

Assessing the benefits of stereoscopic displays to visual search: methodology and initial findings

Hayward J. Godwin^{*a}, Nick S. Holliman^b, Tamaryn Menneer^a, Simon P. Liversedge^a,
Kyle R. Cave^c and Nick Donnelly^a

^aUniversity of Southampton, Southampton, UK, SO17 1BJ; ^bUniversity of York, York, UK, YO10 5GB; ^cKyle R. Cave, University of Massachusetts Amherst, Amherst, MA 01003, USA

ABSTRACT

Visual search is a task that is carried out in a number of important security and health related scenarios (e.g., X-ray baggage screening, radiography). With recent and ongoing developments in the technology available to present images to observers in stereoscopic depth, there has been increasing interest in assessing whether depth information can be used in complex search tasks to improve search performance. Here we outline the methodology that we developed, along with both software and hardware information, in order to assess visual search performance in complex, overlapping stimuli that also contained depth information. In doing so, our goal is to foster further research along these lines in the future. We also provide an overview with initial results of the experiments that we have conducted involving participants searching stimuli that contain overlapping objects presented on different depth planes to one another. Thus far, we have found that depth information does improve the speed (but not accuracy) of search, but only when the stimuli are highly complex and contain a significant degree of overlap. Depth information may therefore aid real-world search tasks that involve the examination of complex, overlapping stimuli.

Keywords: visual search, overlap, stereoscopic displays

1. INTRODUCTION

Visual search is an important activity, being utilized in everyday tasks such as searching for your keys on a messy desk, as well as security-critical tasks such as airport baggage screening [1] and radiology [2]. During visual search, an observer attempts to detect a given target object from amongst a set of non-target (distractor) objects [3]. In a typical laboratory-based visual search task, participants are seated in front of a computer screen and asked to search stimuli (i.e., images) containing objects that do not overlap with one another, as demonstrated in Figure 1. In order to prevent participants from being able to guess whether the target is present, targets are usually presented on 50% of trials (see also [4], [5]). Finally, most search tasks involve asking participants to search for just one target object.

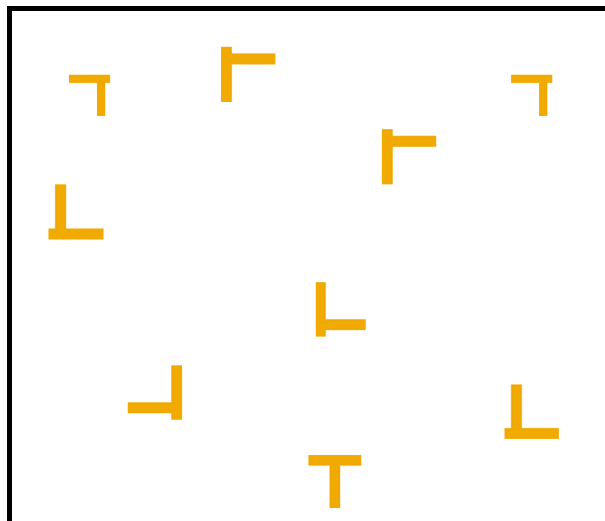


Figure 1. Example of a standard visual search stimulus. Participants are asked to search for a T shape.

Displays such as those in Figure 1 have proven to be highly valuable in terms of being able to use visual search to explore a range of aspects of visual cognition (for a review, see [6]). However, one problem is that these stimuli are markedly different to the types of search tasks that we engage with in everyday life. Many real-world tasks involve searching stimuli (or the environment) in which objects overlap one another, and furthermore, are presented at different levels of depth to one another. In fact, considering real-world search tasks such as searching one's desk for a set of keys, for the face of a friend in a crowd, or for products in a supermarket, it may well be the norm for objects to be overlapping at different depths to one another, rather than the exception. The real world is rarely so organized and sparsely populated with objects, buildings, occluders and people that every object is fully visible and distinct from all other objects.

With that in mind, the goal of the present project is to examine, in detail, the influence of overlap and depth upon search performance. In terms of overlap, previous research has examined visual search performance in stimuli containing overlapping objects. In some studies, this has involved presenting participants with complex and cluttered stimuli, such as computer-generated overviews of cities [7] or complex maps containing many overlapping details for reference [8]. Other studies have used simple images of objects, and placed the objects on top of one another to create overlap [9]. To summarize the results of these studies, it has generally been found that overlap does indeed impair search performance, as measured by Reaction Times (RTs), which increase under the presence of overlap. One problem with these studies is that, although they did use stimuli containing overlapping objects, they did not precisely control the level of overlap in the stimuli. Our first goal was therefore to conduct a series of experiments where the overlap levels between objects were precisely controlled and measured. This enables us to gain a very direct measure of the point at which increasing levels of overlap cause performance to break down.

In terms of effects of depth, there has been conflicting evidence as to whether the presence of depth in stimuli can aid visual search performance (see [10]). However, other studies do suggest that depth can be used as a cue to aid segmentation processes in overlapping stimuli [11]. Our second goal was therefore to introduce different levels of depth into the stimuli. We predicted that the presence of depth in the overlapping stimuli would benefit search performance.

To assess the extent to which overlap impairs search performance and depth aids search performance, we used standard measures of RTs and response accuracy, as well as eye-movement metrics. We anticipated that increasing overlap would slow RTs and decrease response accuracy, whereas the presence of depth in the stimuli would decrease RTs and increase response accuracy. Eye movement behavior was recorded in order to give us detailed insights into the online cognitive processing taking place during search [12]. Put simply, by recording the eye movement behavior of the participants, we were able to determine the changes in behavior that took place as overlap and depth were manipulated, focusing on the types of errors made by participants when they were searching. Specifically, we examined the errors in terms of a taxonomy that has been developed previously [13]–[16]. It has been found that participants often fail to detect targets when searching because they fail to fixate (i.e., look at) the target object. These have been termed errors of *perceptual selection*. Participants also often directly fixate target objects, but still fail to detect them. These have been termed errors of *perceptual identification*. We anticipated that both types of error would be more likely as increasing levels of overlap were introduced, and that both types of errors would be reduced when depth information was available within the stimuli.

In order to fully assess the benefits of depth and problems associated with searching overlapping stimuli, we sought to use a range of different stimulus types. As noted above, since X-ray baggage screening contains complex overlapping transparent images, our goal was to focus first of all on a simplified version of the X-ray stimuli, building up to the most complex stimuli, step-by-step. To that end, we conducted four experiments overall. Two of the experiments involved the presentation of transparent stimuli, and two involved the presentation of opaque stimuli. Using both opaque and transparent stimuli enabled us to compare and contrast the benefits of having the visual information physically removed by the presence of overlap (in opaque stimuli) compared with having the visual information not removed, but obscured by the combination of colors that occurs when transparent objects overlap (see Figure 2 below). The stimulus types required were therefore as follows:

1. Opaque polygons
2. Transparent polygons
3. Real-world images (opaque)
4. X-ray images (transparent)

The opaque polygons and transparent polygons served as a natural counterpart to one another. Their colors were selected based on an existing range of colors that we have used in previous studies [17]. The real-world images were selected to serve as an opaque counterpart to the X-ray images. This was because X-ray images are made to be transparent, and, as such, it was not possible to ‘convert’ or ‘repurpose’ them for use as opaque images. Since the X-ray images were of everyday items that are often carried in baggage, the real-world images were selected to be an opaque equivalent of such items. Examples of the various stimulus types are presented below in Figure 2.

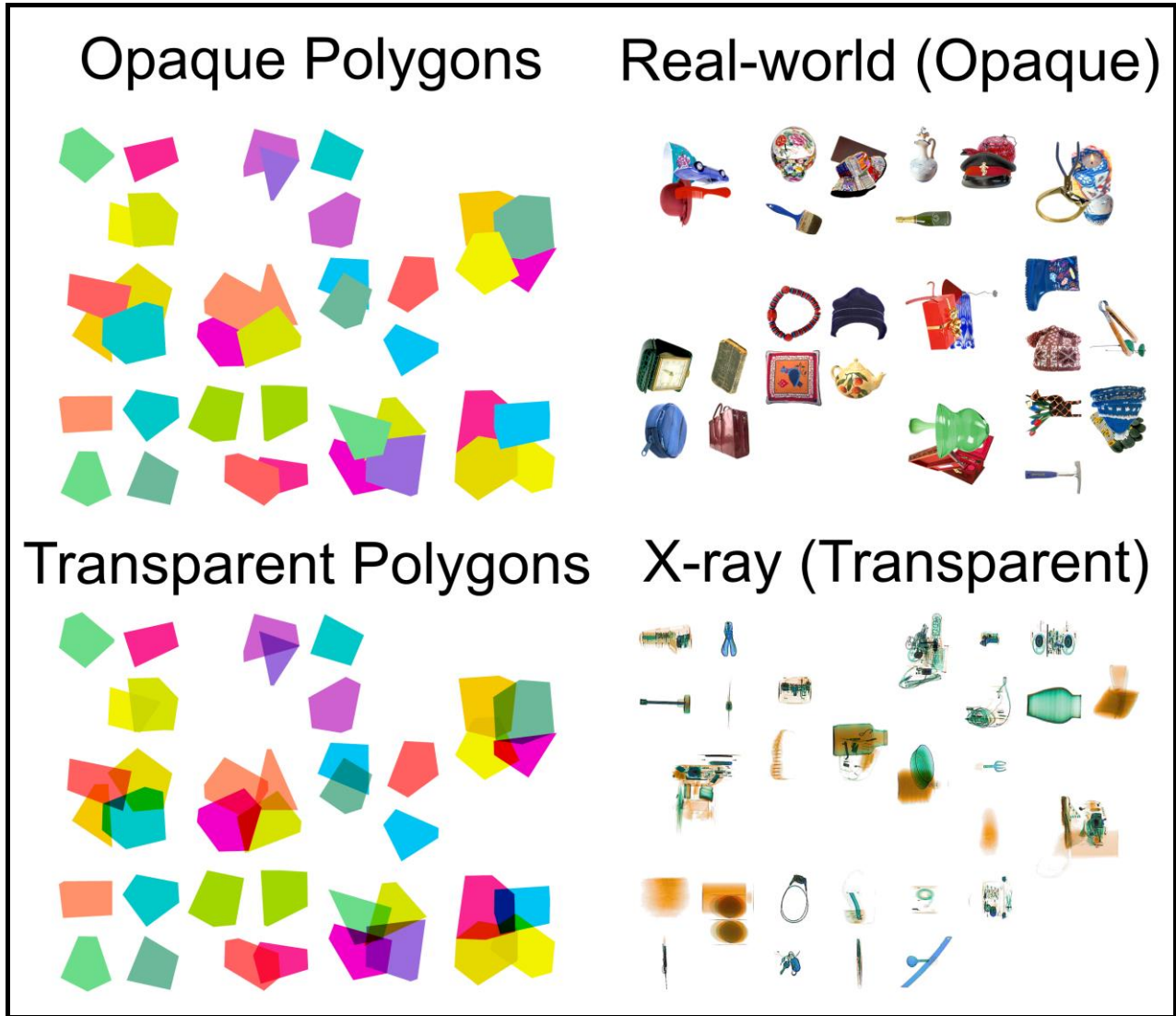


Figure 2. Example stimuli from each of the four experiments.

2. HARDWARE SETUP

2.1 Eye-tracking laboratory

We used an SR Research Eyelink 1000 eye-tracker to record the eye-movement behavior of the participants as they searched the stimuli (see Figure 3 for setup). The Eyelink 1000 offers very high spatial and temporal resolution, and is also able to record eye movement behavior through polarized spectacles, making it an ideal choice for our purposes.

Participants are seated in front of a computer display, facing the eye-tracker. Their head position is stabilized by asking them to rest on a chin rest. Participants respond using a response box, pressing one button to indicate that they have detected a target object, and a different button to indicate that they believe no targets to be present in the display.



Figure 3. Lab setup example. Participant is seated in front of the stereoscopic display (left side) while the experimenter controls the running of the experiment from a separate computer (right side). During actual data collection, the lights are dimmed in the room.

2.2 Stereoscopic displays

After testing various options, we used a Hyundai W243s monitor with a 60Hz refresh rate and a resolution of 1920 x 1200 pixels to present the stimuli to the participants. Participants sat 80cm from the monitor in a dimly-lit room. In order to provide the percept of depth, this monitor requires participants to wear polarized spectacles combined with the presentation of images wherein alternate rows of pixels are setup to be presented to the left and right eyes (see Figure 4).

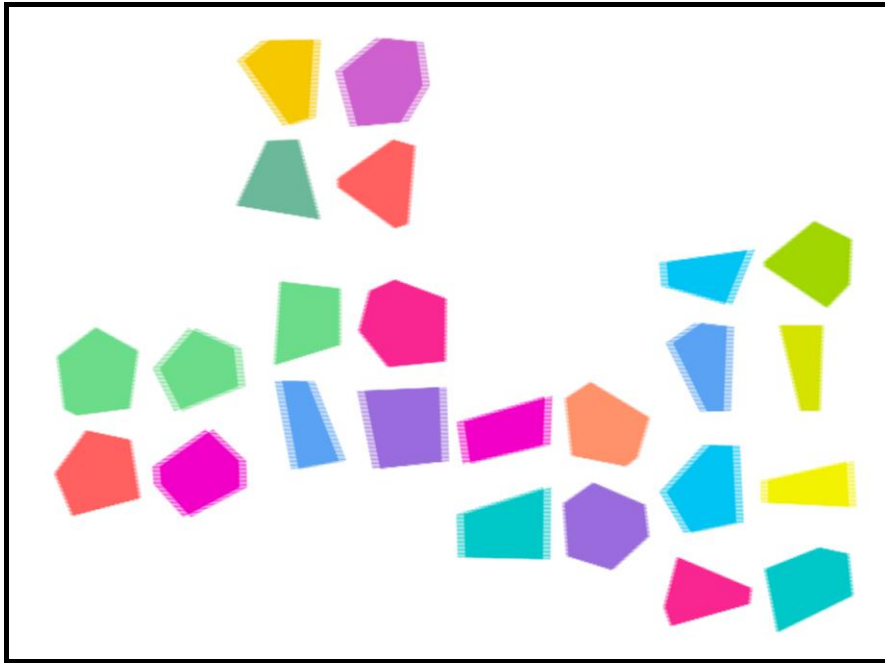


Figure 4. Stereoscopic image example.

3. SOFTWARE DEVELOPMENT: CREATING STEREOSCOPIC IMAGES

3.1 Overview of Experimental Design

The experimental design was as follows. For the four experiments (opaque polygons, transparent polygons, real-world and X-ray), the design was identical. The following factors were varied:

- **Overlap.** Set to 0%, 45% or 90% of each object's area.
- **Setsize.** The number of objects in each stimulus. Set to 24 or 40.
- **Target Presence.** Whether a target was present or absent.
- **Complexity.** Set to high or low. For high complexity, a maximum of four objects could overlap one another at any one point; for low complexity, a maximum of two objects could overlap one another at any one point
- **Depth.** Set to multi-plane (3D) or single-plane. For single-plane, all objects were presented on one depth plane only.
- **Search Type.** The number of targets being searched for. Set to one or two.

For each experiment, we required 32 participants to take part in a total of 1,728 trials. The experiments are substantial: each participant required at least six hours to complete their participation.

3.2 Selection of Target Objects

For all experiments, participants were given up to two target types to search for at any one point in time. There were some differences in target selection depending upon the stimuli, which we will now describe.

For the polygon images, each participant was asked to search for a different pair of polygons that varied both in their color and their shape. This was in order to prevent the results being attributable to the characteristics of a particular polygon, which could be the case if all participants searched for the same two images. Using different target pairs across

different participants required a substantially larger number of images to be generated for the polygon experiments, in order to cater for the fact that different targets needed to be added to the images for different participants.

For the real-world images, participants were asked to search for images of weapons (guns/knives) and toys. For the X-ray images, participants were asked to search for weapons (guns/knives) and Improvised Explosive Devices (IEDs).

3.3 Requirements of the experimental software

We used SR Research Experiment Builder to control the data collection for the experiments as part of this project. Experiment Builder experiments require three key components to control the experiments:

1. **A list of the trials and stimuli that need to be presented to the participants.** The necessary lists were produced using our software as standard .txt files that could be read in by Experiment Builder. A simple interface presented at the start of each data collection session enables the experimenter to select the correct .txt file to be used to control stimulus presentation for that particular testing session.
2. **The stimuli (in a standard image format) that are to be presented to the participants.** The images that are to be presented to participants are produced using our software and saved as .png files that are imported into Experiment Builder.
3. **Interest Area Set (.ias) files.** A key requirement of Experiment Builder (and other eye-tracking software) is a method of ‘marking up’ the stimuli presented to participants in order to identify regions of interest to the experimenter. In the context of these experiments, the regions of interest bounded the edges of each individual object. When analyzing the data collected, it was then possible to determine which object(s) were being examined during each individual fixation. The relevant .ias files required by Experiment Builder were also generated by our software.

3.4 Overview of Search Commander

With a clear view of the requirements of the experimental software, we developed a new piece of software, dubbed *Search Commander*, to output the necessary trial lists, stimuli, and interest area files for our experiments. For this piece of software, we wanted the ability to have a high degree of fine control over the images, and the power and flexibility to output those images rapidly. We also wanted to be able to have a readily-available GUI interface that could aid us in managing the sizeable number of files that needed to be generated. Finally, in order to speed up the process of image generation, we wanted to be able to use Search Commander on separate machines to generate different images in parallel.

With our needs considered, we decided to use C#.net to develop the Search Commander software. Figure 5 provides an overview of the interface.

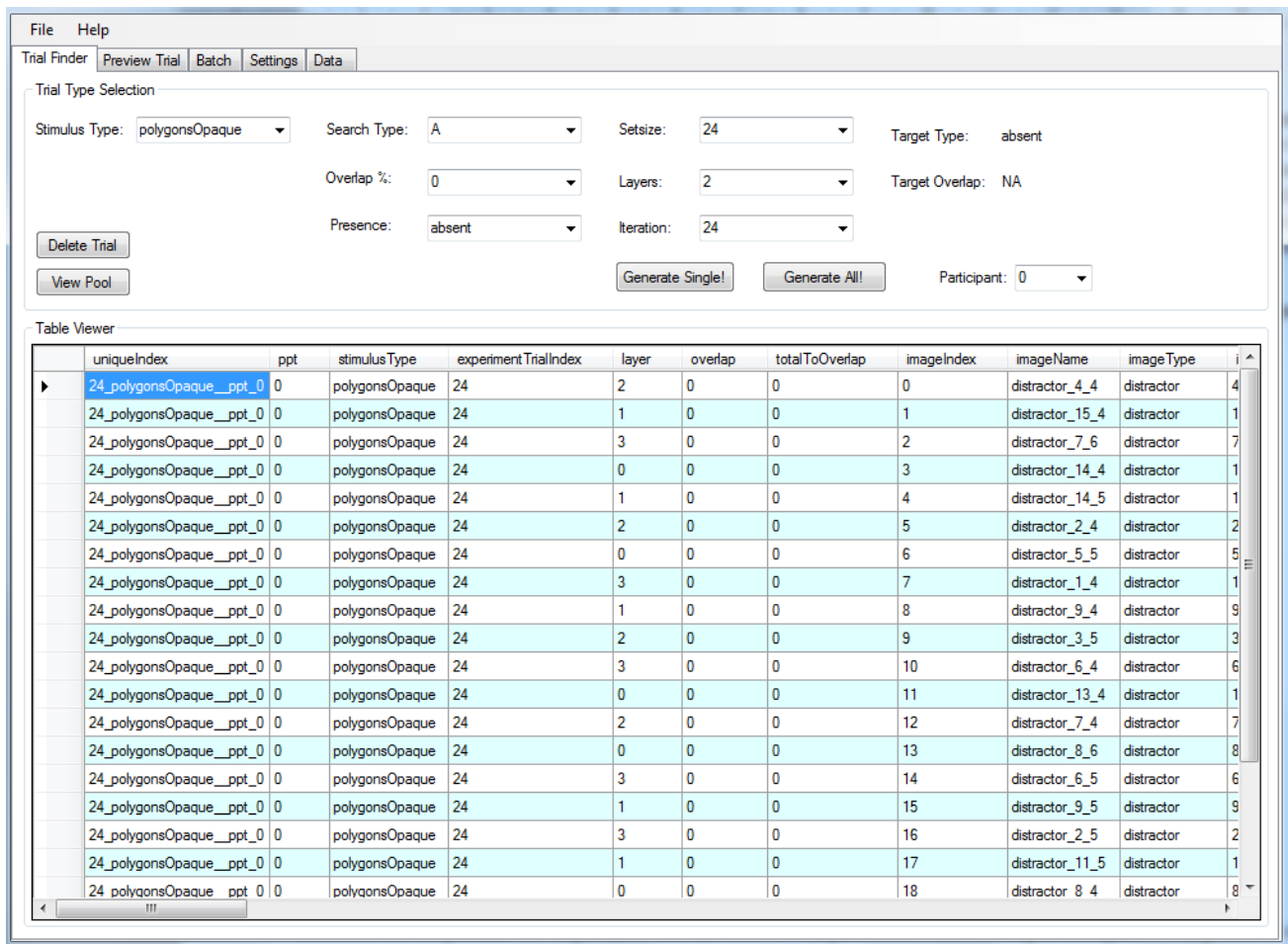


Figure 5. Search Commander interface. Tabs are used to control overall settings; ‘Trial Type Selection’ is used to select, show, delete, or generate a particular trial; the ‘Table Viewer’ is used to examine an extensive set of properties for a given trial that has been selected.

3.5 Database and file management

Since we wanted to be able to batch output trial files on multiple computers in parallel, it was important that we used an appropriate database and file management approach. Furthermore, in order to facilitate any analyses that we wanted to carry out at a later date, it was also important that a standard file format was used to manage the database and any associated files: a file format that could be read by other software (particularly R). To that end, we used XML files to save additional trial information so that it could be accessed at a later point for further analysis.

In order to prevent any duplication of trial information during the parallel batch output process, for each trial type, we specified a pre-defined range of indices that belong to that particular trial type. For example, indices in the range of 0-24 involve searching for one target, with a setsize of 24, 0% overlap, low complexity, and all contain a target. During the batching process, it was then a simple case of outputting the required trial types to avoid producing multiple copies of the same trials on different computers. When instructed to begin a batch output operation, the software first checks to see if a given trial has already been created and saved within the directory tree; if the trial has already been created, the software then skips that trial and proceeds to the next. This prevents it from writing over any already-created trials.

3.6 Randomization of image generation

In a typical visual search experiment, each participant will be presented with unique, randomly generated stimuli (e.g., see [18]). This is made possible by the fact that the placement of individual objects on a blank background, with no overlapping or depth, can be randomized and presented rapidly to participants. Here, since the stimuli required more time

to be generated, particularly in terms of the overlap, combined with the fact that we wanted to check the stimuli containing depth for any potential visual artefacts, we were therefore required to pre-generate all of the images and stimuli prior to data collection. As shown in Figure 5, Search Commander enables a user to delete a given trial; the trial can then be re-generated.

For each individual stimulus, the following steps were taken to generate an image:

1. **Determine trial type from the GUI interface.** The trial type determines, for example, how many objects are to be presented in the stimulus (referred to as the Setsize).
2. **Select the images to be presented.** One of the key requirements for the image selection was that we wanted, overall, the same number of each image type to be presented across all of the trials. For example, we wanted the same number of six-sided orange polygons to appear as six-sided blue polygons. In order to achieve this balance, we utilized a *pooling* approach, whereby a database of object details was maintained. The database contained details of each type of polygon to be used, and, every time one was added to a trial, a counter was incremented. During the image selection process, the images that were selected first to be added to a given trial were the ones that had, up to that point, been used the *least* often. This approach helped to make sure an equal number of selections of each object type. For the real-world and X-ray experiments, instead of using object types, we simply used a counter for each object in the image library, so that, for example, no specific image was used more times than any other.
3. **Randomization of image locations.** Upon each stimulus, we drew out a virtual (i.e., invisible) grid of ‘master cells’. Each master cell contained four objects. We avoided the edges of the display since the percept of depth is reduced to some extent at these locations. Once master cells were selected to contain objects, we ‘jittered’ the master cells within the stimulus, allowing them to move into unoccupied locations. This process reduced the degree to which the stimuli appeared ‘ordered’ or ‘regular’, see Figure 6.
4. **Overlapping of images.** Within each master cell, when overlap was required, the objects were set to overlap with one another, as described in 3.7.
5. **Generation of stereoscopic images.** The images were then interleaved to produce the percept of depth, as described in 3.8.

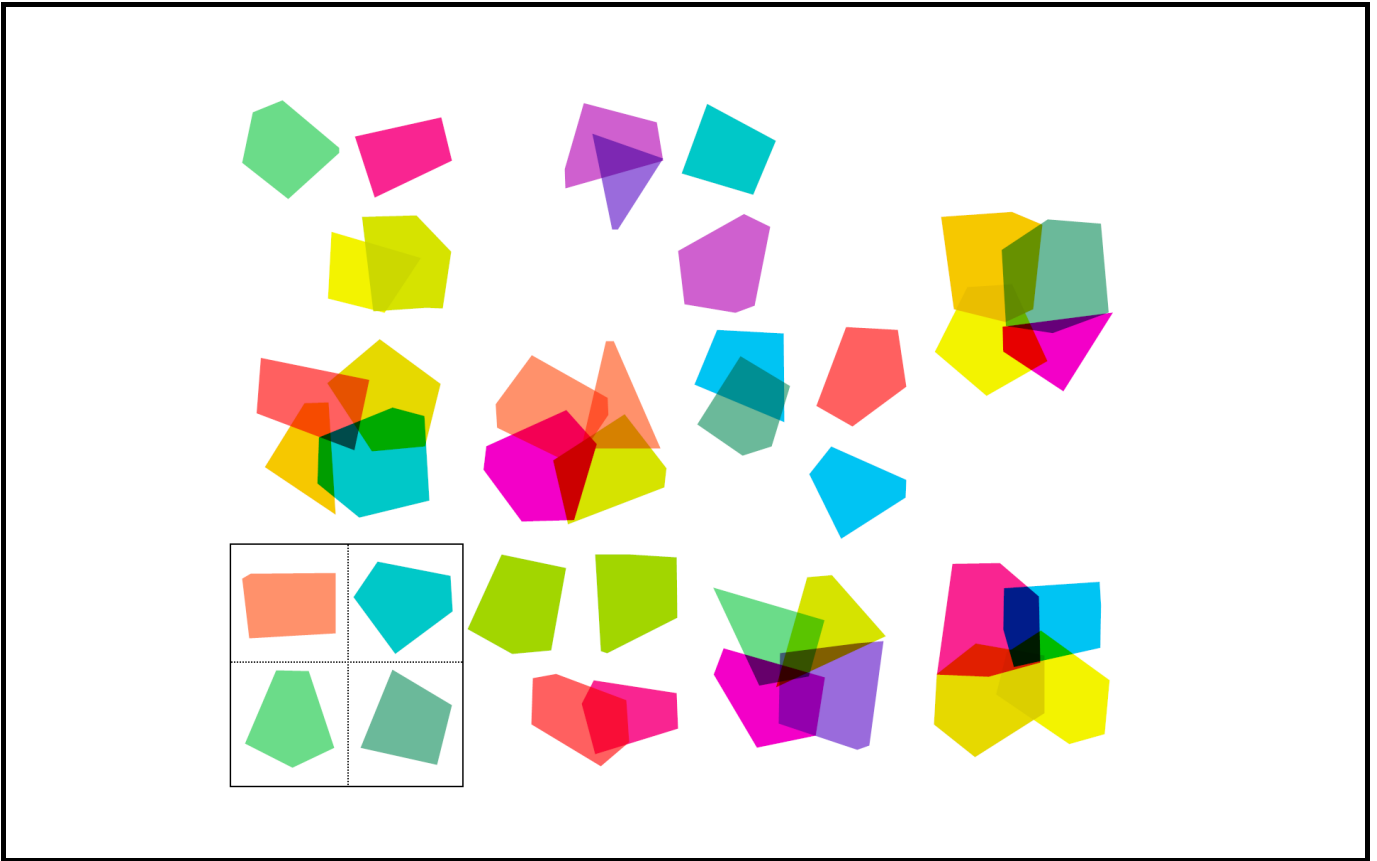


Figure 6. Virtual grid and master grid cell example. Here, a single stimulus has a master grid cell highlighted in a black box (thick border); the inner grid cells containing each object are highlighted with dotted borders.

3.7 Ensuring the correct level of overlap

We used an iterative approach to ensure the correct level of overlap for a given trial. After the level of overlap for a master grid cell was determined and the objects for that grid cell had been selected, the objects that were to overlap others were moved, one pixel per iteration, until the desired level of overlap was attained. During this process, the ‘best fit’ of overlap and position was recorded so that, once the movement of the object(s) went above the desired level of overlap, the ‘best fit’ solution could then be reverted to. Figure 7 illustrates this process.

In order to measure the level of overlap, we maintained a record of overall area for each object, and then, upon each iteration, simply counted the number of pixels that overlapped between the two objects. Given the size of the images, amount of overlap, and number of trials, this was the slowest component of the trial generation process.

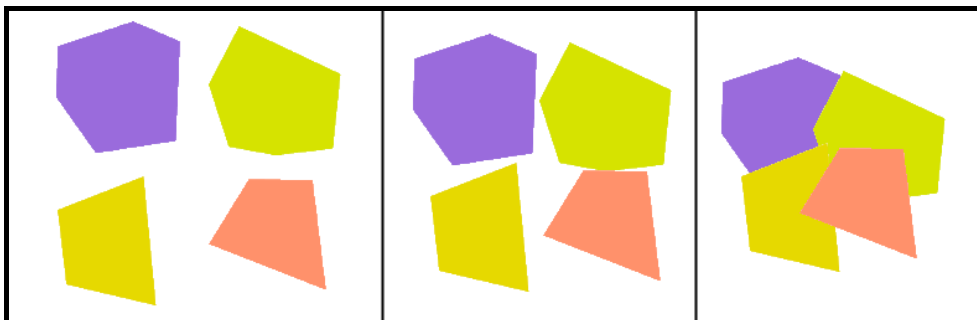


Figure 7. Overlap illustration. The objects began being positioned in their respective inner grid cells (left panel), and were then moved closer to one another (middle panel), until the desired level of overlap was reached (right panel).

3.8 Generation of stereoscopic images

The objects in each stimulus were placed on one of four 'layers'. Two layers were set to appear in front of the plane of the computer monitor, and two were set to appear behind the plane of the computer monitor, with the offset of the interleaved images being set to: -6, -2, 2 and 6 pixels for the respective layers.

In order to measure the overlap in the images, and to interleave and generate the stereoscopic images, and also to calculate the transparency within the images, we used the *lockbits* approach [19] in C#. This was necessary because, although C# does provide a number of image manipulation and drawing functions, they can be relatively slow given C#'s managed nature, making the extensive drawing and image manipulation/measurement that we were engaged in to require a considerable amount of time to complete. The *lockbits* approach involves directly manipulating the bytes in memory that correspond to the RGB values for any given pixel. In order to do so, the image must first be locked into memory and its memory address determined. Each individual pixel can then be iterated through, assessed and modified as necessary at high speed. Once the operations on any image are complete, the image can then be unlocked from memory, re-used or then garbage collected as necessary. After locking the images into memory, they were then modified as follows to produce the required interleaved images for stereoscopic viewing:

1. Determine which objects needed to be included in a particular image. This is outlined above in 3.6.
2. Place the objects in their respective locations on the image. This is outlined above in 3.6.
3. For each layer, move the object locations to the left/right depending upon the disparity for that particular layer.
4. Interleave the left/right images in order to produce the to-be-presented interleaved stimulus. This process is depicted in Figure 8.

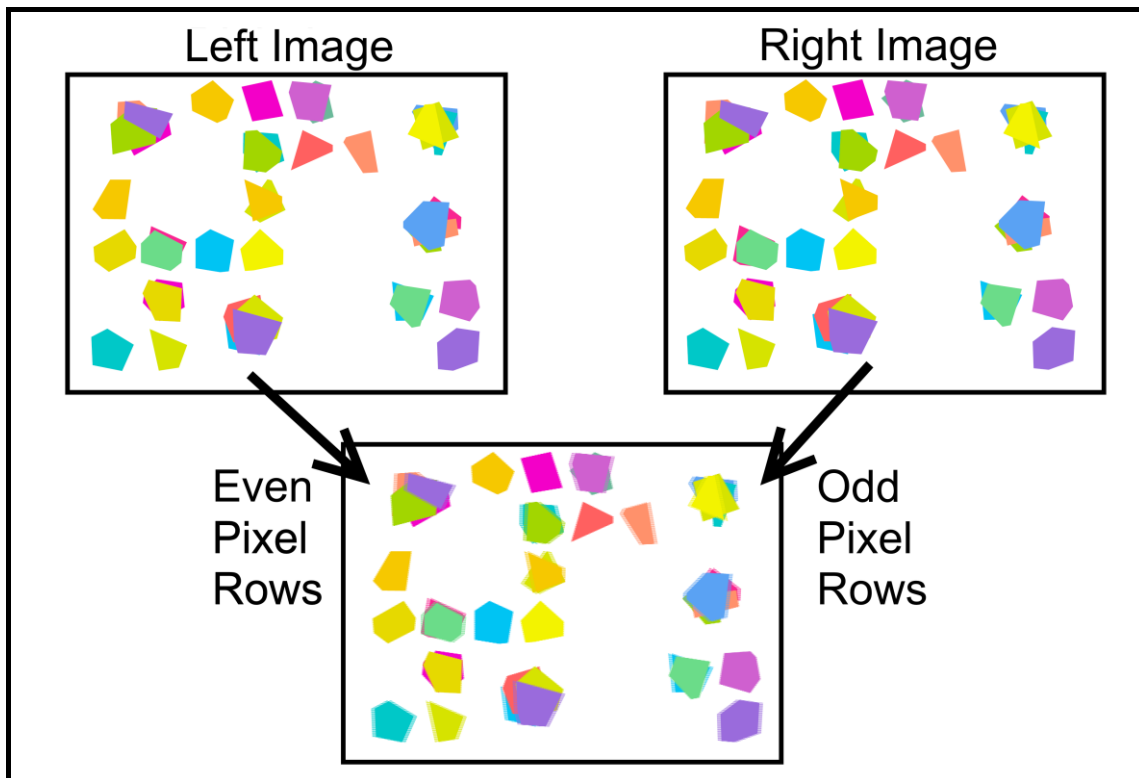


Figure 8. Illustration of the process used to interleave images. A Left and Right Image were created. Even-numbered rows of pixels from the left image were taken and odd-numbered rows of pixels from the right image were taken to form a final stereoscopic image.

4. CONCLUSIONS AND OVERVIEW OF RESULTS

In the present paper, we have outlined the approach and methodology taken as part of a substantial series of experiments targeted at examining the influence of overlap and depth upon visual search performance when searching for targets in stereoscopic stimuli. We are currently involved in writing up and analyzing our results, but we have, to date, found evidence to suggest that, in the stimuli containing opaque polygons at least, the presence of depth in the stimuli does indeed improve performance by speeding the time needed to make decisions [20]. We are currently engaged in analyzing the results for the remaining experiments.

Finally, in terms of potential routes that could be taken to improve our existing approach, we believe that the main route to improving the approach would be focused on decreasing the time required to generate the stimuli. As noted above, we did take steps at various stages with a view to making the stimulus generation process more rapid, but the core issue is that the computation and manipulation of overlap in many thousands of complex images is a process that is, however approached, likely to be a slow one. However, since, for each of our experiments, the stimuli only needed to be generated once, the speed of the generation process did not adversely affect our ability to progress as planned with the data collection.

REFERENCES

- [1] T. Menneer, N. Donnelly, H. J. Godwin, and K. R. Cave, "High or low target prevalence increases the dual-target cost in visual search.," *J. Exp. Psychol.*, vol. 16, no. 2, pp. 133–144, 2010.
- [2] C. F. Nodine and L. Kundel, "Using eye movements to study visual search and to improve tumor detection," *RadioGraphics*, vol. 7, no. 6, pp. 1241–1250, 1987.
- [3] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.
- [4] H. J. Godwin, T. Menneer, K. R. Cave, and N. Donnelly, "Dual-target search for high and low prevalence X-ray threat targets.," *Vis. cogn.*, vol. 18, no. 10, pp. 1439–1493, 2010.
- [5] J. M. Wolfe, T. S. Horowitz, and N. M. Kenner, "Rare items often missed in visual searches," *Nature*, vol. 435, no. May, pp. 6–7, 2005.
- [6] M. P. Eckstein, "Visual search : A retrospective," *J. Vis.*, vol. 11, no. 5:14, pp. 1–36, 2011.
- [7] M. B. Neider and G. J. Zelinsky, "Cutting through the clutter : Searching for targets in evolving complex scenes," *J. Vis.*, vol. 11, no. 7, pp. 1–16, 2011.

- [8] M. R. Beck, M. C. Lohrenz, and J. G. Trafton, "Measuring search efficiency in complex visual search tasks: global and local clutter.," *J. Exp. Psychol. Appl.*, vol. 16, no. 3, pp. 238–50, Sep. 2010.
- [9] M. J. Bravo and H. Farid, "Recognizing and segmenting objects in clutter.," *Vision Res.*, vol. 44, no. 4, pp. 385–96, Feb. 2004.
- [10] J. P. McIntire, P. R. Havig, and E. E. Geiselman, "Stereoscopic 3D displays and human performance: A comprehensive review," *Displays*, vol. 35, no. 1, pp. 18–26, Jan. 2014.
- [11] K. Nakayama, S. Shimojo, and G. Silverman, "Stereoscopic depth: its relation to image segmentation, grouping, and the recognition of occluded objects," *Perception*, vol. 18, no. 1, pp. 55–68, 1989.
- [12] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends Cogn. Sci.*, vol. 4, no. 1, pp. 6–14, 2000.
- [13] H. L. Kundel and C. F. Nodine, "Interpreting chest radiographs without visual search.," *Radiology*, vol. 116, pp. 527–532, 1975.
- [14] H. J. Godwin, T. Menneer, C. A. Riggs, K. R. Cave, and N. Donnelly, "Perceptual Failures in the Selection and Identification of Low-Prevalence Targets in Relative Prevalence Visual Search," *Atten. Percept. Psychophys.*, vol. 77, no. 1, pp. 150–9, Sep. 2015.
- [15] M. S. Cain, S. H. Adamo, and S. R. Mitroff, "A taxonomy of errors in multiple-target visual search," *Vis. cogn.*, vol. 21, no. 7, pp. 899–921, Aug. 2013.
- [16] J. D. Schwark, J. MacDonald, J. Sandry, and I. Dolgov, "Prevalence-based decisions undermine visual search," *Vis. cogn.*, vol. 21, no. 5, pp. 541–568, May 2013.
- [17] M. J. Stroud, T. Menneer, K. R. Cave, and N. Donnelly, "Using the dual-target cost to explore the nature of search target representations.," *J. Exp. Psychol. Hum. Percept. Perform.*, vol. 38, no. 1, pp. 113–122, Feb. 2012.
- [18] H. J. Godwin, M. C. Hout, and T. Menneer, "Visual similarity is stronger than semantic similarity in guiding visual search for numbers.," *Psychon. Bull. Rev.*, vol. 21, no. 3, pp. 689–95, Jun. 2014.
- [19] B. Powell, "Using the LockBits method to access image data," 2003. [Online]. Available: <http://bobpowell.net/lockingbits.aspx>.
- [20] H. J. Godwin, T. Menneer, S. P. Liversedge, K. R. Cave, N. S. Holliman, and N. Donnelly, "Searching for overlapping objects in depth: Depth speeds search, but does not improve response accuracy.," in *Vision Sciences Society Annual Conference*, 2014.