Environment Adaptive AR Label Visualization

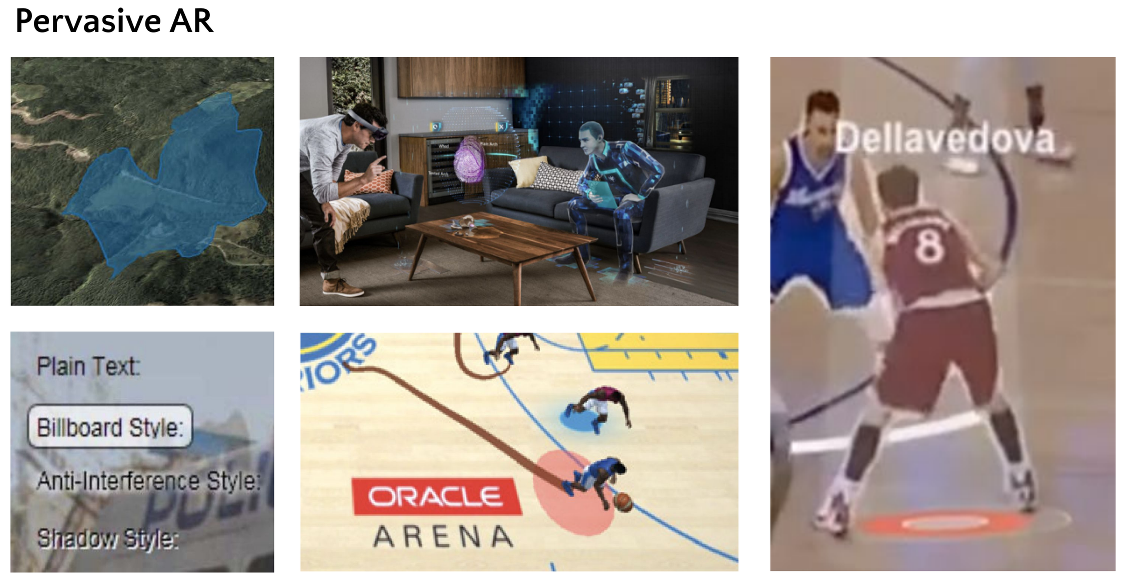
Ashley Kwon, Yuanbo Li, Eva Schiller, Brayden Goldstein-Gelb, James Tompkin
Brown University Computer Science

Abstract

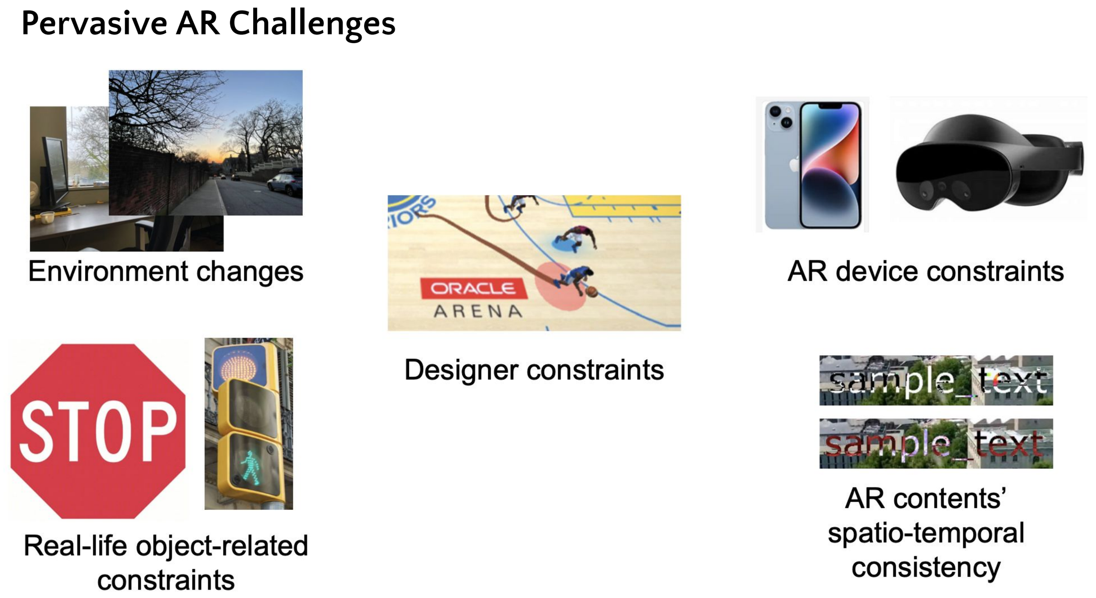
The recent increase in immersive content and diverse environments in which people use AR drew researchers' attention to different modes of content visualization in AR. Pervasive AR is an emerging field that addresses challenges in dynamically changing AR contents based on users' displays and characteristics of surrounding real-world environments [1]. Previous publications discussed how appearances of labels such as colors and brightness influence the labels' visibility in AR systems [2, 3]. However, most of the models that these studies introduced were tested only with 2D videos and photos and did not have functionalities that allowed labels to change colors based on backgrounds with a wide range of colors. In our project, we develop a style and color changing model for AR labels, which dynamically generates labels' appearances based on the background. Focusing on applications in dynamic environments that mimic see-through AR (VST), in which an AR device overlays contents on a video of users' surroundings, the model uses per-pixel color assignments of labels. The model aims to enhance labels' visibility in various environments.

Introduction

Pervasive AR



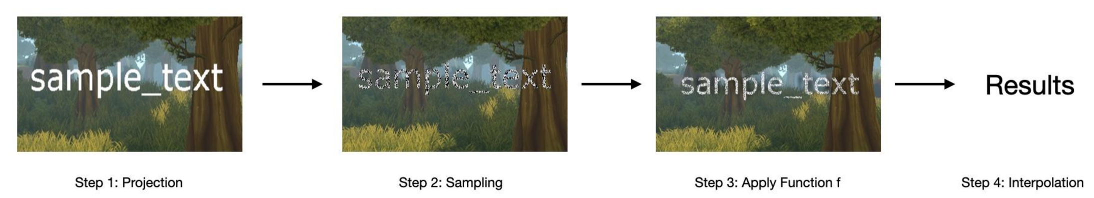
Pervasive AR Challenges



Goal
Design a system that dynamically changes AR labels' appearances based on background to facilitate user visibility

Real-Time Label Style and Color Generation

Our method takes as input a 2D image, which has been captured and rendered by the camera, and a 2D label text that needs to be superimposed on the image. Our approach enables real-time text rendering and dynamically adjusts the appearance of the label based on the background. Our method consists of four steps, which are illustrated below.



Projection
In the first step, we project the 2D label text onto the 3D background image. We denote the set of background image pixels as $\mathcal{B}$ and the set of label pixels as $\mathcal{L}$. We define a surjective function $g : \mathcal{B} \rightarrow \mathcal{L}$ that maps each pixel in the background image to the corresponding pixel in the label text. This projection ensures that all label pixels are covered by the projection.
Sampling
In the second step, we select a subset of λn, $\lambda \in (0,1)$ pixels from the total of n pixels in the labels and a user-defined parameter, λ.
Color Adjustments
In the third step, we apply the color adjustment function f to the selected subset of λn pixels in the label, in order to meet two constraints. The first constraint is to maximize the contrast between the label pixels and the background. The second constraint is to minimize the contrast among the label pixels. To achieve this, we have implemented various algorithms and our method involves applying layers of filters.
Interpolation
In the final step, we employ thin plate spline interpolation to obtain the remaining $(1 - \lambda)n$ pixel values.

Label Style Assignment Algorithms

Shadow
The shadow is created using a duplicate of the label. The duplicate is scaled, set to black, and a gaussian blur of desired kernel size and standard deviation is applied. The opacity is also adjusted to ensure that the shadow is not too noticeable while maintaining visual separation from the background.

Outline
The outline is generated by applying a Sobel filter to the label. This detects the edges of the label. The color of the outline at a given pixel is set to either black or white depending on which has the most contrast with the sample pixel.

Label Color Assignment Algorithms

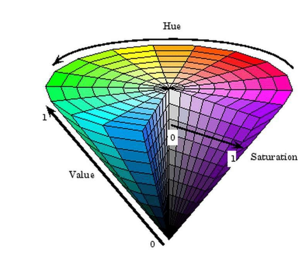
We built various functions that increase contrast with the sampled pixel:

Palette-based contrast
For a fixed palette of colors, P, we take the furthest color in the palette from our sample color.

$$\underset{c \in P}{\operatorname{argmax}} ||b_i - c||$$

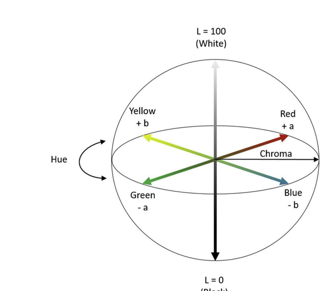
HSV-based contrast
Choose values of H and V of that maximize distance to the sample color in the HSV color space

$$H' = H + 180^\circ$$
$$S' = S$$
$$V' = \begin{cases} 1 & \text{if } V \leq 0.5 \\ 0 & \text{if } V > 0.5 \end{cases}$$

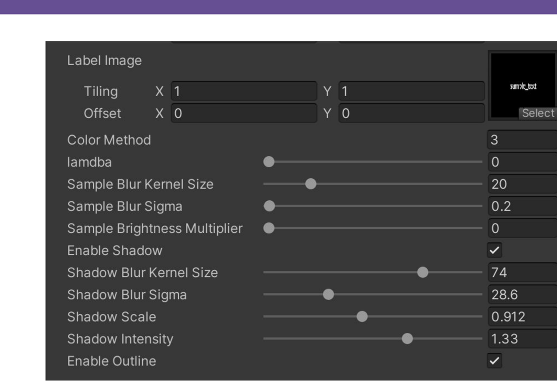


CIELAB-based contrast
The sample color is converted to CIELAB, a perceptually uniform color space. The values of a* and b*, which represent hue, are set to constant values of a desired color and L, which represents perceptual lightness, is set to a value that increases its difference from the sample color.

$$L' = \begin{cases} 100 - L & \text{if } 0 < L \leq 25 \text{ or } 75 < L \leq 100 \\ 125 - L & \text{if } 25 < L \leq 50 \\ 75 - L & \text{if } 50 < L \leq 75 \end{cases}$$
$$a^{*\prime} = a^*_{desired}$$
$$b^{*\prime} = b^*_{desired}$$



Style and Color Assignment Interface



Color assignment function can be selected live using **Color Method**. **Lambda** modifies the extent of blending between sampled pixels and the label color assigned by our chosen color method.

We can adjust the sample prior to color assignment using **Sample Blur Kernel Size, Sample Blur Sigma,** and **Sample Brightness Multiplier**.

Shadowing can be toggled with **Enable Shadow**. We can adjust properties of the shadow using **Shadow Blur Kernel Size**, **Shadow Blur Sigma** and **Shadow Scale**. **Shadow Intensity** modifies the opacity of the shadow. Outline can be toggled with **Enable Outline.**

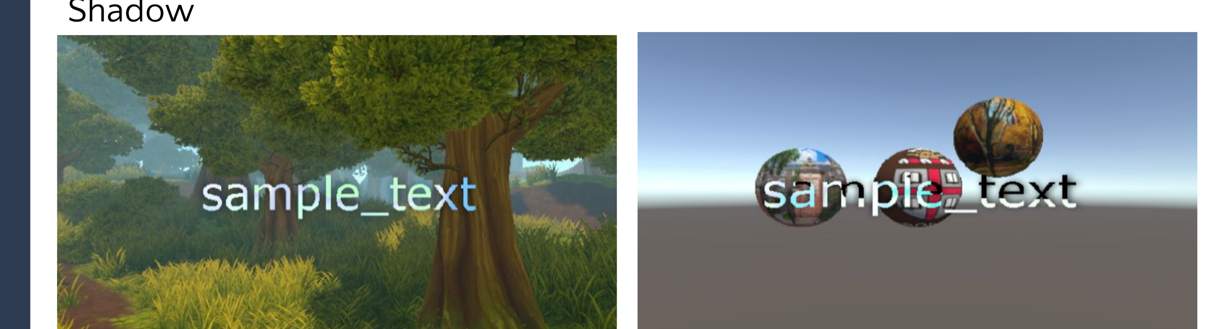
Results

Palette-based contrast

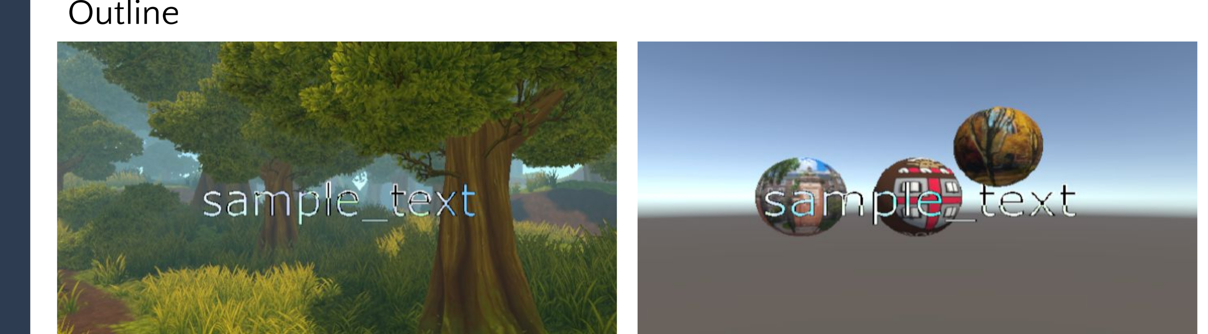
HSV-based contrast

CIELAB-based contrast

Shadow



Outline



References

[1] J. Grubert et al. "Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality". In *IEEE Transactions on Visualization and Computer Graphics* 23.6 (2017), pp. 1706–1724.
[2] J. Gabbard, J. Swan, D. Hix, "The Effect of Text Drawing Styles, Background Textures, and Natural Lighting on Text Legibility in Outdoor Augmented Reality". In: *Presence* 15.1 (2006), pp. 16–32.
[3] J. Jankowski et al. "Integrating Text with Video and 3D Graphics: The Effects of Text Drawing Styles on Text Readability". In *Proceedings of the SIGCHI Conference on Human Factor Systems* (2010).