(1) Overview

Title
A random dot kinematogram for online visual psychophysics

Paper Authors
1. Rajananda, Sivananda;
2. Lau, Hakwan;
3. Odegaard, Brian

Paper Author Roles and Affiliations
1. Designed software; coded software; tested software; uploaded software to GitHub & CodePen;
   Affiliation: University of California-Los Angeles
2. Designed software
   Affiliations: University of California-Los Angeles
   Brain Research Institute (UCLA)
   Hong Kong University
3. Designed software; proofread code; tested software
   Affiliations: University of California-Los Angeles

Abstract
Online experiments using visual stimuli have become increasingly common in recent years, but many frequently-used stimuli in visual psychophysical experiments have yet to be developed for web-based platforms. Here, we introduce the first open-access random-dot kinematogram (RDK) for use in web browsers. This fully-customizable RDK offers options to implement several different types of noise (random position, random walk, random direction) and parameters to control aperture shape, coherence level, the number of dots, and many other features. We include links to commented JavaScript code for easy implementation in online experiments, as well as an example of how this stimulus can be integrated as a plugin with an online experimental library (jsPsych).

Keywords
Random dot kinematogram, random dot motion, moving dot stimulus, online experiments, visual psychophysics, visual stimuli, JavaScript, jsPsych

Introduction

Over the last several decades, random dot kinematograms (RDKs) have emerged as an effective psychophysical stimulus to evaluate low-level motion processing [1–5]. While different versions of this type of stimulus have been created across several software platforms (e.g., MATLAB and PsychoPy), currently, no implementation of this stimulus exists for public use in online experiments. Recently, our research group developed an RDK in JavaScript to be used for presentation in standard web browsers. This RDK incorporates several distinct features that have emerged in different versions used by researchers, and allows users to customize noise types and other parameters for different paradigms. In this short article, we explain a few important elements of this new stimulus and provide links to both raw code and online examples to guide implementation for vision scientists in future experiments.

Implementation and architecture

In RDKs, a certain percentage of dots are designated as "signal" to move in one coherent direction, and the remaining percentage of dots are designated as "noise" to move in random directions. However, as noted by [6], several options exist regarding how signal and noise can be drawn in frame-by-frame presentations. To create the "signal," dots can either move in the direction of coherent motion in *all* frames (referred to as the "same" rule), or move in the direction of coherent motion in only a specified proportion of frames (referred to as the "different rule"). To create the "noise," dots can either be drawn in a random position in the aperture on each frame ("random position"), move to an adjacent position in a random direction in each frame ("random walk"), or have a consistent direction of motion, designated randomly at the beginning of the trial ("random direction"). Modeling our stimulus after the random dot kinematogram from the software PsychoPy [7,8], we parameterized these different combinations of signal and noise to yield six different display options.

Additionally, dots in RDKs have a "dot life", which determines the number of frames that pass before a dot disappears and reappears somewhere else within the aperture. However, if a dot reaches the end of the aperture and its dot life has not ended, then a dot can either reappear randomly in the aperture, or be reinserted from an opposite edge. We include features to customize dot life length, the reinsertion rule, as well as the number of dot sets to cycle through per frame, adapting procedures from established RDK versions in MATLAB [9].

Finally, we also include links to code that integrates this RDK as a plugin with jsPsych, a library for creating and running experiments in web browsers [10]. We include parameters to control motion direction, coherence level, the total number of dots, dot color, aperture shape, aperture size, and how far dots move from one frame to the next. While our raw code can be implemented on any platform that uses JavaScript, incorporating our plugin with the jsPsych library may be particularly advantageous for researchers looking to use this RDK in experiments assessing

reaction times; a comparison of reaction times assessed with jsPsych and a standard software package (e.g., Psychophysics Toolbox in MATLAB), revealed that while reaction times measured by jsPsych tended to be slightly slower than Psychophysics Toolbox, response time variability was quite comparable between both software packages [11]. This indicates that response time measurements in jsPsych are sensitive enough to detect differences caused by experimental manipulations.

Quality control

One primary concern in conducting visual psychophysical experiments online is that timing issues in the display could arise due to differences in internet connectivity speeds, monitor types (i.e., liquid crystal displays vs. cathode ray tubes), hardware, or web browsers used by participants. In testing our new software, we have found that running the exact same code for our RDK in different browsers yields slightly different results. For example, we have identified small differences in the average frames per second when testing our RDK using Google Chrome, Firefox, and Safari on the same computer (Figure 1).

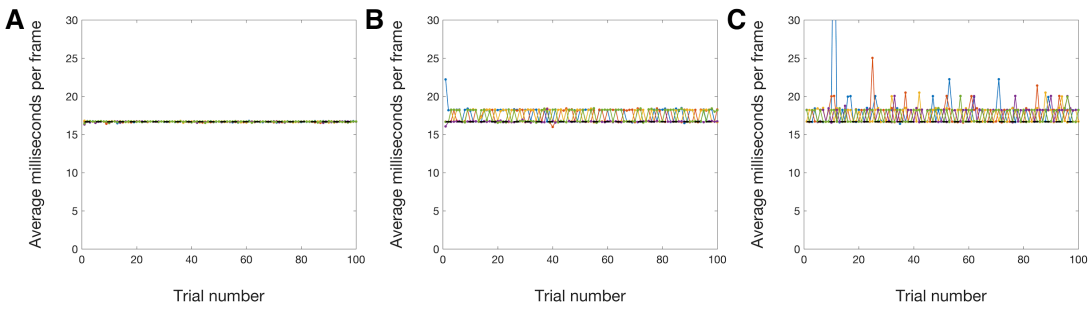

Figure 1. The average number of milliseconds per frame in a test of this stimulus on three web browsers. Shown here is an example of timing differences that emerge when conducting 100 stimulus presentations on the same computer in different browsers. The dashed black line denotes the ideal frame rate for a 60Hz monitor (16.667 milliseconds per frame). Each color represents results from a different run of 100 trials. Five runs in each browser were conducted in total, and all three browsers were run on OS X. (A) The average milliseconds per frame for each trial when the RDK was displayed in Google Chrome. Across five runs, the average milliseconds per frame were consistent. (B) Results from the same test when the RDK was displayed in Firefox. Small deviations occurred in the average milliseconds per frame using this browser. (C) Results from the same test when the RDK was displayed in Safari. This browser was the most variable of the three that were tested.

What is needed to ensure accuracy and precision when conducting online psychophysics are real-time measures of not only the average frame rate of the display, but also the number of frames that are actually presented. Our RDK records the number of frames used in each presentation, which serves as a valid index of presentation clarity and coherence (Figure 2). We recommend that users analyze this data in one of two ways: offline, to exclude particular trials from relevant analyses, or online, to re-present trials with missed frames to ensure balanced numbers of trials across conditions. In our tests of this stimulus, we did not notice visible differences between trials when 11 or 12 frames were used, but a visible stutter was apparent when the number of frames was below 10. Quality checks on other aspects of our stimulus (e.g., the locations where dots are re-drawn after

disappearing, the number of dots drawn in the stimulus, whether the colors are rendered properly) demonstrated its viability, but we welcome feedback from users if issues arise.

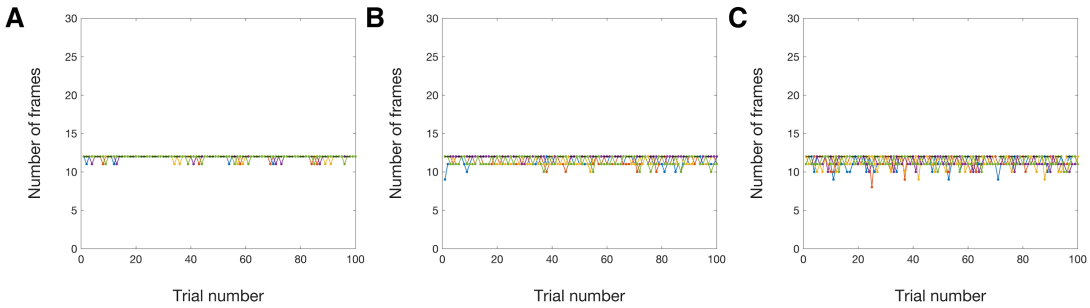

Figure 2. The number of frames displayed per trial in a test of this stimulus in three web browsers on a computer running OS X. In this test, we displayed our RDK for 200ms on each trial for 100 consecutive trials. Shown in each panel are results across five different runs in each browser, with each run denoted by a different color. (A) The results of the test in Google Chrome. This browser showed the most consistency across the three that were tested, and matched our targeted goal of 12 frames on the majority of trials. A small number of trials deviated slightly, using 11 frames to present the stimulus. (B) The results of this test in Firefox. This web browser was slightly more variable, containing more trials that used 11 frames, and even a few trials that displayed 10 or fewer frames. (C) The results of this test in Safari. This web browser displayed the most variability in the number of frames that were displayed. We note that while we did not see visual differences between trials when 11 and 12 frames were presented, trials with fewer than 10 frames appeared to "stutter" when displayed in the browser window. Thus, we recommend that users analyze this metric to evaluate which trials should be dropped from analysis or re-presented when using this stimulus in online experiments.

Finally, we also recommend that users build in explicit instructions to only have one browser window open when participating in experiments with this stimulus and to close other programs while it is being used, as we have noticed slight timing idiosyncrasies that emerge when browser tabs and programs are running in the background.

(2) Availability

Operating system

This plugin is functional in the most recent versions of Safari, Firefox, and Chrome (i.e., the most up-to-date versions available for use in September, 2017). However, based on our tests, Chrome appears to be the most reliable browser in presenting an equal number of frames across trials for this stimulus. We welcome feedback from users if compatibility issues exist in older web browsers.

Programming language

JavaScript/CSS/HTML

Additional system requirements

None

Dependencies
The raw code for the RDK posted on CodePen works without any additional frameworks or libraries. The jsPsych RDK plugin requires the jsPsych library, jQuery, and a jsPsych CSS stylesheet to work properly. These scripts and links for the jsPsych version are added in the header of the main experiment file (see the identifier of (3) under 'Software location' below).

List of contributors
Sivananda Rajananda - Development & Design
Hakwan Lau - Design
Brian Odegaard - Design & Code Review

Software location:
(1) **Demonstration of the RDK stimulus with modifiable parameters**
Code repository
Name: CodePen
Identifier: https://codepen.io/vrsivananda/pen/xLORQe
Licence: GNU General Public License
Date published: 14/09/17
Description: The CodePen link shown above provides users with a visual demonstration of different stimulus attributes; by changing the RDK parameters, users can see (for example) how different signal & noise rules change the appearance of the stimulus. We recommend that users input different values in the "Set Parameters" section of the code to visualize different features of this stimulus.

(2) **jsPsych plugin for use in visual experiments**
Code repository
Name: GitHub
Identifier: https://github.com/vrsivananda/RDK/blob/master/jspsych-5.0.3/plugins/jspsych-RDK.js
Licence: GNU General Public License
Date published: 14/09/17

(3) **Sample jsPsych experiment implementing this stimulus**
Code repository
Name: GitHub
Identifier: https://github.com/vrsivananda/RDK.git
Licence: GNU General Public License
Date published: 14/09/17

Language
JavaScript/CSS/HTML

(3) Reuse potential
Recently, platforms have been created which make it possible to conduct psychological and perceptual experiments online, including the new "jsPsych"

JavaScript library [10] and Amazon's mTurk website for subject recruitment. RDKs are a commonly-used stimulus to evaluate motion perception thresholds in visual psychophysics; our release of the first open-access RDK for online experiments will be of use to visual scientists, psychologists, and any researcher interested in investigating motion perception in large numbers of subjects.

We think use of this stimulus in online experiments will be of particular value to researchers interested in studying populations that may be difficult to access (e.g., the elderly, individuals in various worldwide locations, individuals living far from testing sites, etc.). This new RDK also facilitates fast and efficient data collection in motion perception experiments. In addition, psychophysicists and researchers will be able to develop and extend our code under the GNU license to improve or modify the RDK to serve specific research purposes. We hope our creation here is of use to the greater vision science community and researchers interested in studying motion perception, and that efforts can be taken to adapt other types of visual psychophysical stimuli (Gabor patches, etc.) for online presentation as well.

Competing interests

The authors declare that they have no competing interests.

References

1. Williams DW, Sekuler R. Coherent global motion percepts from stochastic local motions. Vision Res. 1984;24: 55–62. Available: https://www.ncbi.nlm.nih.gov/pubmed/6695508

2. Britten KH, Shadlen MN, Newsome WT, Movshon JA. The analysis of visual motion: a comparison of neuronal and psychophysical performance. J Neurosci. 1992;12: 4745–4765. Available: https://www.ncbi.nlm.nih.gov/pubmed/1464765

3. Roitman JD, Shadlen MN. Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. J Neurosci. 2002;22: 9475–9489. Available: https://www.ncbi.nlm.nih.gov/pubmed/12417672

4. Law C-T, Gold JI. Neural correlates of perceptual learning in a sensory-motor, but not a sensory, cortical area. Nat Neurosci. 2008;11: 505–513. doi:10.1038/nn2070

5. Watamaniuk SN, Sekuler R, Williams DW. Direction perception in complex dynamic displays: the integration of direction information. Vision Res. 1989;29: 47–59. doi:10.1016/0042-6989(89)90173-9

6. Scase MO, Braddick OJ, Raymond JE. What is Noise for the Motion System? Vision Res. 1996;36: 2579–2586. doi:10.1016/0042-6989(95)00325-8

7. Peirce JW. PsychoPy--Psychophysics software in Python. J Neurosci Methods. 2007;162: 8–13. doi:10.1016/j.jneumeth.2006.11.017

8. Peirce JW. Generating Stimuli for Neuroscience Using PsychoPy. Front Neuroinform. 2008;2: 10. doi:10.3389/neuro.11.010.2008

9. Kiani R, Churchland AK, Shadlen MN. Integration of direction cues is invariant to the temporal gap between them. J Neurosci. 2013;33: 16483–16489. doi:10.1523/JNEUROSCI.2094-13.2013

10. de Leeuw JR. jsPsych: a JavaScript library for creating behavioral experiments in a Web browser. Behav Res Methods. 2015;47: 1–12. doi:10.3758/s13428-014-0458-y

11. de Leeuw JR, Motz BA. Psychophysics in a Web browser? Comparing response times collected with JavaScript and Psychophysics Toolbox in a visual search task. Behav Res Methods. 2016;48: 1–12. doi:10.3758/s13428-015-0567-2