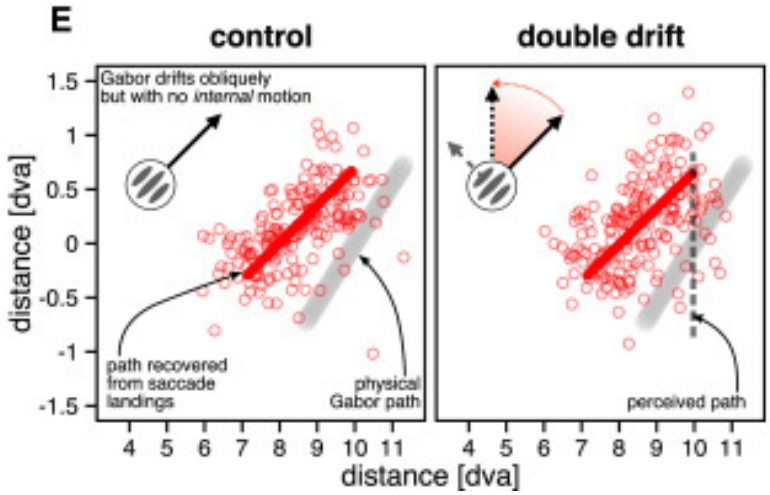
- Fixation/gaze control
- Send messages plus receive gaze location
  - Error message if left fixation, feedback on correct saccade
- Delays in timing may
  - Delay primary stimulus onset
  - Cause inappropriate feedback (wrong event)

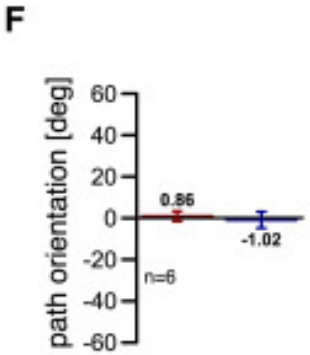
Illusion

- Separate awareness and saccadic programming? (Lisi & Cavanagh, 2015; and others)
  - https://gfycat.com/gifs/detail/GenuineGrimyAfricanrockpython
  - Another case of action vs perception









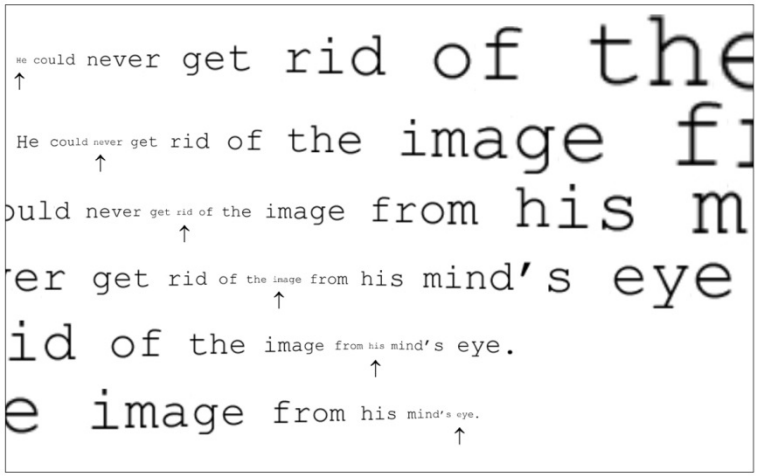
Gaze contingent

- All of the previous complexity, PLUS
  - Change experiment display based on eye tracker information
  - Requires precise timing from both sources and COORDINATION of timing from both

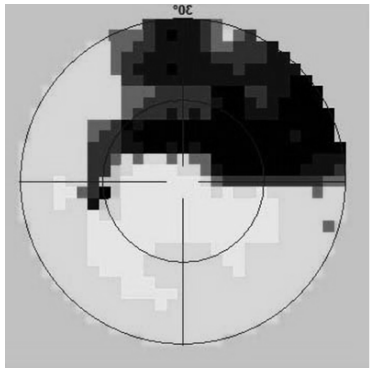
- Example 1: Samples
  - Gaze position in real time
  - Covert visual search/reading, gaze location feedback (usually not aware)
  - Spatiotemporal error: difference bwtween actual gaze location and reported gaze location
  - Requires low latency on gaze position to minimize spatio temporal error
  - Requires good calibration for spatial accuracy (less if large circle)

1. Gaze contingent (foveal)

Reading: parafoveal enhanced



Viewing: parafoveal blurred



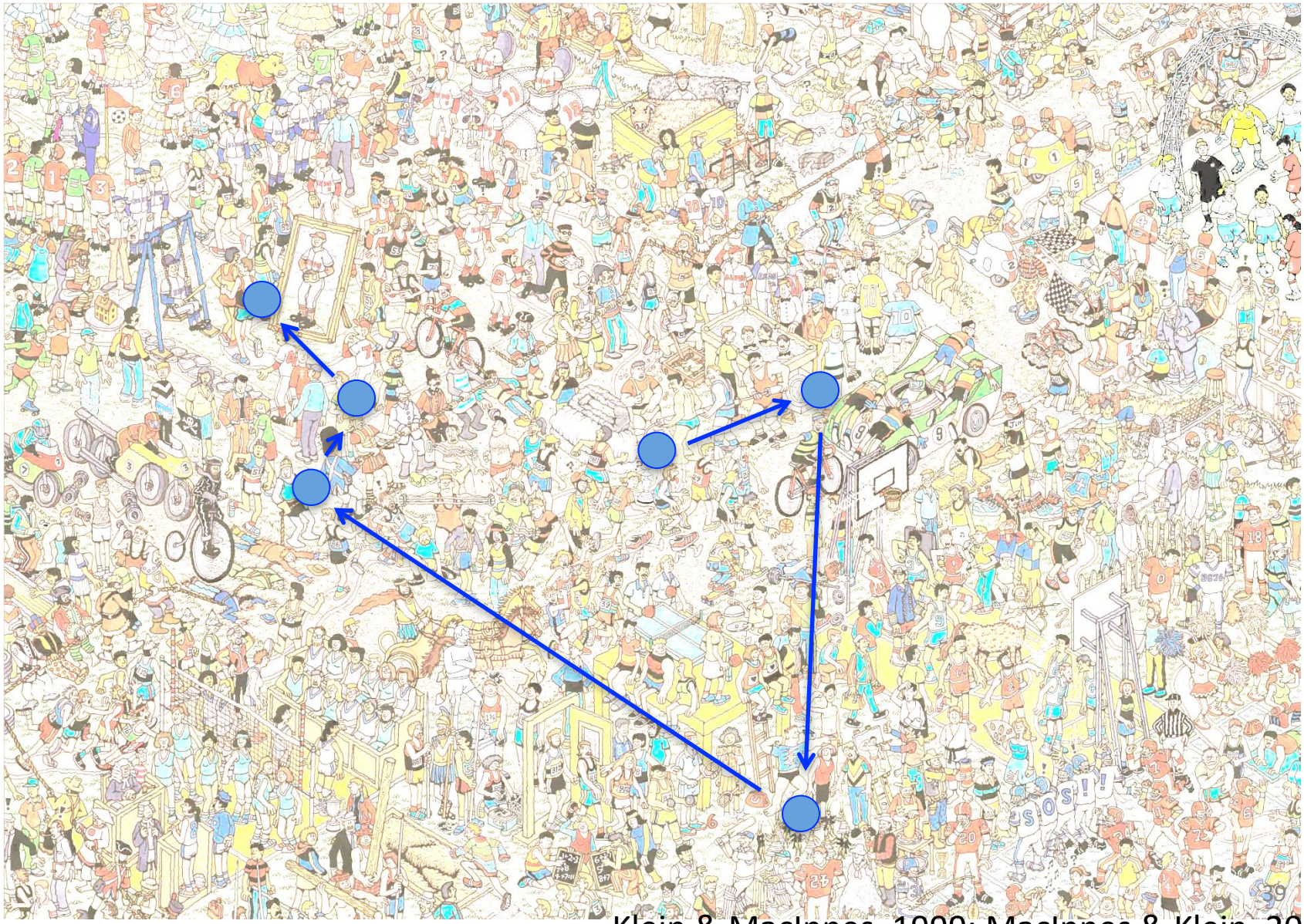
Simulate scotoma; parafoveal or foveal removed

Timing lag will mean off-centre scotoma
Timing delays will cause jumpiness in animation

Gaze contingent

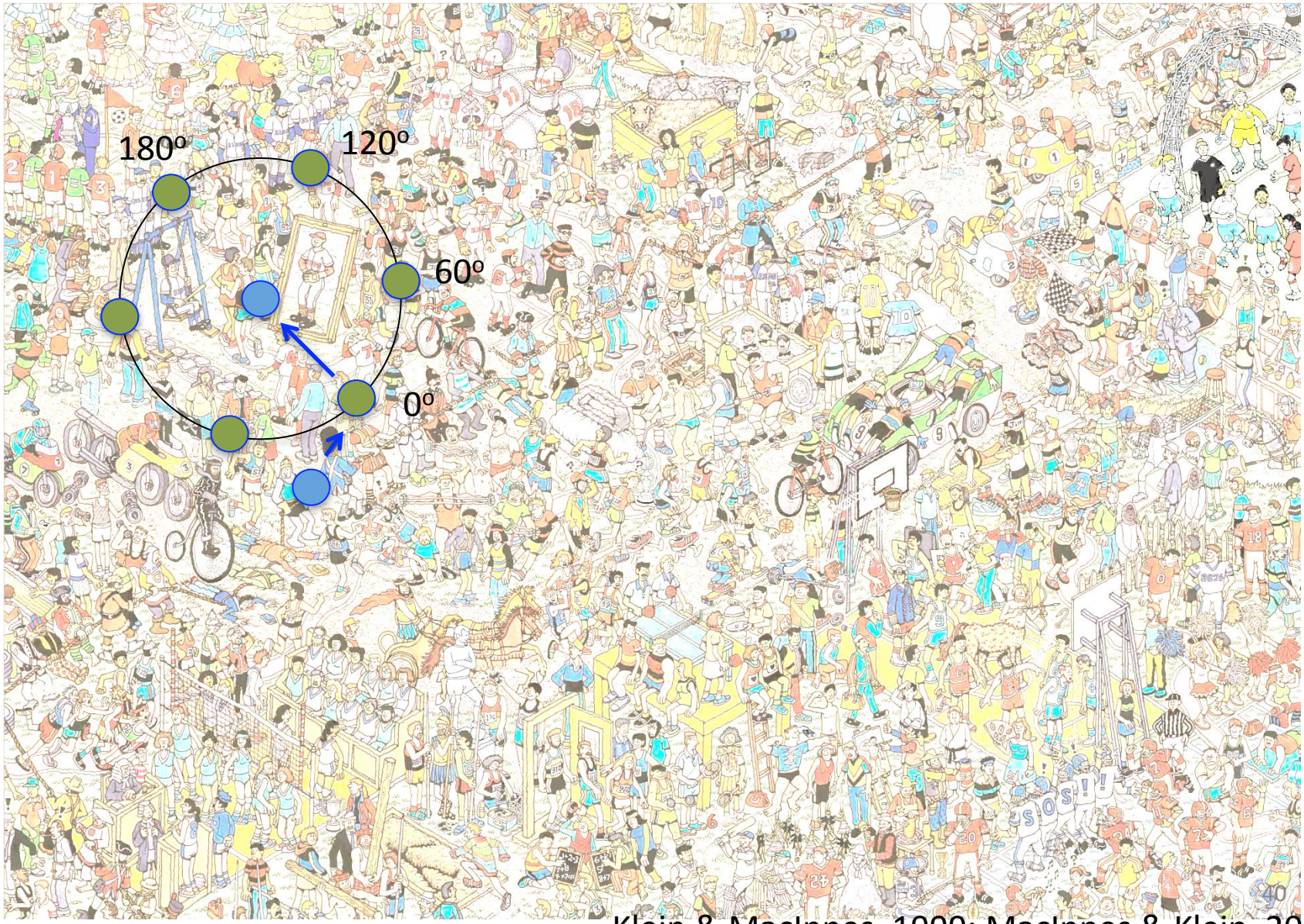
- Example 2: Events
  - Onset, Where's wally IOR
  - Turn on location based on previous gaze
  - Events in real time (fixation started) and fixations require multiple samples to calculate (velocity)
  - 150-300ms fixation. Minus 2-4 ms gaze latency, Minus 10 ms sample fixation detect, 10ms screen refresh
  - As long as new target appears before next saccade programming begins.

Fixations: IOR in complex search



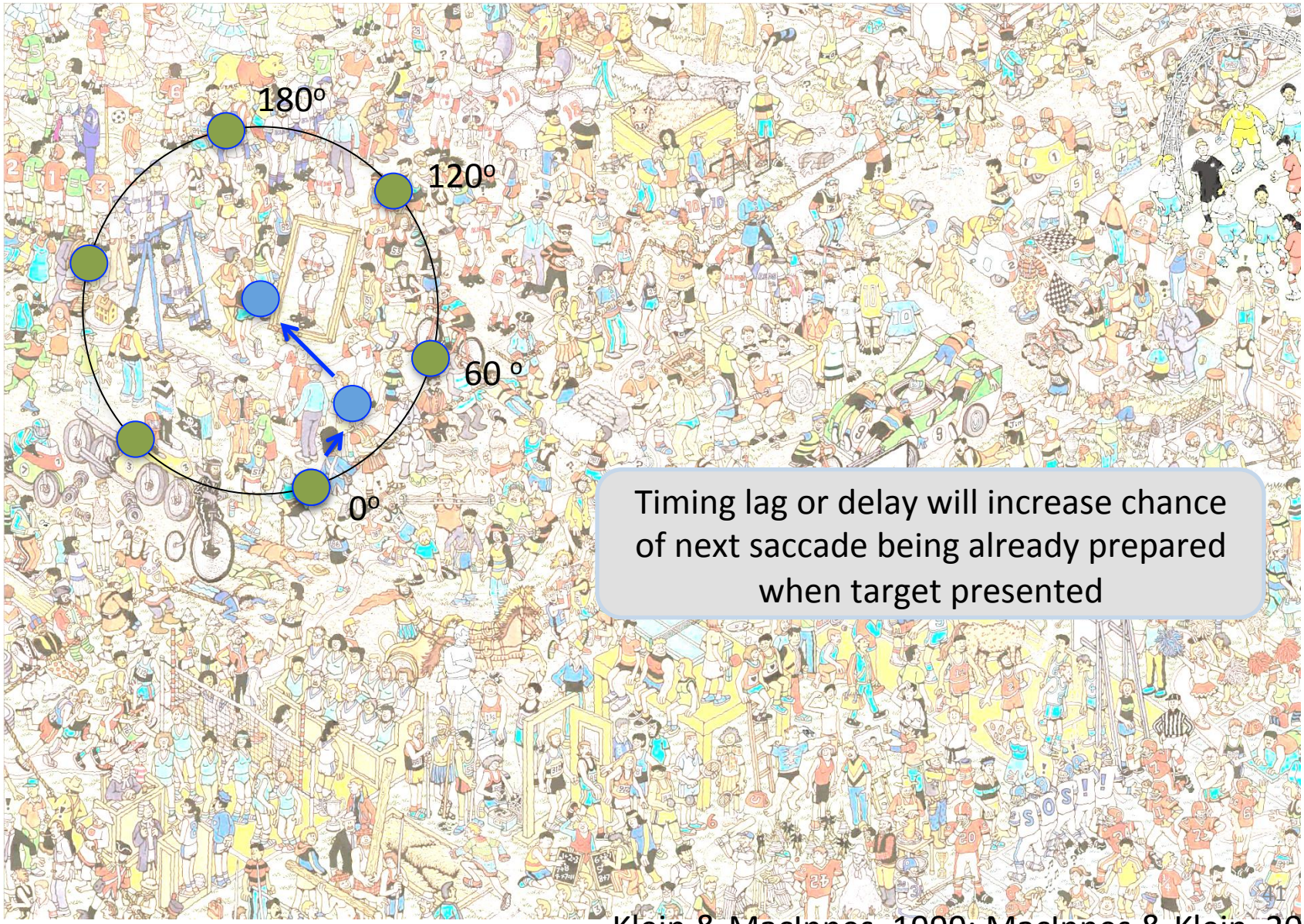
Klein & MacInnes, 1999; MacInnes & Klein, 2003

Fixation contingent: one-back



Klein & MacInnes, 1999; MacInnes & Klein, 2003

Fixation contingent: two-back



Timing lag or delay will increase chance of next saccade being already prepared when target presented

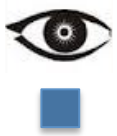
Klein & MacInnes, 1999; MacInnes & Klein, 2003

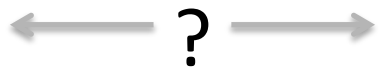
Example 3

- Saccadic suppression of displacement
  - Change location in mid saccade
  - Events in real time (saccade started) and saccades require multiple samples to calculate (velocity)
  - 20-45ms saccade duration. Minus 2-4 ms latency, Minus 10 ms fixation detect, 10ms screen refresh???
- Longer saccade, longer duration
  - Saccade onset, or gaze location trigger
  - Minimal eye tracker latency (2-4 ms)
  - High refresh monitor (144hz < 7ms)
  - Minimal saccade onset detection (6-10ms)
  - Long amplitude saccades!
  - But still send message of visual change completed and exclude trial if uncertain

Saccadic Suppression of Displacement (SSD)

Timing lag or delay will increase chance of movement after saccade has ended. More trials discarded



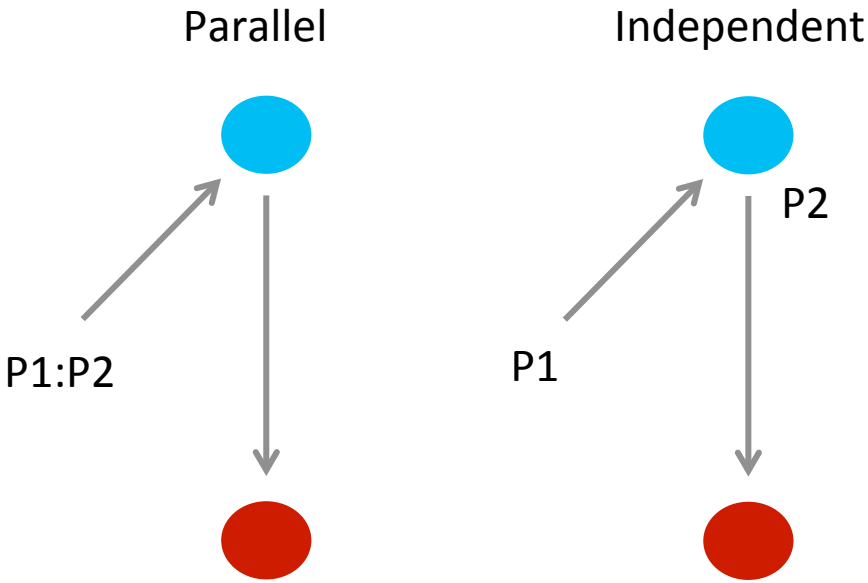


Bridgeman, Hendry, & Stark, 1975

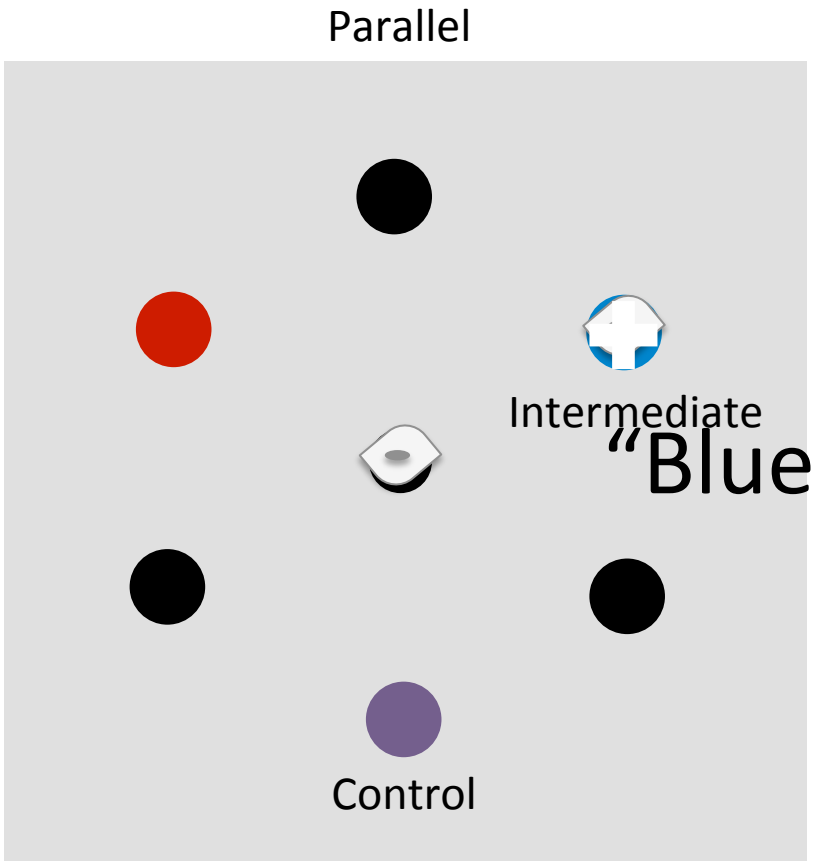
Gordienko & MacInnes, 2016

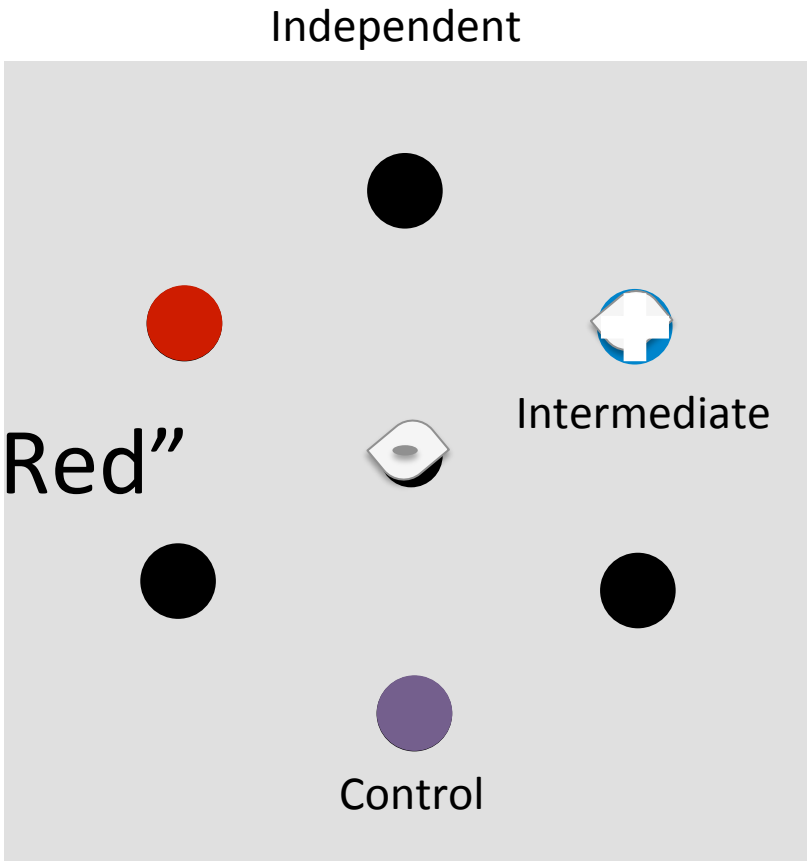
Saccade sequence

- Saccades in a sequence can be planned together in **parallel** or as **independent** saccades (McPeek, etal 2000)
  - The intermediate location was not the source of saccadic program
  - But it is still attended (Godijn & Theeuwes, 2003)
- Gaze contingent eye tracking allows changing display in mid saccade



Saccade sequences

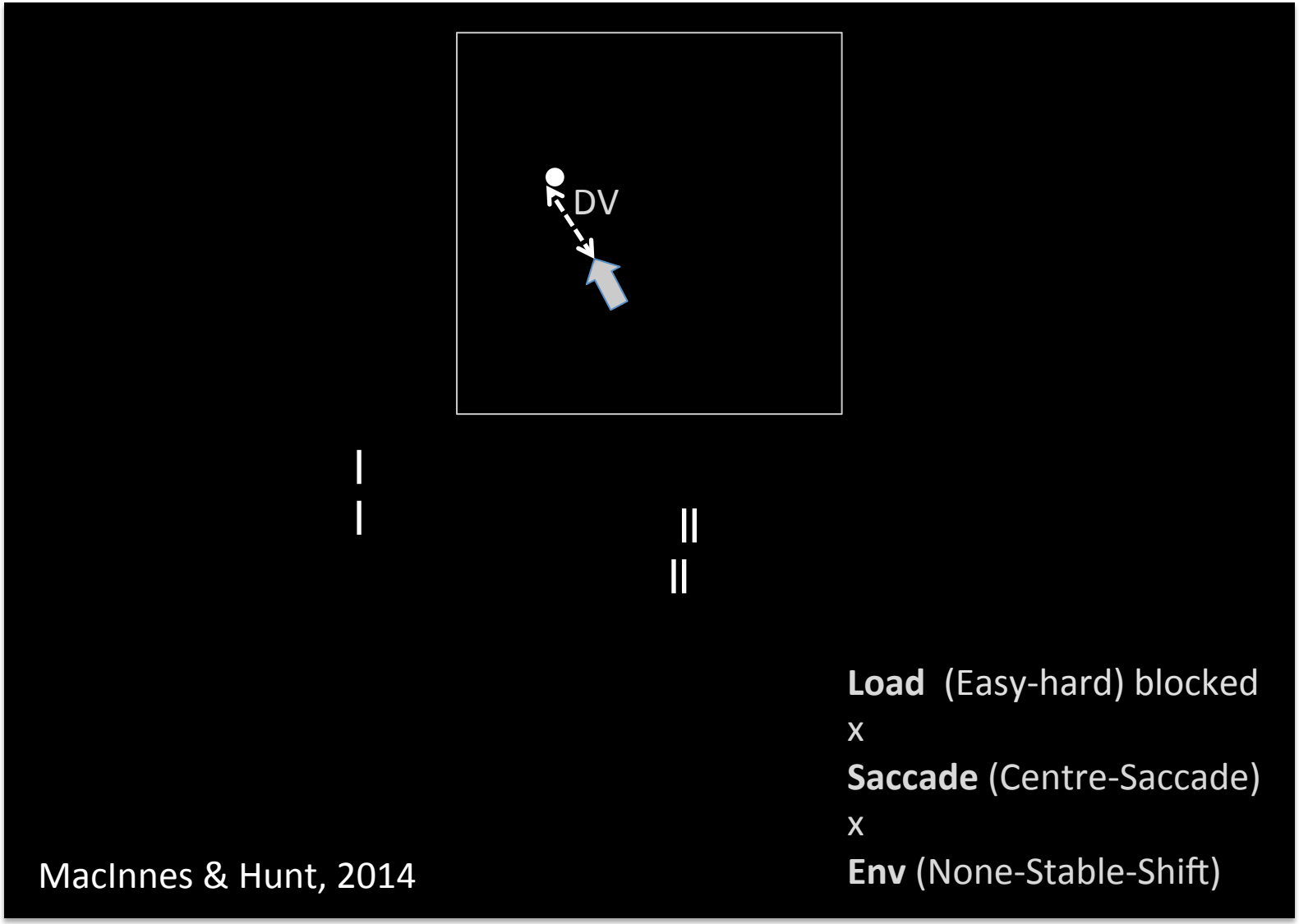




"Blue, Red"

MacInnes, Krüger & Hunt, QJEP, 2015

E2 : Unstable gaze contingent landmark



Summary

- Not all experiments require extreme attention to timing concerns
  - Make sure you know which ones do
- All of your experiments will benefit from extreme attention to timing concerns
- Complex designs, like gaze-contingent experiments allow you to ask questions in a way that others might not be able to
- We *think* that Open Sesame is an example of experiment software that does this right