

Active Text Drawing Styles for Outdoor Augmented Reality: A User-Based Study and Design Implications

Joseph L. Gabbard¹

Center for Human-Computer Interaction
Virginia Tech

J. Edward Swan II²

Computer Science & Engineering
Mississippi State University

Deborah Hix³

Center for Human-Computer Interaction
Virginia Tech

Si-Jung Kim⁴

Industrial Systems Engineering
Virginia Tech

Greg Fitch⁵

Industrial Systems Engineering
Virginia Tech

ABSTRACT

A challenge in presenting augmenting information in outdoor augmented reality (AR) settings lies in the broad range of uncontrollable environmental conditions that may be present, specifically large-scale fluctuations in natural lighting and wide variations in likely backgrounds or objects in the scene. In this paper, we present a active AR testbed that samples the user's field of view, and collects outdoor illuminance values at the participant's position. The main contribution presented herein is a user-based study (conducted using the testbed) that examined the effects on user performance of four outdoor background textures, four text colors, three text drawing styles, and two text drawing style algorithms for a text identification task using an optical, see-through AR system. We report significant effects for all these variables, and discuss design guidelines and ideas for future work.

CR Categories: H.5 [Information Interfaces and Presentation]: H.5.1: Multimedia Information Systems — Artificial, Augmented, and Virtual Realities; H.5.2: User Interfaces — Ergonomics, Evaluation / Methodology, Screen Design, Style Guides

Keywords: Outdoor Augmented Reality, Optical See-Through Display, Text Drawing Styles, Text Legibility, Empirical Study

1 INTRODUCTION

Presenting legible augmenting information in the outdoors is problematic, due mostly to uncontrollable environmental conditions such as large-scale fluctuations in natural lighting and the various types of backgrounds on which the augmenting information is overlaid. There are often cases where the color and/or brightness of a real-world background visually and perceptually conflicts with the color and/or contrast of graphical user interface (GUI) elements such as text, resulting in poor or nearly-impossible legibility. This issue is particularly true when using optical see-through display hardware.

Several recent studies in AR have begun to experimentally confirm that which was anecdotally known amongst outdoor AR practitioners, but not yet documented — namely, that text legibility is significantly affected by environmental conditions, such as color and texture of the background environment as well as natural illuminance at both the user's and background's position [1; 2; 3; 4; 5].

One strategy to mitigate this problem is for visual AR representations to actively adapt, in real-time, to varying conditions

of the outdoor environment. Following this premise, we created a working testbed to investigate interactions among real-world backgrounds, outdoor lighting, and visual perception of augmenting text. This testbed senses the condition of the environment using a real-time video camera and lightmeter. Based on these inputs, we apply *active algorithms* to GUI text strings, which alter their visual presentation and create greater contrast between the text and the real-world backgrounds, ultimately supporting better legibility and thus user performance. This concept easily generalizes beyond text strings to general GUI elements.

This paper presents a direct follow-on study to our user-based study presented at VR 2005 [2]. Since that time, we have evolved our testbed to the point where we can conduct outdoor studies using real-world backgrounds (as opposed to static posters used in the prior study) and any number of active algorithms.

In our previous study [2; 1], we altered the color of the text itself (under active drawing conditions) to increase contrast between the text and the real-world background. A problem with this approach is that the rendered text color can potentially be very different from the GUI designer's *intended* text color. Since color is widely used to encode semantics (e.g., in military systems blue is used to indicate friendly entities while red is used to indicate enemy entities), we are interested in researching active text drawing techniques that maintain the intended text color of GUI elements while employing real-time sensors in the environment to *visually enhance* the GUI elements to achieve greater legibility. This can be done, for example, by applying a lightweight outline of the text, whose color is *actively* determined to optimize contrast, and thus, legibility.

The focus of the work reported here is studying the effect of environmental conditions on AR text legibility, with a motivation of designing active text drawing styles that are optimal for dynamic environmental conditions. This paper describes work related to the study, our concept of visually active AR user interfaces, our visually active AR testbed (updated since our previous study [2]), and a new user-based study conducted using the updated testbed. We also present results of the user-based study, including a general discussion, and resulting design implications.

2 RELATED WORK

Much of the HCI work that has examined user performance on text legibility tasks has occurred in static settings (e.g., 2D desktop or application settings), where text color and background color do not necessarily change in real-time, and more often than not, can be defined a priori by user interface designers. More recently, work in both the 2D realm as well as in the VR and AR fields has examined methods for optimizing text legibility. Some of the methods studied employ real-time, or *active*,

¹ jgabbard@vt.edu (corresponding email)

² swan@cse.msstate.edu

³ hix@vt.edu

⁴ hikim@vt.edu

⁵ gfitch@vt.edu

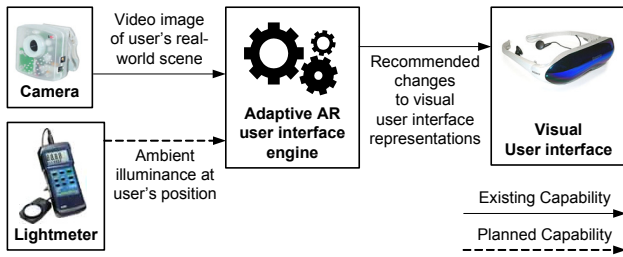


Figure 1. Conceptual drawing of our visually active AR user interface testbed components.

algorithms to increase legibility, while others rely on perceptual design principals.

One of the more common (and important) aspects of AR text legibility that has been examined is that of label placement within an AR scene. These techniques seek to place labels so that (1) labels are associated with object(s) being labeled, while (2) optimizing legibility by reducing clutter and/or overlapping of labels [4; 6]. These techniques can also be considered “active” in the sense that they use information about the real-world scene and make real-time adjustments (via placement algorithms) to the user interface to support improved user performance.

In [1] and [2] we presented results of an experiment that examined the effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor AR. Our work provided clear empirical evidence that user performance on a text legibility task is significantly affected by background texture, text drawing style, and text color. We also showed that the real-world background may affect the amount of ambient illuminance at the user’s position and that the combination of this illuminance and text drawing style ultimately affects user performance.

Leykin and Tuceryan [3] present an approach to automatically determine if overlaid AR text will be readable or unreadable, given dynamic and widely varying textured-background conditions. Their approach employed a real-time classifier that used text features, as well as texture features of the background image, to determine the legibility of overlaid text. They conducted a series of experiments in which participants categorized overlaid text as “readable” vs. “unreadable”, and used their experimental results to train the classification system.

A few studies have produced methods for optimizing transparent text overlaid onto 2D GUI backgrounds — a perceptual usage scenario that is similar to that of optical see-through AR. For example, Paley [7] describes techniques such as the use of outline and color variations to increase legibility of overlaid text; in this paper we also report a text drawing style that uses both character outlining and alternate color schemes for the outline. Harrison and Vicente [8] describe a similar technique used to overlay transparent text (such as drop down menus) onto 2D GUI backgrounds. They present an “anti-interference” font, which uses an outline technique similar to that presented herein. They also describe an empirical evaluation of the effect of varying transparency levels, the visual interference produced by different types of background content, and the performance of anti-interference fonts on text menu selection tasks.

3 VISUALLY ACTIVE AR USER INTERFACES

As mentioned, our general approach to a visually active AR user interface employs real-time sensors (e.g., video camera and/or illuminance meter) to capture information about a user’s visual

Table 1. Summary of variables studied in experiment.

Independent Variables

participant	24	counterbalanced
outdoor background texture (Figure 4)	4	brick, building, sidewalk, sky
text color	4	white, red, green, cyan
text drawing style (Figure 5)	4	none, billboard, drop shadow, outline
text drawing style algorithm	2	maximum HSV complement, maximum brightness contrast
repetition	3	1, 2, 3

Dependent Variables

response time	in milliseconds
error	0 (correct), 1, 2, 3 (incorrect)

field of view to optimize text (or other graphics’) legibility. The intent of such a system is to maintain a highly-usable, flexible user interface given constantly dynamic changes in lighting and background occurring in outdoor usage contexts. A simple example of an active change would be to increase the intensity of all user interface graphics under sunny or bright environmental conditions, and to automatically dim those graphics under nighttime conditions. A slightly more advanced example, which we utilized in our user-based study, uses this information in real time to determine a legible color for an augmenting text label given the current background color (e.g., light sky or dark green foliage). The components of our visually active AR user interface testbed are presented conceptually in Figure 1.

Assuming an AR system that employs sufficiently accurate tracking, and given the geometry of a camera’s lens, it is possible to know where the user’s head is looking. Eye-trackers could even indicate the user’s specific point of regard. Cameras could then sample the entire scene or, alternatively, using a zoom function, sample a part of the scene (e.g., an object or area of interest) to obtain information specific to the user task or simply specific to the direction of a user’s gaze.

A suite of image processing tools, algorithms, and techniques can be used to further digest the scene, including, for example, feature identification and recognition. Once a scene is divided into features (e.g., sky, trees, grass, etc.), the active AR user interface can perform detailed application-specific operations on the feature region to compute appropriate changes to user interface augmentations.

4 THE EMPIRICAL USER-BASED STUDY

We conducted a study that examined the effects on user performance of outdoor background textures, text colors, text drawing styles and text drawing style algorithms for a text identification task. We captured user error and user response time. Table 1 summarizes the variables we examined.

4.1 Our Visually Active AR Testbed

Our recent instantiation of a visually active AR user interface serves as a testbed for empirically studying different text drawing styles and active text drawing algorithms under a wide range of outdoor background and illuminance conditions. Figure 2 shows our testbed, which employs a real-time video camera to capture a user’s visual field of view and to specifically sample the portion of the real-world background on which a specific

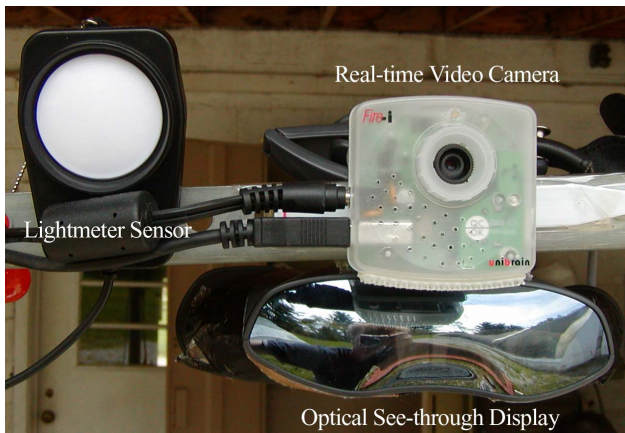


Figure 2. AR display, video camera and lightmeter components of our visually active AR testbed.

user interface element (e.g., text) is overlaid. It also employs a real-time lightmeter (connected via RS232) to provide real-time natural illuminance information to the active system. The user study reported in this paper only actively uses the camera information; the testbed recorded lightmeter information but did not use it to drive the active algorithms. We anticipate developing algorithms that are actively driven by the lightmeter in the future.

As shown in Figure 2, the AR display, camera and lightmeter sensor are mounted on a rig, which in turn is mounted on a tripod (not shown in the figure). Participants sit in an adjustable-height chair so that head positions are consistent across all participants. At this time, our testbed does not use a motion tracking system. For this experiment, we fixed the participants' field-of-view on different backgrounds by repositioning the rig between background conditions. We used previously captured camera images of backgrounds to assist in the positioning procedure and to ensure that each participant's FOV is the same for each background.

Our testbed uses the text's screen location and font characteristics to compute a screen-aligned bounding box for each text string. It then computes the average color of this bounding box, and uses this color to drive the active text drawing algorithms – which in turn determine a text drawing style color. For example, if using a billboard drawing style (see Figure 5), the active text drawing algorithm uses the sampled background color as an input to determine what color to draw the billboard. The specific text drawing styles and text drawing style algorithms are discussed in more detail below.

Our testbed was implemented as a component of the BARS system [9], and uses an optical see-through display, a real-time video camera, a lightmeter, and a mobile laptop computer equipped with a 3D graphics card. The optical see-through display was a Sony Glasstron LDI-100B biocular optical see-through display, with SVGA resolution and a 28° horizontal field of view in each eye. We used a UniBrain Fire-i firewire camera (with settings of YUV 4:2:2 format, 640 X 480 resolution, 30Hz, and automatic gain control and exposure timing). The lightmeter is an Extech 407026 Heavy Duty Light Meter with RS232 interface to measure illuminance at the user's position. Our laptop system (and image generator) was a Pentium M 1.7 GHz computer with 2 gigabytes of RAM and an NVidia GeForce4 4200 Go graphics card generating monoscopic images, running under Windows 2000. We used this same com-



Figure 3. Our experimental task required participants to identify the pair of identical letters in the upper block (e.g., "Vv"), and respond by pressing the numeric key that corresponds to the number of times that letter appears in the lower block (e.g., "2"). Note that this image is a screen capture (via camera) of the participants' field of view and overlaid text, and is not an exact representation of what participant's viewed through the AR display.

puter to collect user data. Figure 2 shows the HMD, camera, and lightmeter components.

4.2 Task and Experimental Setup

We designed a task that abstracted the kind of short reading tasks, such as reading labels, which are prevalent in many proposed AR applications. For this study, we purposefully designed the experimental task to be a low-level visual identification task. That is, we were not concerned with participants' semantic interpretation of the data, but simply whether or not they could quickly and accurately read information. Moreover, the experimental task was designed to force participants to carefully discern a series of random letters, so that task performance was based strictly on legibility. The task was a relatively low-level cognitive task consisting of visual perception of characters, scanning, recognition, memory, decision-making, and motor response.

As shown in Figure 3, participants viewed random letters arranged in two different blocks. The upper block consisted of three different strings of alternating upper and lower case letters, while the lower block consisted of three strings of upper case letters. The participant was first instructed to locate a *target letter* from the upper block; this was a pair of identical letters, one of which was upper case and the other lower case (e.g., "Vv" in Figure 3). Placement of the target letter pair in the upper block was randomized, which forced participants to carefully scan through the block. We considered several other visual cues such as underlining, larger font size, and bold text for designating the target letter; however, we realized that this would result in a "pop-out" phenomenon wherein the participant would locate the target without scanning the distracting letters.

We used the restricted alphabet "C, K, M, O, P, S, U, V, W, X, Z" to minimize variations in task time due to the varying difficulty associated with identifying two identical letters whose upper and lower case appearance may or may not be similar. A post-hoc analysis showed an effect size of $d = .07$ error for letter, which is small when compared to the other effect sizes reported in this paper.



Figure 4. We used four real-world outdoor background textures for the study. Shown above are (clockwise starting in upper left): brick, building, sky, and sidewalk. Stimulus text strings (both upper and lower blocks) were completely contained within the background of interest (as shown in Figure 3). The images represent participants' field of view when looking through the display.

After locating the target letter, the participant was then instructed to look at the lower block and count the number of times the target letter appeared in the lower block. Placement of the target letters in the lower block was randomized. Participants were instructed that the target letter would appear 1, 2, or 3 times. The participant responded by pressing the “1”, “2”, or “3” key to indicate the number of times the target letter appeared in the lower block. In addition, participants were instructed to press the “0” key if they found the text completely illegible.

To minimize carryover effects of fatigue, a rest break was also provided every 28 trials; participants were instructed to close their eyes and relax. The length of the rest break was determined by each participant. After each rest break, the next task was presented to the participant in a similar manner. The entire experiment consisted of 336 trials for each participant.

We wanted to conduct the study under outdoor illuminance conditions, because while indoor illuminance varies by ~3 orders of magnitude, outdoor illuminance varies by ~8 orders of magnitude [10]. However, we could not conduct the study in direct sunlight, because graphics on the Glasstron AR display become almost completely invisible. We also needed to protect the display and other equipment from outdoor weather conditions. We addressed these issues by conducting our study in a covered breezeway overlooking an open area. Since this location required participants to face south (i.e., towards the sun as it moves across the sky), we positioned the participant at the edge of the breezeway, so that their heads (and thus the display) were shaded from the sun, but their vertical field of view was not limited by the breezeway's roof structure. We ran the experiment between April 6th and May 10th 2006, in Blacksburg Virginia, during which time the sun's elevation varied between 23° and 68° above the horizon.

We conducted experiments at 10am, 1pm, and 3pm, and only on days that met our pre-determined natural illuminance lighting requirements (between 2000 and 20,000 lux). Using the lightmeter displayed in Figure 2, we measured the amount of ambient illuminance at the participant's position every trial. Our goals were to quantify the effect of varying ambient illumination on task performance, and to ensure that ambient illuminance fell into our established range. However, our current finding is that between-subjects illumination variation, which represents differ-

	None	Billboard	Drop Shadow	Outline
Brick	KzPxMc PwMxCp MoWcVv	KzPxMc PwMxCp MoWcVv	WzSpOw ZoSpSx ZkOwVv	KvWoZk XvXzUc SsCxMz
Building	KzPxMc PwMxCp MoWcVv	KzPxMc PwMxCp MoWcVv	WzSpOw ZoSpSx ZkOwVv	KvWoZk XvXzUc SsCxMz
Sidewalk	KzPxMc PwMxCp MoWcVv	KzPxMc PwMxCp MoWcVv	WzSpOw ZoSpSx ZkOwVv	KvWoZk XvXzUc SsCxMz
Sky	KzPxMc PwMxCp MoWcVv	KzPxMc PwMxCp MoWcVv	WzSpOw ZoSpSx ZkOwVv	KvWoZk XvXzUc SsCxMz

Figure 5. We used four text drawing styles: none, billboard, drop shadow and outline (shown on the four outdoor background textures). Note that the thumbnails shown above were sub-sampled from the participant's complete field of view.

ences in the weather and time of day, was much larger than the variation between different levels of experimental variables. Therefore, we do not report any effects of illuminance in this paper.

4.3 Independent Variables

Outdoor Background Texture: We chose four outdoor background textures to be representative of commonly-found objects in urban settings: *brick*, *building*, *sidewalk*, and *sky*. Note that three of these backgrounds (all but *building*) were used in our previous study [2; 1], but at that time were presented to the participant as large posters showing a high-resolution photograph of each background texture. In this new study, we used actual real-world backgrounds, as shown in Figure 4 (these images represent the participant's entire field of view when looking through the AR display). Stimulus strings were positioned so that they were completely contained within each background (Figure 3).

We kept the brick and sidewalk backgrounds covered when not in use, so that their condition remained constant throughout the study. The sky background varied depending upon cloud cover, haze, etc., and in some (rare) cases would vary widely as cumulus clouds wandered by. We considered including a grass background, but were concerned that the color and condition of the grass would vary during the months of April and May, moving from a dormant green-brown color to a bright green color.

Text Color: We used four text colors commonly used in computer-based systems: *white*, *red*, *green*, and *cyan*. We chose white because it is often used in AR to create labels and because it is the brightest color presentable on an optical see-through display. Our choice of red and green was based on the physiological fact that cones in the human eye are most sensitive to certain shades of red and green [11; 12]. These two text colors were also used in our first study. We chose cyan to represent the color blue. We chose not to use a “true” blue (0, 0, 255 in RGB color space), because it is a dark color and is not easily visible in optical see-through displays.

Text Drawing Style: We chose four text drawing styles (Figure 5): *none*, *billboard*, *drop shadow*, and *outline*. These are based on previous research in typography, color theory, and human-computer interaction text design. *None* means that text is drawn “as is”, without any surrounding drawing style. We included the *billboard* style because it is commonly used in AR applications and in other fields where text annotations are overlaid onto photographs or video images; arguably it is one of the de-factor drawing styles used for AR labels. We used billboard in our previous study [2]. We included *drop shadow* because it is commonly used in print and television media to offset text from backgrounds. And, we included *outline* as a variant on drop shadow that is visually more salient yet imposes only a slightly larger visual footprint. Also, the outline style is similar to the “anti-interference” font described by Harrison and Vicente [8]. Another motivation for choosing these drawing styles was to compare text drawing styles with small visual footprints (drop shadow, outline) to one with a large visual footprint (billboard).

Text Drawing Style Algorithm: We used two active algorithms to determine the color of the text drawing style: *maximum HSV complement*, and *maximum brightness contrast*. These were the best active algorithms from our previous study [2]. As discussed above, the input to these algorithms is the average color of the screen-aligned bounding box of the augmenting text (Figure 3). We designed the *maximum HSV complement* algorithm with the following goals: retain the notion of employing color complements, account for the fact that optical see-through AR displays cannot present the color black, and use the HSV color model [13] so we could easily and independently modify saturation. We designed the *maximum brightness contrast* algorithm to maximize the perceived brightness contrast between text drawing styles and outdoor background textures. This algorithm is based on MacIntyre’s maximum luminance contrast technique [14; 15]. These algorithms are described in detail in [2].

Repetition: We presented each combination of levels of independent variables three times.

4.4 Dependent Variables

Also as summarized in Table 1, we collected values for two dependent variables: response time and error. For each trial, our custom software recorded the participant’s four-alternative forced choice (0, 1, 2, or 3) and the participant’s response time. For each trial, we also recorded the ambient illuminance at that moment in time, the average background color sampled by the camera, as well as the color computed by the text drawing style algorithm. This additional information will allow us to calculate (post-hoc) pair-wise contrast values between text color, text drawing style color, and background color; however, at this time we have not yet completed these analyses. In this paper we report an analysis of the error and response time data.

4.5 Experimental Design and Participants

We used a factorial nesting of independent variables for our experimental design, which varied in the order they are listed in Table 1, from slowest (participant) to fastest (repetition). We collected a total of 24 (participant) \times 4 (background) \times 4 (color) \times [1 (drawing style = *none*) + [3 (remaining drawing styles) \times 2 (algorithm)]] \times 3 (repetition) = 8064 response times and errors. We counterbalanced presentation of independent variables using a combination of Latin Squares and random permutations. Each

participant saw all levels of each independent variable, so all variables were within-participant.

Twenty-four participants participated, twelve males and twelve females, ranging in age from 18 to 34. All participants volunteered and received no monetary compensation; some received a small amount of course credit for participating in the experiment. We screened all participants, via self-reporting, for color blindness and visual acuity. Participants did not appear to have any difficulty learning the task or completing the experiment.

4.6 Hypotheses

Prior to conducting the study, we made the following hypotheses:

- (1) The brick background will result in slower and less accurate task performance because it is the most visually complex.
- (2) The building background will result in faster and more accurate task performance because the building wall faced north and was therefore shaded at all times.
- (3) Because the white text is brightest, it will result in the fastest and most accurate task performance.
- (4) The billboard text drawing style will result in the fastest and most accurate task performance since it has the largest visual footprint, and thus best separates the text from the outdoor background texture.
- (5) Since the text drawing styles are designed to create visual contrast between the text and the background, the presence of active text drawing styles will result in faster and more accurate task performance than the *none* condition.

5 RESULTS

For error analysis we created an error metric e that ranged from 0 to 3:

$$e = \begin{cases} |c - p| & \text{if } p \in \{1, 2, 3\} \\ 3 & \text{if } p = 0 \end{cases},$$

where $e = 0$ to 2 was computed by taking the absolute value of c , the correct number of target letters, minus p , the participant’s response. $e = 0$ indicates a correct response, and $e = 1$ or 2 indicates that the participant miscounted the number of target letters in the stimulus string. $e = 3$ is used for trials where users pressed the “0” key (indicating they found the text illegible). Our rationale is that not being able to read the text at all warranted the largest error score, since it gave the participant no opportunity to perform the task. Our error analysis revealed a 14.9% error rate across all participants and all 8064 trials. This error rate is composed of 5.2% for $e = 1$, 0.5% for $e = 2$, and 9.2% for $e = 3$.

For response time analysis, we removed all repetitions of all trials when participants indicated that the text was illegible ($e = 3$), since these times were not representative of tasks performed under readable conditions. This resulted in 7324 response time trials (~91% of 8054 trials). Overall, we observed a mean response time of 5780.6 milliseconds (msec), with a standard deviation of 3147.0 msec.

We used repeated-measures Analysis of Variance (ANOVA) to analyze the error and response time data. For this ANOVA, the *participant* variable was considered a random variable while all other independent variables were fixed. Because our design was unbalanced (the text drawing style *none* had no drawing

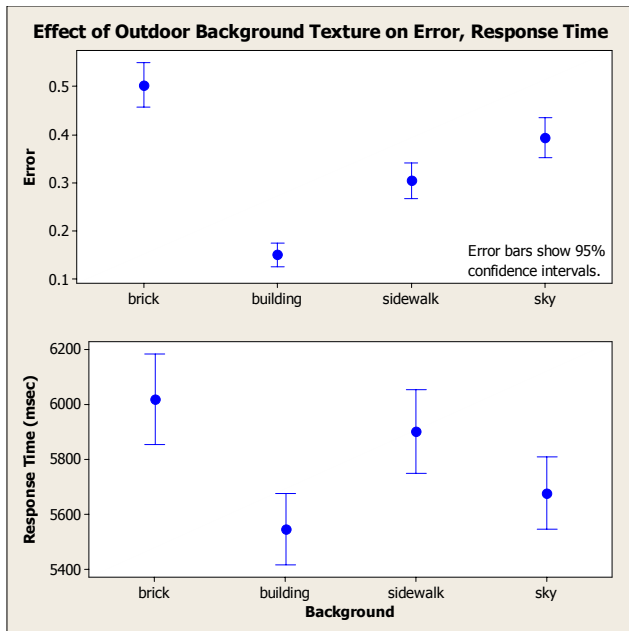


Figure 6. Effect of background on error ($N = 8064$) and response time ($N = 7324$). In this and future graphs, N is the number of trials over which the results are calculated.

style algorithm), and because we removed trials for the response time analysis, we could not run a full factorial ANOVA. Instead, we separately tested all main effects and two-way interactions of the independent variables. When deciding which results to report, in addition to considering the p value, the standard measure of *effect significance*, we considered d , a simple measure of *effect size*. $d = \max - \min$, where \max is the largest mean and \min the smallest mean of each result. d is given in units of either *error* or *msec*.

5.1 Main Effects

Figure 6 shows the main effect of background on both error ($F(3, 69) = 23.03, p < .000, d = .353$ error) and response time ($F(3, 69) = 2.56, p = .062, d = 471$ msec). Participants performed most accurately on the building background, and made the most errors on the brick background. A similar trend was found for response time. These findings are consistent with hypothesis 1 and hypothesis 2.

There was little difference in error under sidewalk and sky conditions ($d = .089$ error), and similar results for response time ($d = 225$ msec). We observed a relatively large amount of illuminance reflecting off the brick background, and we hypothesize that this illuminance, as well as the complexity of the brick background texture, explain why brick resulted in poor performance. Similarly, we hypothesize that the lack of reflected sunlight and homogeneity of the building background account for the lower errors and faster response times.

Contrary to hypothesis 3, there was no main effect of text color on either error ($F(3, 69) = 2.34, p = .081, d = .075$ error) or response time ($F(3, 69) = 1.81, p = .154, d = 253$ msec). However, when we examined the subset of trials where drawing style = *none*, we found significant main effects of both error ($F(3, 69) = 5.16, p = .003, d = .313$ error) and response time ($F(3, 69) = 8.49, p < .000, d = 1062$ msec). As shown in right-hand column of Figure 8, participants performed less accurately and more slowly with red text, while performance with the other text colors (cyan, green, white) was equivalent ($d = .063$ error, $d = 166$

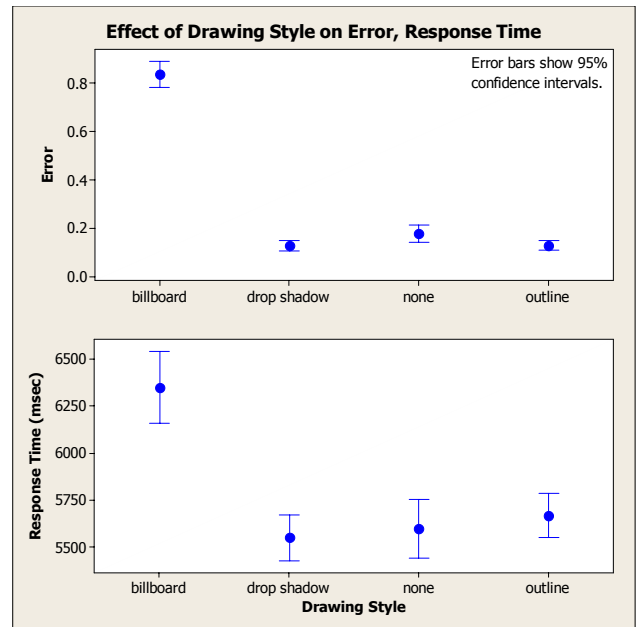


Figure 7. Effect of text drawing style on error ($N = 8064$) and response time ($N = 7324$).

msec). This result may be due to the luminance limitations of the Glasstron display, resulting in less luminance contrast for red text as compared to cyan, green, and white text. This result is consistent with the finding in our previous study that red performed poorly [2, 1], and provides further design guidance that pure red text should be avoided in see-through AR displays used in outdoor settings. Furthermore, together with the lack of an effect of text color over all of the data, these findings suggest that our active drawing styles may enable more consistent participant performance across all text colors, which would allow AR user interface designers to use text color to encode interface elements.

Figure 7 shows the main effect of text drawing style on both error ($F(3, 69) = 152, p < .000, d = .711$ error) and response time ($F(3, 69) = 11.6, p < .000, d = 797$ msec). In both cases, participants performed less accurately and more slowly with the billboard text drawing style, while performance across the other text drawing styles (drop shadow, outline, none) was equivalent ($d = .051$ error, $d = 118$ msec). These findings are contrary to hypothesis 4. As explained in Section 4.3, our active text drawing style algorithms use the average background color as an input to determine a drawing style color that creates a good contrast between the drawing style and the background. Furthermore, the drawing style is a graphical element that *surrounds* the text, either as a billboard, drop shadow, or outline. A limitation of this approach is that it does not consider the contrast between the text color and the surrounding graphic. Both drop shadow and outline follow the shape of the text letters, while billboard has a large visual footprint (Figure 5). Therefore, it is likely that in the billboard case, the contrast between text color and the billboard color is *more* important than the contrast between billboard color and background color, while the opposite is likely true for the drop shadow and outline styles. These findings are consistent with this hypothesis.

Additionally, we propose that there are (at least) two contrast ratios of interest when designing active text drawing styles for outdoor AR: that between the text and the drawing style, and that between the text drawing style and the background. Both

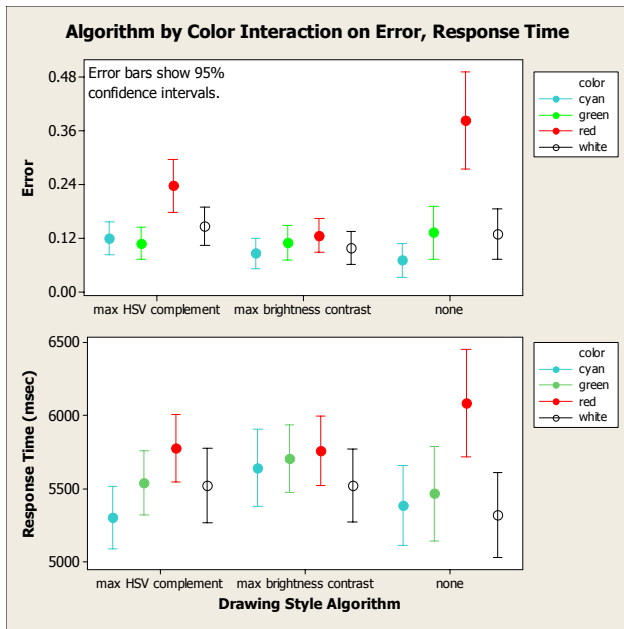


Figure 8. Effect of drawing style algorithm by text color on error ($N = 5760$) and response time ($N = 5615$) for the trials where drawing style \neq billboard. The right-hand column shows the effect of text color on error ($N = 1152$) and response time ($N = 1109$) for the trials where drawing style = none.

the size of the text drawing style and whether or not it follows the shape of the letters likely determines which of these two contrast ratios is more important.

Since our billboard style was not compatible with our background-based drawing style algorithms, and because it exhibits a large effect size, we removed the billboard drawing style and performed additional analysis on the remaining data set.

Figure 8 shows that drawing style interacted with text color using this subset of data, on both error ($F(6, 138) = 2.96, p = .009, d = .313$ error) and response time ($F(6, 138) = 2.95, p = .010, d = 1062$ msec). The effect size of text color was the smallest with the maximum brightness contrast algorithm ($d = .040$ error, $d = 221$ msec), followed by the maximum HSV complement algorithm ($d = .129$ error, $d = 589$ msec), and followed by text drawn with no drawing style and hence no algorithm ($d = .313$ error, $d = 1062$ msec). Figure 9 shows that drawing style algorithm also had a small but significant main effect on error ($F(2, 46) = 3.46, p = 0.04, d = .074$ error). Participants were most accurate when reading text drawn with the maximum brightness contrast algorithm, followed by the maximum HSV complement algorithm, and followed text drawn with no algorithm. Tukey HSD post-hoc comparisons [16] verify that maximum brightness contrast is significantly different than the other algorithms, while *maximum HSV complement* and *none* do not significantly differ.

It is important to note that the maximum brightness contrast drawing style algorithm does not exist by itself, but instead is manifested within the drawing style. More importantly, the algorithm resulted in less errors for the sky and background conditions (see Figure 9, bottom), suggesting that there are some backgrounds where the addition of active drawing styles can provide a real benefit (although we did not find an algorithm by background interaction for this data set ($F(6, 138) = 1.21, p = .304, d = .234$ error)). Similar to the findings for text color, the effect size of background was the smallest with the maximum brightness contrast algorithm ($d = .089$ error), followed by the

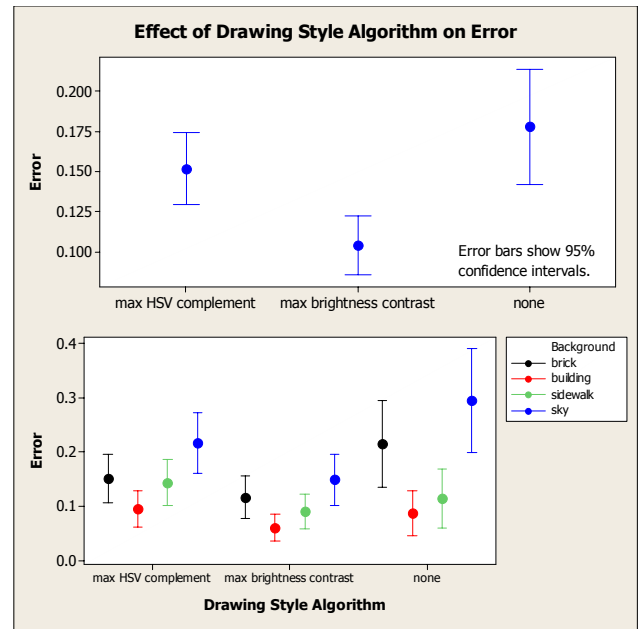


Figure 9. Effect of text drawing style algorithm on error ($N = 5760$) for the trials where drawing style \neq billboard.

maximum HSV complement algorithm ($d = .122$ error), and followed by text drawn with no drawing style and hence no algorithm ($d = .208$ error).

Taken together, these results show that when drawing style \neq billboard, the maximum brightness contrast algorithm resulted in the overall best error performance (Figure 9, top), as well as the least variation in performance over color for error and response time (Figure 8), and the least variation over background for error (Figure 9, bottom). More generally, these results suggest that the presence of active text drawing styles can both decrease errors and reduce variability over the absence of any text drawing styles (i.e., the *none* condition) — especially those active drawing styles that employ the maximum brightness contrast drawing style algorithm.

6 DISCUSSION AND RESULTING IMPLICATIONS FOR DESIGN

We've successfully implemented an active AR user interface testbed that is capable of demonstrating the utility of active text drawing styles. Our empirical findings suggest that the presence of active drawing styles effects user performance for text legibility, and that as we continue to research and design active drawing styles, we should take into account at least two kinds of contrast ratios: the contrast ratio between the text and the drawing style, as well as the contrast ratio between the drawing style and the background. Although not explored here, there are likely times where a third contrast ratio (text color to background) is of interest – an indeed, in active systems may indicate whether or not an intervening drawing style is even needed at all!

A finding consistent with our previous study [1], is clear empirical evidence that user performance on a visual search task, which we believe is representative of a wide variety of imagined and realized AR applications, is significantly affected by background texture (Figure 6), text drawing style (Figure 7), text color (Figure 8), and active drawing style algorithm (Figures 8 and 9). These findings suggest that more research is needed to understand how text and background color interact,

and how to best design active systems to mitigate performance differences.

One limitation of our study was that we did not use any “control” colors for our three text drawing styles. That is, every time a text drawing style was drawn, it used an active color determined via the drawing style algorithm. Including a control drawing style color (e.g., white) would have allowed us to verify the benefit of drawing styles independent of whether or not the styles were active or not. This limitation did not preclude us however, from comparing the drop shadow to the outline drawing style.

In terms of design implications, our error analyses suggest the color red should not be used without an accompanying text drawing style, especially when the AR display is not designed for outdoor use (and thus, does not provide bright graphics). Moreover, when using a large footprint text drawing style (e.g., billboard), designers should use text-based active drawing style algorithms that strive to create good contrast between the text color and the color of the surrounding graphic. When using text drawing styles that have a small visual footprint (e.g., outline or drop shadow), use background-based active drawing style algorithms that strive to create good contrast between the text drawing style color and the outdoor background texture.

7 FUTURE WORK

We intend to perform further and more detailed analysis on the data from this study, to better understand the perceptual underpinnings of our visual search task under the varied conditions. Specifically, we plan to closely examine the pair-wise contrast ratios between text color, text drawing style color, and outdoor background textures and the relative importance of each pair-wise contrast ratio for our given text drawing styles (including the none drawing style).

Moreover, we plan to conduct a study that systematically varies the contrast ratio between text color and text drawing style color, so that we can better understand what is the minimum contrast needed for effective task performance on text legibility tasks. We also will be further analyzing the contrast ratios between text color and background color for the trials where no text drawing style was present; again to better understand what amount of contrast is needed to improve text legibility. Once we better understand these contrast thresholds, we will use this knowledge to inform more sophisticated drawing style algorithms and to determine appropriate text drawing styles under varying environmental conditions.

We plan to normalize the collected illuminance data to allow us to perform additional analysis, and provide more evidence regarding the effects of illuminance on text legibility. And, we plan to perform additional “meta-level” analysis of our experimental task, to understand for example, if placement of the target letter, or shape of target letter confounds results in any way. This will help us design better experimental tasks for future empirical work.

Lastly, we plan to upgrade some testbed components, specifically the AR optical see-through display and the real-time camera.

ACKNOWLEDGEMENTS

Our work on active drawing styles for outdoor AR has been supported by the Office of Naval Research (ONR) and the Naval Research Lab, under Program Managers Dr. Behzad Kamgar-Parsi and Dr. Larry Rosenblum. We’d also like to thank Mark

Livingston of the Naval Research Lab, as well as Dr. Steve Feiner of Columbia University for loaning us the Glasstron AR display, without which this work could not have happened. We’d like to thank Dr. Woodrow Winchester III for supporting this research as part of an ISE6614 class project. Lastly, we thank Mr. Phil Keating, Dr. Llyod Hipkins, and Mr. Claude Kenley for supporting our research and allowing us to conduct the user-based study described herein at their “Virginia Tech Weed Science – Turfgrass Noncrop Aquatics” facilities.

REFERENCES

- [1] Gabbard, J.L., Swan II, J.E., Hix, D. (2006). “The Effects of Text Drawing Styles, Background Textures, and Natural Lighting on Text Legibility in Outdoor Augmented Reality”, Invited paper to Presence: Teleoperators & Virtual Environments, Spring 2006, Vol. 15, No. 1, Pages 16-32.
- [2] Gabbard, J.L., Swan II, J.E., Hix, D., Schulman, R.S., Lucas, J., & Gupta, D. (2005). “An Empirical User-Based Study of Text Drawing Styles and Outdoor Background Textures for Augmented Reality”. In Proceedings of IEEE Virtual Reality 2005, pp. 11-18.
- [3] Leykin, A., Tuceryan, M. “Automatic determination of text readability over textured backgrounds for augmented reality systems”, In Proceedings of the 3rd IEEE and ACM Symposium on Mixed and Augmented Reality (ISMAR 2004), pages 224 – 230, 2004.
- [4] Azuma, R., & Furmanski, C. “Evaluating label placement for augmented reality view management”. In Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003), pages 66–75, 2003.
- [5] Piekarski, W. & Thomas, B. (2002). “ARQuake: The Outdoor Augmented Reality Gaming System”, Communications of the ACM, Volume 45, Issue 1, pp. 36–38.
- [6] Bell, B., Feiner, S., & Höllerer, T. “View management for virtual and augmented reality”. In Proceedings of the 14th annual ACM symposium on User interface software and technology, pages 101–110. ACM Press, 2001.
- [7] Paley, W.B. “Designing better transparent overlays by applying illustration techniques and vision findings”. In Proceedings of the 16th annual ACM symposium on User interface software and technology, Supplement, pages 57–58. ACM Press, 2003.
- [8] Harrison B.L. & Vicente K.J. (1996). “An Experimental Evaluation of Transparent Menu Usage”, In Proceedings CHI ‘96, pp. 391–398.
- [9] Livingston, M.A., Rosenblum, L., Julier, S.J., Brown, D., Baillot, Y., Swan II, J.E., Gabbard, J.L., & Hix, D. (2002). “An Augmented Reality System for Military Operations in Urban Terrain”, In Proceedings of the Interservice / Industry Training, Simulation, & Education Conference (I/ITSEC ‘02), Orlando, FL, December 2–5.
- [10] Halsted, C.P. (1993). “Brightness, Luminance and Confusion”, Information display, March.
- [11] Hecht, E. (1987). Optics (2nd edition), Addison Wesley.
- [12] Williamson, S.J. & Cummins, H.Z. (1983). Light and Color in Nature and Art, Wiley and Sons, NY.
- [13] Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F., & Phillips, R.L. (1993). Introduction to Computer Graphics (2nd edition), Reading, MA, Addison-Wesley.
- [14] MacIntyre, B. (1991). A Constraint-Based Approach To Dynamic Colour Management For Windowing Interfaces, Master’s thesis, University of Waterloo, Available as Department of Computer Science Research Report CS-91-55.
- [15] MacIntyre, B. & Cowan, W. (1992). “A Practical Approach to Calculating Luminance Contrast on a CRT”, ACM Transactions on Graphics, Volume 11, Issue 4, pp. 336–347.
- [16] Howell, D.C. (2002). Statistical Methods for Psychology (5th edition), Duxbury.