VoxAR: Adaptive Visualization of Volume Rendered Objects in Optical See-Through Augmented Reality

Saeed Boorboor, Matthew S. Castellana, Yoonsang Kim, Chen Zhu-Tian, Johanna Beyer, Hanspeter Pfister, *Fellow, IEEE*, and Arie E. Kaufman, *Fellow, IEEE*

***Abstract*—We present VoxAR, a method to facilitate an effective visualization of volume-rendered objects in optical see-through head-mounted displays (OST-HMDs). The potential of augmented reality (AR) to integrate digital information into the physical world provides new opportunities for visualizing and interpreting scientific data. However, a limitation of OST-HMD technology is that rendered pixels of a virtual object can interfere with the colors of the real-world, making it challenging to perceive the augmented virtual information accurately. We address this challenge in a two-step approach. First, VoxAR determines an appropriate placement of the volume-rendered object in the real-world scene by evaluating a set of spatial and environmental objectives, managed as user-selected preferences and pre-defined constraints. We achieve a real-time solution by implementing the objectives using a GPU shader language. Next, VoxAR adjusts the colors of the input transfer function (TF) based on the real-world placement region. Specifically, we introduce a novel optimization method that adjusts the TF colors such that the resulting volume-rendered pixels are discernible against the background and the TF maintains the perceptual mapping between the colors and data intensity values. Finally, we present an assessment of our approach through objective evaluations and subjective user studies.**

***Index Terms*—Adaptive Visualization, Situated Visualization, Augmented Reality, Volume Rendering.**

I. Introduction

THE transformative ability of augmented reality (AR) to fuse the digital world of bits with the physical world of atoms has provided new opportunities to visualize 3D spatial scientific data. Over the decades, there have been significant advances in methods for presenting virtual information to users. However, in contrast to virtual reality (VR) and mixed reality (MR), optical see-through (OST) AR has only been sparsely adopted for scientific visualization [1]. This can be attributed to a fundamental challenge in OST-AR: virtual content augmented onto the user's field-of-view (FoV) blends with the physical environment. As such, it is possible to inaccurately perceive the rendered pixels, which is a critical limitation in scientific visualization. The full potential of AR is realized when data is effectively visualized with respect to its referent in the physical space. Towards this goal, methods have been developed for the intuitive placement [2], [3], visibility enhancement [4]–[7], and color correction [8], [9] of virtual objects rendered in OST head-mounted displays (HMDs), albeit as separate objectives [10].

A fundamental utility of volume rendering visualization is exploring and interpreting volume data. Typically, this is achieved using transfer functions (TFs) that map the intrinsic values of the data to a spectrum of optical properties, such as color and opacity. In terms of visual perception, most existing AR techniques are not adequately designed to address the challenges of visualizing volume-rendered objects. This motivated us to design VoxAR – a method for augmenting volume-rendered objects in OST-HMDs that adapts to real-world surroundings. Specifically, VoxAR adopts a two-fold approach: first, it determines an optimal position for displaying a virtual object in the user's FoV, based on user-defined preferences, and second, it adjusts the colors of the TF to distinguish the rendered volume from the background.

The placement of a virtual object in the scene can significantly impact data understanding and decision-making [11]. Unlike a controlled desktop setting, spatial locations of virtual AR visualizations cannot be pre-calculated for the real-world, and must be determined in-situ. Existing MR frameworks and toolkits [3], [12] adaptively place virtual objects in the scene by solving a set of rules and user-defined semantic preferences, such as distance between the virtual object and its physical referent, its distance from the FoV, and surface magnetism. To improve data visual perception in OST-AR, we introduce an additional measure for the perceptual color difference between the real-world backdrop and TF colors. VoxAR is designed using GPU shaders that evaluate all candidate 3D spatial locations in the FoV in parallel and solve an optimal location that best satisfies a composite of user-defined semantic preferences, minimizing environmental limitations.

Following placement, the colors of the input TF may require further adjustment for discernibility with respect to the background. Existing solutions to alleviate color blending can be broadly categorized into contrast enhancement [4], [5], [9] and re-colorization [13]. However, unlike the nature of the virtual objects addressed in most existing techniques, where colors broadly indicate the presence of an attribute, volume rendering enables reasoning and understanding about the data attributes through a perceptual mapping of defined TF colors and the visual output. In VoxAR, we introduce a novel approach that optimally shifts the input TF to reduce color blending of the volume-rendered object with the real-world background while maintaining the visual characteristics in mapping data attributes. Rather than post-processing, we perform the TF enhancement prior to the volume rendering

Boorboor, Castellana, Kim, and Kaufman are with the Department of Computer Science, Stony Brook University. Zhu-Tian, Beyer, and Pfister are with John A. Paulson School of Engineering and Applied Sciences, Harvard University.

step. Specifically, we have designed an objective function that shifts the input TF in the CIE $L^*a^*b^*$ (CIELAB) space such that it satisfies a set of constraints designed to (1) maximize the visual color difference between the TF spectrum and the background colors, (2) maintain color properties similar to the input TF, and (3) ensure a valid $L^*a^*b^*$ to RGB-space transformation. Given the 3D search space, we use CMA-ES [14] as a solver to find the TF color intervals with minimum objective cost.

We have developed VoxAR as an end-to-end system in Unity3D game engine for Microsoft Hololens2 OST-HMD, and demonstrate our results using TF presets from widely used volume rendering applications. To evaluate the effectiveness of our approach, we conduct user studies and show that VoxAR significantly enhances a user's ability to perceive and analyze volume-rendering visualizations in AR. In summary, we define our contributions as follows:

- an end-to-end OST-AR framework for the placement and adaptive visualization of volume-rendered objects,
- a real-time implementation for solving an optimal placement based on user preferences while minimizing color overlap with the real-world surroundings,
- a novel method to adjust the TF color based on background colors while preserving the perceptual mapping between volume data attributes and the input TF.

II. Related Work

Sec. II-A presents works on adaptive object placement and its formalized constraint systems, and Sec. II-B discusses techniques to improve color perception in AR. We have observed that (1) placement and color constraints have not been effectively explored as a coupled problem, and (2) they do not satisfactorily address the challenges of volume visualization.

A. Visualization and Object Placement

In context-aware MR, virtual experience is dynamically adapted to context-specific information, based on the application goals. For example, contextual information can be derived from depth information: Google's DepthLab [15] utilizes depth information to create believable interactions with the environment. Microsoft's FLARE [16] analyzes a scene to generate location-specific AR layouts based on detected geometry and surfaces. Location-based data can also provide context. Systems for situated visualizations, such as SemanticAdapt [17] and RagRug [18] modify virtual objects based on semantic associations with real-world objects. VoxAR utilizes a context-aware approach, generating scene-specific candidate locations for placement based on color and depth information, and evaluating spatial and environmental objectives managed as user-selected preferences and pre-defined constraints.

Many AR authoring systems allow specifying objectives for automatically determining the behavior of AR objects. Microsoft's Mixed Reality Toolkit [19] uses solvers that compute the position and orientation of AR objects based on preferences, such as, surface magnetism, constant view size, etc. Unity MARS [2] uses Reasoning APIs to collect information about the scene and extracts objects into a higher-level database of special semantic information, stored as *traits*. Users can construct and assign *conditions* to AR objects, which are used to evaluate real-world placements based on how well the corresponding traits satisfy the conditions. Evangelista et al. have developed AUIT [3] that allows users to customize the placement process: hook into existing abstracted data sources, write their own objectives, determine solvers and when to trigger them, and how to transition between changing states. Each of these tools can handle goals relating to transform-related properties of AR objects. In VoxAR, we extend the scope of the placement objectives to address the visual properties of the object and scene. Our approach provides a novel implementation technique for incorporating such image-based objectives alongside spatial objectives, taking advantage of specialized shaders for increased efficiency.

B. Visualization Color Enhancement

Solutions to alleviate color-blending can be divided into hardware [20], [21] and software-based methods. Given the scope of our work, we describe here software-based techniques. For hardware-based solutions, we refer the reader to a survey by Itoh et al. [22]. Existing works that address color blending in OST-HMDs [23] can be broadly categorized as color correction and visibility improvement solutions. Color correction involves sensing background colors and subsequently subtracting them from the colors of the virtual content [8]. However, such compensation typically results in a decrease in brightness. For improved color reproduction, Hincapi et al. [9] have developed SmartColor, a real-time algorithm that performs per-pixel background subtraction in the CIELAB space. The work focuses primarily on text visibility and achieves real-time by discretizing the color space, which does not adequately capture the full TF spectrum. Fukiage et al. [7] have introduced a framework that enables consistent visibility of semi-transparent objects by measuring the background color and texture. To accurately align the real-world background environment to the OST display rendering, Langlotz et al. [24] have introduced a hardware prototype that allows pixel-precise radiometric compensation. A major limitation in applying these approaches to volume rendering is that they aim to improve per-pixel color blending based on background luminance and do not consider TF-based color-component channels. A recent work closest to our goals is that of Zhang et al. [4]. In this work, the authors present a constraint-based system to preserve the contrast between virtual objects and the background and maintain consistency with the original displayed color. However, their approach is threshold-intensive. That is to say, for effective results, the suitable hue thresholds need to be adjusted, especially for varying backgrounds and TF colors.

While there have been works to improve volume rendering quality in video see-through AR [25], [26], to the best of our knowledge, no work sufficiently addresses the challenge of visibility improvement for volume visualization in OST-AR. Most approaches address blending as a post-processing problem, whereas we have designed VoxAR to solve the TF

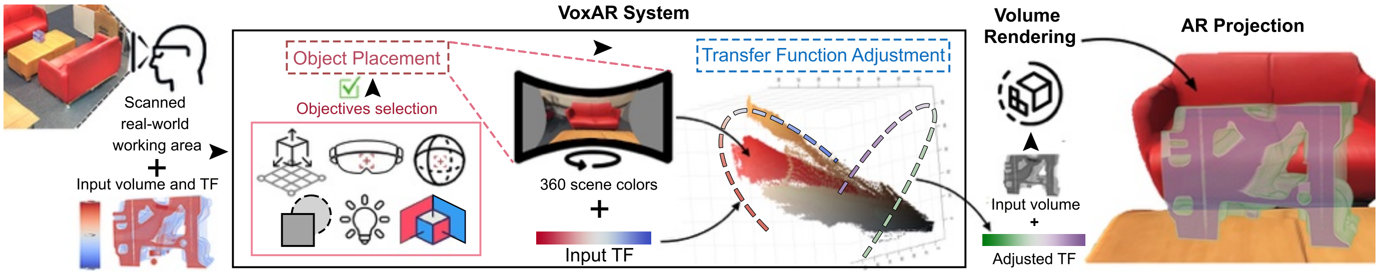

Fig. 1. An illustration of the VoxAR pipeline. For a volume to be rendered using an input TF function, in a user's real-world FoV, VoxAR first determines an optimal placement. This is evaluated based on an objectives selection set, which the users opt for from our formulated list. The placement scene is then used to adjust the input TF such that the resultant volume rendering facilitates an effective visualizing experience.

color optimization prior to rendering. Moreover, for techniques that solve color enhancement in the CIELAB color space, we noticed that most works assume a valid projection of their solution to RGB, whereas VoxAR ensures a valid and perceptually meaningful conversion as part of its constraint.

III. VoxAR Design and Workflow

We introduce a two-step method that precedes the volume rendering and augmentation pipeline. First, by evaluating spatial and environmental objectives, managed as user-selected preferences and pre-defined constraints, VoxAR finds an appropriate position for placing the virtual object in the real-world (Sec. IV). Next, it adjusts the input TF colors such that the resultant volume-rendered object is discernable from its background. Importantly, this adjustment attempts to best preserve the perceptual mapping between data attributes and the colors assigned in the input TF (Sec. V).

One key utility of AR applications is interacting with virtual objects while navigating the real-world. Constantly adapting positions and colors during visualization and analysis can risk introducing inconsistencies in data perception. Thus, to support changing FoV, following initial placement, VoxAR performs the TF optimization based on the background colors surrounding the provided position. Subsequently, it continues to evaluate the placement objective score with an additional constraint of maintaining color discernibility of the adjusted TF. If the objective score falls below a threshold, an alternative optimal position is suggested to the user. To this end, the user must first scan a working area, allowing the system to generate a 3D scene model and evaluate its semantics. VoxAR pipeline overview is shown in Fig. 1.

IV. VoxAR Visualization Object Placement

For a given scene instance, VoxAR solves the placement of the virtual object in the real-world by (1) determining sets of candidate locations, (2) analyzing the candidate locations based on the user-selected objectives, and (3) placing the object at the location which *best* satisfies the objectives.

A. Determining Candidate Locations for Placement

The search space for placing a virtual object in the physical space can be constrained to a finite set of visually distinguishable, semantically meaningful, and environmentally favorable locations in 3D space, and evaluated based on user preferences and constraints. VoxAR uses two distinct categories of candidate locations: *surface magnetism* [27], [28] and *discretized 3D*. For placement in discretized 3D, we evaluate regular, discrete locations within a bounding volume of the working area using a fixed orientation. However, for surface magnetism, we additionally consider the object orientation and the scene surface normals.

To achieve a real-time system, we introduce *placement maps*, a texture-based data structure that stores the spatial information for each scene pixel. A placement map comprises three textures with the following attributes:

- **Validity map:** a binary score for pixel evaluation.
- **Position map:** for a valid pixel, its corresponding 3D position, mapped to the texture (r, g, b) tuple.
- **Rotation map:** for a valid pixel, the quaternion of its normal, mapped to the texture (r, g, b, a) tuple.

This representation allows parallel evaluation of multiple positions for multiple objectives on a GPU.

Using the inverse projection matrix of the AR HMD camera I, the placement maps for surface magnetism and discretized 3D are generated as follows. For surface magnetism, a single placement map PM_{surf} consists of:

- position map $POS_{surf}[x, y] = I * (x, y, depth(x, y))$
- rotation map $ROT_{surf}[x, y] = orient(normal(x, y))$
- validity map $VM_{surf}[x, y] = is\_surface(x, y)$

where *depth* and *normal* are externally calculated using the scanned 3D model of the scene, *orient* returns a quaternion-based orientation from a provided normal, and *is_surface* is a boolean based on surface detection information.

In contrast, discretized 3D creates multiple placement maps from a user-specified bounding volume derived from the camera frustum. Given a discrete set of camera-space values along the camera positive Z axis $\{z \mid z = \text{near}+k\cdot\text{interval}, k \in \mathbb{Z}, \text{near} \leq z \leq \text{far}\}$, where near, far, and interval can be adjusted, and single quaternion Q representing an orientation in 3D space, a set of placement maps $PM_{3D} = \{PM_{z_i} \mid \forall z_i \in z, \exists PM_{z_i}\}$ is generated where:

- position map $POS_{z_i}[x, y] = I * (x, y, z_i)$
- rotation map $ROT_{z_i}[x, y] = (Q.x, Q.y, Q.z, Q.w)$
- validity map $VM_{z_i}[x, y] = 1$

In effect, we create a discrete 3D grid of points via placement maps, with width and height resolution corresponding to the device FoV, and an adjustable depth resolution based on

$near$, far, and $interval$. Higher depth resolutions consider more areas in 3D space at the expense of overall system performance. Although the discretized 3D placement maps come from the same 3D volume, each placement map is evaluated by objectives independently of all others.

B. Defining the Placement Objectives

We have identified and implemented a set of objectives based on our review of adaptive placement-related objectives [2], [3], [19], [27]. The objectives were chosen by how well they contribute to the goal of AR volume visualization, with a primary focus on volume visibility and a secondary focus on semantically meaningful placement. We have additionally designed a new color objective for the perceptual contrast between the TF and real-world background colors.

Below, we describe and formulate the scores for each objective. Unless otherwise specified, the output of the objective is a score $\in [1, 0]$, using:

$$Score(x, y, \lambda) = \begin{cases} 1, & \text{if } \lambda \text{ and } valid \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

O1 **Surface magnetism** associates an object against a horizontal or vertical surface or anywhere in the 3D space. Given the set of placement map types S, where $S = \{$*3D*, *surface*$\}$, the score for each pixel is determined using Eq. 1 where λ is true if the type of the corresponding placement map has been expressed as a user preference.

O2 **Point proximity** allows users to tether a virtual object to a 3D point in the real-world, (x_p, y_p, z_p), with a proximity of maximum distance d_p. The score is calculated from Eq. 1 where $\lambda = \|(x_p, y_p, z_p) - \text{pixel2world}(x, y)\| \leq d_p$. The $\text{pixel2world}(x, y)$ provides a 3D position using the placement map's position map.

O3 **Center of screen projection** projects the object toward the center of a user's FoV. For a device screen space center, (x_s, y_s), and a scaling factor, s, the score map for this constraint is calculated using:

$$Score(x, y, s) = e^{-\alpha}, \quad \alpha = \frac{\|(x, y) - (x_s, y_s)\|}{s} \tag{2}$$

O4 **Color discernibility** is our novel objective that maximizes the visual difference between the TF and real-world scene colors. Given an input TF and real-world scene colors projected in CIELAB, the objective score for each pixel is its minimum perceptual difference from the TF (explained in detail in Sec. V). The per-pixel result is then averaged using a 2D kernel, explained below.

O5 **Visibility** ensures the virtual object is not occluded by scene objects. For a pixel 3D location, its score is determined by performing oriented ray-cube intersection tests using a bounding box positioned at the object-sized 2D kernel. Using Eq. 1, λ is the intersection test result.

O6 **Environmental** avoids the placement of the virtual object against challenging environmental conditions (e.g., light sources that cause OST-HMD projection flushing). For a visible light source, its pixels are set to 0.

Users can define any combination of objectives. For each objective, a weight and a constraint level, categorizing it as *requirement* or *preference*, must be specified. A *required* objective must always be met. That is, if the placement map has a validity value of 0, that pixel will not be considered for placement, regardless of other objectives. We identify objectives **O5** and **O6** as hard constraints since they impact the effectiveness of the visualization and, therefore, are always considered with minimum pre-defined weights.

Some objectives, such as **O2**, evaluate the placement map on a per-pixel basis. Other image-related objectives, such as **O4**, require more complex evaluations since rendered volumes generally take up multiple pixels onscreen. To define a 2D kernel, we have designed a custom shader that, for each pixel, calculates the size of the bounding box of the AR object, centered at the pixel and oriented to the corresponding rotation.

C. Final Placement

Each candidate location in each placement map is evaluated for each objective, and the per-pixel score is stored in a *score map* texture. To determine the locally optimal pixel location within each score map group, score maps *SM* are aggregated using:

$$SM_{agg}[x, y] = \begin{cases} 0 & \text{if } \exists SM_i((SM_i[x, y] = 0) \wedge H_i) \\ \sum_{i=1}^{n} SM_i[x, y] * w_i & \text{otherwise} \end{cases} \tag{3}$$

where SM_{agg} is the aggregated score map, (x, y) is the current pixel location, SM_i is an objective-specific score map, w_i is the corresponding objective weight, and H_i is 1 if the objective for SM_i is classified as *hard* and 0 otherwise.

Finally, the 3D position and rotation of the globally optimal pixel location is queried from the placement map used to augment the virtual object in the scene.

D. Re-evaluating Changing FoV

For changing FoV, continuous re-evaluation is needed to ensure that the user-selected objectives are consistently met. To achieve this, after initial placement, VoxAR examines the current placement per second and calculates an aggregate objective score based on the location the object appears in the current FoV. When an object is outside the updated FoV, objectives **O1**, **O4**, **O5**, and **O6** maintain their most recent scores, **O3** returns 0, and **O2** is re-calculated.

To avoid spontaneous repositioning, a new optimal position is determined once the current objective score falls below a defined threshold. It is important to note that to maintain the perceptual mapping between the TF and data attributes, VoxAR does not re-adjust the TF. Thus, during re-evaluation, **O4**, color discernibility, becomes a *requirement* constraint.

On determining a new position, the user receives a visual suggestion. If the user accepts, the AR object will be moved to the new location and, subsequently, re-evaluated. Contrarily, the suggestion may move, if the location is no longer good, or disappear, if the prior location score significantly improves.

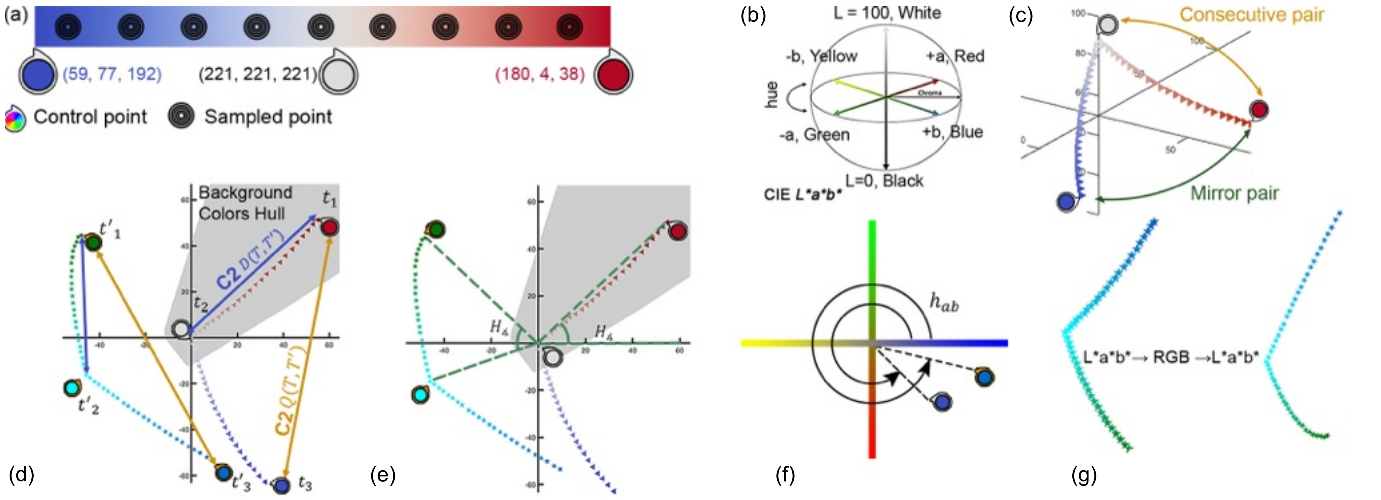

Fig. 2. (a) the linear interpolation of TF colors from its control points. (b) diagram of the CIELAB colorspace. (c) mapping of the TF in (a) from RGB to CIELAB. (d) - (g) illustrations of the TF constraints: (d) intersection of the adjusted control points with the background region (**C1**), and the ΔE calculation between consecutive and mirror pairs (**C2**), (e) hue separation (**C2**), (f) hue measurement (**C3**), and (g) valid projections between color spaces (**C4**).

V. VoxAR Transfer Function Adjustment

In direct volume rendering, TFs classify features within data by mapping data attributes to optical properties, such as color and opacity [29]. For the scope of this work, we consider 1D TFs that map scalar data intensity values to color and opacity. Typically, 1D TFs are defined using a set of control points with assigned color and opacity values, which are then linearly interpolated to construct a complete spectrum (Fig. 2(a)).

To avoid the visual similarity between colors in the real-world and any color along the interpolated TF spectrum, we formulate a constrained optimization that aims to: (1) ensure the colors in the adjusted TF spectrum are perceptually distinguishable from the background and (2) the color control points of the adjusted TF reflect visual characteristics similar to those of the input TF.

A. Defining the Solution Space

To obtain a visually meaningful solution, we solve our optimization in the CIELAB space [30], a device-agnostic 3D space, modeled to represent colors as perceived by the human eye. Specifically, it expresses colors as a measure of perceptual lightness, L^*, *redness*-to-*greenness*, a^*, and *blueness*-to-*yellowness*, b^* (Fig. 2(b)). Based on this representation, the distance between two color values, ΔE_{00} [31], corresponds approximately to the change humans see between colors:

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L}{S_L}\right)^2 + \left(\frac{\Delta C}{S_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2 + R_T \frac{\Delta C}{S_C}\frac{\Delta H}{S_H}} \tag{4}$$

where ΔL, ΔC, and ΔH are the CIELAB lightness, chroma, and hue differences, S_L, S_C, and S_H are the scaling factors, and R_T accounts for the interaction between chroma and hue differences. We will refer to ΔE_{00} as ΔE.

To balance the effect of the size of visual objects size on color appearance [32], Stone et al. [33] have developed a model that provides a minimum scaling factor for $L^*a^*b^*$ that enables effective discernibility of colors more than 50% of the time. In our implementation, we consider a visual angle of $1/3°$ and scale the function interval by 3, as suggested by Gramazio et al. [34].

B. Optimization Constraints

As an initial step, RGB values from the pixels of the real-world scene and TF color control points are projected in the CIELAB space. We have formulated a set of constraints to determine a penalty cost for each adjusted TF candidate, as discussed below. Figs. 2(d)-(g) illustrates the computation of the constraints.

C1 Background discriminability. The primary goal for adjusting the TF is to minimize interference with the real-world. Therefore, for the set of colors in the TF candidate, T', and the background, B, we define this constraint, I, as:

$$I(T,B) = \begin{cases} kE_D, & \text{if } \Delta E(T',B) \leq d \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where E_D is the constraint penalty score and k is a weighted factor, explained below. We choose $d = 11.5$ for the ΔE bound as a scaled *just-noticeable difference* (JND) measure to cater to the possibility of low opacity TF mapping and the semi-transparent projection. Existing works [35], [36] have quantified an empirical benchmark for the minimum perceptual color difference as 2.3, commonly termed JND. Based on our initial experiments and pilot user studies, we empirically determined a scaled factor of JND ×5.

To support changing FoV around the initial placement, a histogram of all colors in the working region is computed. This is achieved by pivoting a virtual 360°camera at the placement position and generating a panoramic scene texture. Naturally, there are many unique colors in the entire scene, and it can become challenging to find visually non-intersecting colors. To effectively reduce the number of background colors, we convert the panorama into superpixels of 10% of the unique colors in the scene texture. Moreover, we noticed that background pixels with a low color distribution or are physically located far from the object make it difficult for the solver to converge to a solution. Thus, we represent $b \in B$ as $(L^*, b^*, a^*, \sum^n (s_n \times d_n))$, where n is the color frequency, s is the superpixel size, and d is the euclidean distance between the physical position corresponding to the center of the superpixel and the object placement position. To reduce the performance overhead of comparing all the background and adjusted TF points, we define a convex hull bounding the background points. Therefore, k in Eq. 5 is the normalized frequency value × of normalized distance value of the vertex closest to the TF point intersecting with the hull.

C2 Perceptual characteristics of the TF control points. To preserve the visual characteristics of the input TF (T), we have identified three attributes to compare in candidate TFs: (1) perceptual color difference, (2) hue separation, and (3) $L^*a^*b^*$ congruence. We formulate this constraint, P, as:

$$P(T,T') = w_d D(T,T') + w_a A(T,T') + w_q Q(T,T') \tag{6}$$

with w_d, w_a, and w_q as weights for each attribute constraint.

The first term, $D(T,T')$, measures the perceptual color difference between control point pairs:

$$D(T,T') = \sum_{i=1}^{n-1} \mid \Delta E(t_i, t_{i+1}) - \Delta E(t'_i, t'_{i+1}) \mid \tag{7}$$

where n is the number of control points in the TFs, $t \in T$, and $t' \in T'$.

The second term, $A(T,T')$ maintains a measure of hue separation between the control points by comparing the angles between consecutive pairs on the a^*b^* plane:

$$\begin{aligned} A(T,T') &= \sum_{i=1}^{n-1} \mid H_{\measuredangle}(t_i, t_{i+1}) - H_{\measuredangle}(t'_i, t'_{i+1}) \mid \\ H_{\measuredangle}(c_1, c_2) &= \cos^{-1}\left(\frac{c_1 \cdot c_2}{\|c1\| \|c2\|}\right) \end{aligned} \tag{8}$$

Finally, we noticed that due to consecutive pair-wise comparisons, in some instances, the solver would optimize the cost by interleaving the shape of the TF curve in such a way that it would satisfy the constraints. Therefore, to preserve the TF global curvature, we additionally check for its congruence by performing mirror comparisons of the control points. We define the third term Q, that checks for congruence, as:

$$\begin{aligned} Q(T,T') &= \sum_{i=1}^{n/2} \mid \Delta E(t_i, t_{n-i+1}) - \Delta E(t'_i, t'_{n-i+1}) \mid \\ &+ \sum_{i=1}^{n/2} \mid H_{\measuredangle}(t_i, t_{n-i+1}) - H_{\measuredangle}(t'_i, t'_{n-i+1}) \mid \end{aligned} \tag{9}$$

C3 Similarity to original color tone. For some applications, it may be important that the adjusted TF retains the hueness of the input TF. Therefore, we formulate this constraint, S, to adjust to a user-defined weight, w_s, as:

$$\begin{aligned} S(T,T') &= \sum_{i}^{n} s(t_n, t'_n), \\ s(c,c') &= \begin{cases} e^{\Delta h_{ab}(c', \lambda_h)}, & \text{if } \Delta h_{ab}(c', \lambda_h) \geq \lambda_h \\ 0, & \text{otherwise} \end{cases} \\ h_{ab}(c) &= \arctan\left(\frac{c_{b^*}}{c_{a^*}}\right) \end{aligned} \tag{10}$$

where $\lambda_h = w_s h_{ab}(t_n)$, h_{ab} is the hueness measured in the a^*b^* space, and Δh_{ab} is the absolute difference.

C4 CIELAB to RGB projection. Due to the difference in 3D gamut sizes, not all CIELAB values have a valid RGB projection. Moreover, not considering gamma correction, projecting the optimization solution from a continuous CIELAB space to a discrete RGB space, may lose perceptual color differentiation on the device. Therefore, for a valid and equally effective adjusted TF in the RGB space, we define V as:

$$V(T'_{rgb}) = w_{pr} \sum_{i=1}^{n} Pr(t'_{rgb,\,i}) + w_{jnd} \sum_{i=1}^{n-1} J\left(t'_{rgb,\,i}, t'_{rgb,\,i+1}\right) \tag{11}$$

$$Pr(c) = \begin{cases} 0, & \text{if } (0,0,0) \leq (r,g,b) \leq (1,1,1) \\ 1, & otherwise \end{cases} \tag{12}$$

$$\begin{aligned} J(c_1, c_2) &= \sum_{i=1}^{k-1} f(\text{lerp}(c_1, c_2, i), \text{lerp}(c_1, c_2, i+1)) \\ f(a,b) &= \begin{cases} 1, & \Delta E(\text{lab2rgb}(a), \text{lab2rgb}(b)) \leq k\,\text{JND} \\ 0, & otherwise \end{cases} \end{aligned} \tag{13}$$

where T'_{rgb} is the set of control points from the candidate T' projected in the RGB color space. Since TFs are a continuous interpolation of control points, we formulate Eq. 13 to uniformly sample RGB values in T'_{rgb}, using a sampling frequency k, and maintain a reasonable JND value along the spectrum when reprojected back to the CIELAB space.

C. Solving the TF Adjustment Final Objective

Conclusively, VoxAR solves for an adjusted TF, TF$_{adj}$ by minimizing the following objective:

$$\begin{aligned} \left[\hat{\mathbf{R}}, \hat{\mathbf{T}}\right] = \text{argmin}_{\mathbf{R},\mathbf{T}}&[I(T',B) + P(T,T') \\ &+ V(T'_{rgb}) + S(T,T')] \end{aligned} \tag{14}$$

$$T' = \left[\mathbf{RT}\right] T \quad , \quad T = \left[t_1, t_2, ..., t_n\right]^\top$$

$$\text{TF}_{adj} = \left[\hat{\mathbf{R}}, \hat{\mathbf{T}}\right] T \tag{15}$$

where $\mathbf{R} \in (\theta_{L^*}, \theta_{a^*}, \theta_{b^*})$ and $\mathbf{T} \in (L^*, a^*, b^*)$ are rotation and translation matrices, respectively. Essentially, we solve for an optimal transformation of the input TF control points in CIELAB, such that the adjusted TF would satisfy the formulated constraints. Given the large search space and possible solutions, we use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14] as the solver for our objective function. CMA-ES is an evolutionary algorithm commonly

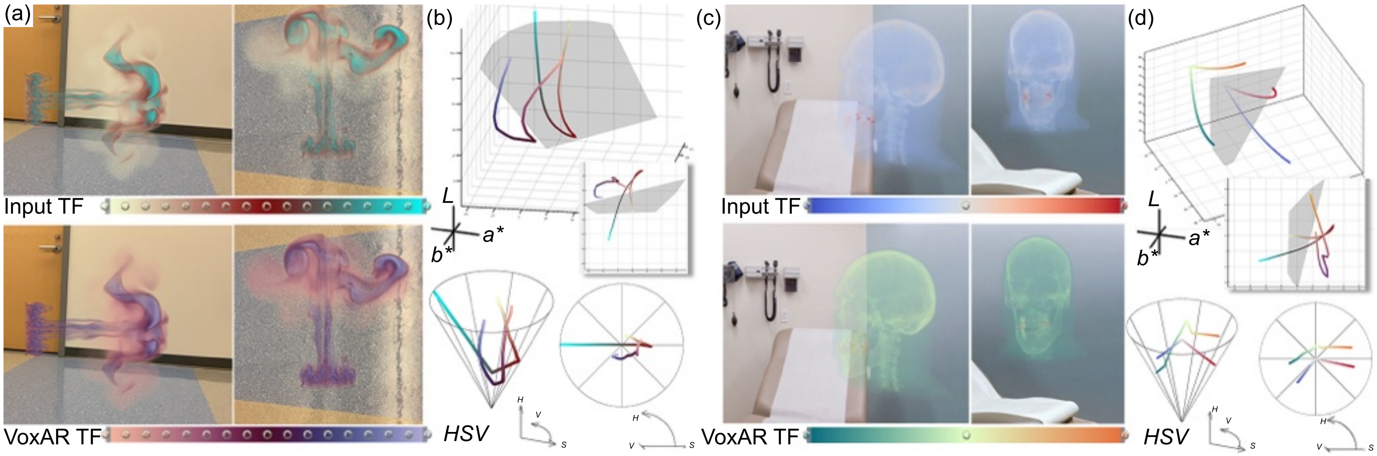

Fig. 3. VoxAR TF adjustment results. (a) and (c) show FoVs of the Jet volume rendered using Haze-Cyan input TF (*top*) and our VoxAR TF *(Bottom)*, and the Skull volume rendered using the Diverging input TF (*top*) and our VoxAR TF *(Bottom)*, respectively. (b) and (d) show plots of the input and VoxAR TF in the CIELAB (*top*) space, with the gray hull representing background colors, and HSV (*bottom*) color space.

used for global optimization of non-linear functions. It is particularly effective in high-dimensional search spaces and can handle noisy and non-convex optimization problems.

At each iteration, CMA-ES generates a population of candidate **R** and **T** according to a multivariate normal distribution with a mean vector and covariance matrix updated based on the history of successful candidate solutions. Specifically, the candidates from a population are used to transform the input TF and are evaluated using the objective function in Eq. 14. The best candidate solutions are then selected to form the next generation, and the mean and covariance matrix of the multivariate normal distribution are updated accordingly to bias the search towards promising regions of the search space. This process is repeated until the maximum number of iterations or a desired level of convergence is met. Thus, CMA-ES uses a combination of random search and adaptation of the search distribution to explore the search space efficiently and converge to an optimal solution.

VI. Implementation

We demonstrate VoxAR on Microsoft HoloLens2, and developed using Unity3D [37] and VTK's holographic remote rendering feature [38] for volume rendering. While the placement objectives are evaluated on the GPU, our TF adjustment is implemented on the CPU. Thus, after receiving a placement result from the HMD, VoxAR solves the adjusted TF on a compute server and passes the result to VTK for volume rendering. Using the Microsoft Mixed Reality Toolkit (MRTK) holographic remoting feature [39], the volume-rendered result is sent to the HMD over a wireless network. Furthermore, the coupled MRTK and VTK system allows users to perform basic volume interactions – in our case, rotation – which is then communicated to VTK for re-rendering and, subsequently, reprojection in AR. VoxAR assumes certain features, such as depth information and spatial mapping, to be obtained using the HMD's API (such as MRTK). Additionally, many AR toolkits are capable of surface detection, classifying specific surfaces as walls, floors, tables, and more.

VII. Results and Evaluation

We now discuss the results of VoxAR, including findings from a user study we conducted to assess the system. Since the results, as seen through an OST-HMD, cannot be shown as images, we use the MRTK additive shader on the HoloLens Mixed Reality Capture to exhibit the visual quality of the rendered semi-transparent pixels. For this work, we have used the following TF presets (with control points):

- Red-White-Blue (Diverging), 3 control points:

- Haze-Cyan, 17 control points:

- Continuous Viridis, 256 control points:

- Continuous Inferno, 256 control points:

For all TF adjustments, we used a large value of $k = 10$ and $w_{pr} = 10$ for **C1** background discernibility and **C4** CIELAB to RGB projection weights, respectively, to avoid invalid solutions that may need manual correction. All other weights, w_d, w_a, w_q, and w_{jnd}, were set to 0.5. We set CMA-ES to run the optimization for a maximum of 20 iterations. We refer the reader to a mixed-reality video capture of our results in the supplementary material.

In Fig. 3, we demonstrate examples of VoxAR used for scientific visualization. Fig. 3(a) *top* shows a gas combustion volume rendered using the Haze-Cyan TF in a hallway. Following placement – using objectives **O3** center screen and **O1** discretized 3D – VoxAR adjusts the TF such that the low-density gas particles, at the lighter end of the input TF, blending with the wall and floor, becomes visually contrasting, as shown in Fig. 3(a) *bottom*. Fig. 3(b) *top* shows the input and VoxAR TFs projected in the CIELAB space. The gray hull represents the real-world colors, captured in 360°centered at the VoxAR placement position. It can be seen here that VoxAR adjusts the TF to avoid intersection with the real-world colors while maintaining the perceptual shape of the input TF. Moreover, Fig. 3(b) *bottom* shows the HSV projection of the input and VoxAR TFs. This example can be extended to situated-AR visualizations, where experts may wish to analyze simulation data connected to its physical referent or location.

Fig. 3(c) *top* and *bottom* demonstrate a head CT volume rendering, using an input diverging TF and the VoxAR adjusted TF, respectively, projected in a clinic examination room, using surface magnetism objective **O1**. Here, we also demonstrate **C3**, hue similarity. Using $\lambda_h = 20$, VoxAR TF adjusts the needed hue amount in the input TF to avoid background color intersection (Fig. 3(d)). This shows that VoxAR can enhance the integration of AR-based medical visualization, specifically in scenarios where medical experts may want to observe and project data in the surroundings or refer to the patient without a blocking video-see-through device. Moreover, the ability of VoxAR TF adjustment to preserve perceptual mapping of colors to data attributes facilitates the sensitive need to visualize medical data as accurately as possible.

For volumes with surface features, it is common practice to design TFs with multi-point transparency (alpha) values to extract nested structures within the data. Fig. 4 compares the visualization of a tooth anatomy – the outer membrane ①, crown ②, and pulp ③ – rendered using an input diverging TF Fig. 4(a) and the VoxAR adjusted TF Fig. 4(b). The alpha values are assigned to visualize the crown and pulp inside the tooth membrane. It can be seen here that the VoxAR TF reduces color blending of the semi-transparent enamel with the background in surface-like shaded rendering.

Furthermore, we compare our technique with the most recent work in AR color enhancement by Zhang et al. [4] (using $\lambda_E = 0.4$, as suggested in the publication). Fig. 5(a) shows a synthetic volume of shapes with data intensities spread uniformly across the data ranges, rendered using the Haze-Cyan TF. While [4] aids in recovering the shapes blended in the background, highlighted using the dotted annotation in (b), their algorithm does not retain the color consistency of the TF, as pointed out using the arrow. This is because [4] performs a pixel-wise operation of the virtual object against its corresponding background color. By designing an algorithm that precedes the volume-rendering step, VoxAR, in contrast, takes a more wholesome approach and determines an optimized color spectrum by evaluating all the background colors simultaneously. The improvement in color contrast and color consistency using VoxAR is shown in Fig. 5(c). To emphasize the importance of a volume-render-specific approach, in Fig. 5(d)-(f), we demonstrate [4] and VoxAR to visualize a neuron volume in a biology lab space. For an input inferno TF Fig. 5(d), it can be seen that the main objective of the enhancement optimization in [4] is to contrast the virtual object pixels from the underlying real-world pixels. In contrast, VoxAR is more meaningful as it does not perform a pixel-wise enhancement but an overall color-to-color enhancement, *preceding* the rendering pipeline. This becomes more significant for changing FoV, where the VoxAR rendered volume will maintain its rendering result.

In Fig. 6, we demonstrate VoxAR placement for a small working region with varying colors, a surface, and a point defined at the scene's center with proximity assigned to cover the area. The evaluation was conducted for a TF overlapping with the background, shown in Fig. 6(a). For objectives **O4** color discernibility and **O1**, VoxAR places the volume on the surface and in front of the green background, away from the overlapping yellow color (Fig. 6(b)). In contrast, for **O2** point proximity and **O1** anywhere in 3D, VoxAR places the volume in front of the blue background (Fig. 6(c)).

Finally, we demonstrate VoxAR placement updates for changing FoV. For a change from the green to blue FoV in Fig. 7(a), (b) shows the initial placement for the green FoV.

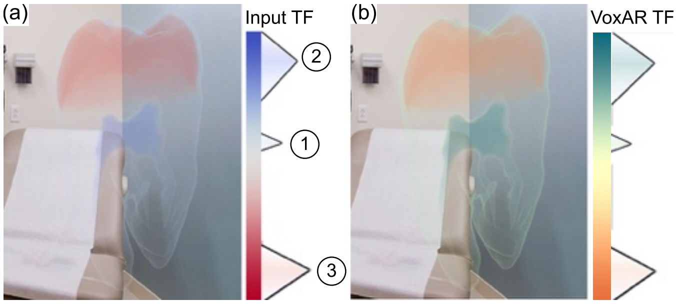

Fig. 4. Surface-like shaded rendering of a tooth volume showing the crown ② and pulp ③ enclosed in the enamel membrane ①, visualized using (a) diverging TF and (b) the adjusted VoxAR TF.

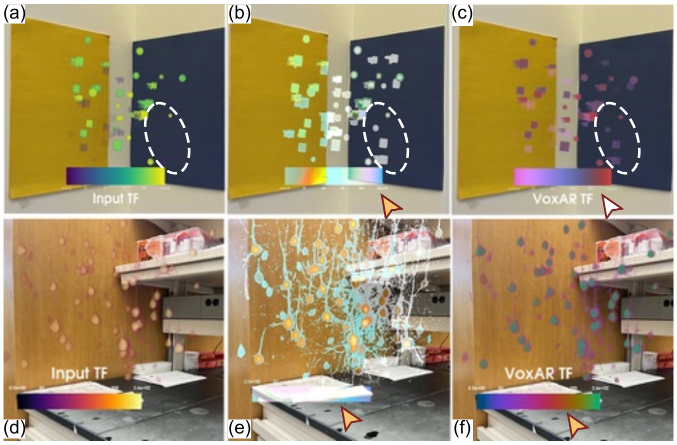

Fig. 5. For a Viridis TF in (a), (b) is the result using [4], and (c) is the VoxAR result. The white dotted annotation shows that both [4] and VoxAR improve visibility against the background, however, VoxAR maintains the TF color consistency, as seen on the color bar pointed by the arrow. Likewise, for an inferno TF in (d) rendering the neuron dataset, (e) is the result using [4], and (f) is the VoxAR result.

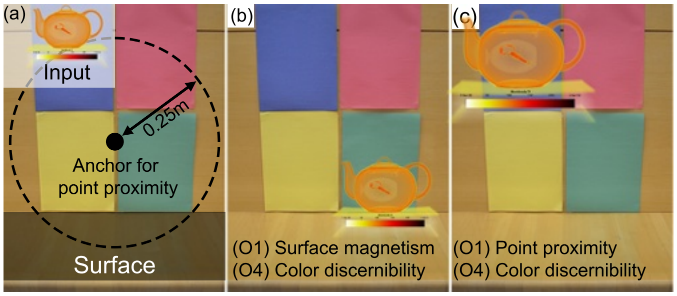
(a)
Input
0.25m
Anchor for point proximity
Surface
(b)
(O1) Surface magnetism
(O4) Color discernibility
(c)
(O1) Point proximity
(O4) Color discernibility

Fig. 6. Evaluation of VoxAR surface placement. For an input TF to be positioned in a scene with colored placeholders and a surface shown in (a), (b) and (c) demonstrate the results based on the objectives provided.

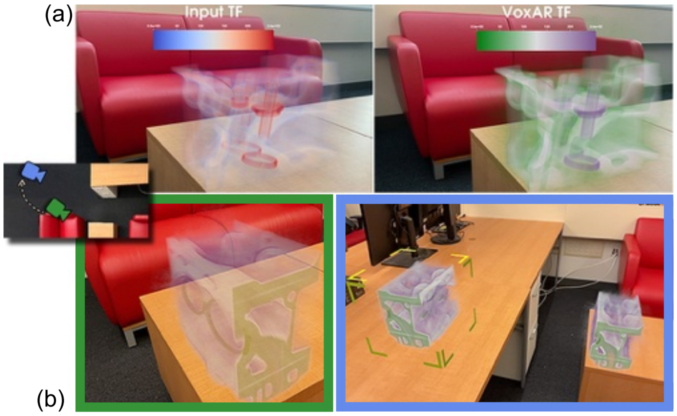

Fig. 7. (a) Engine rendered at VoxAR position, using input diverging TF (*left*) and VoxAR TF (*right*). (b) Placement update for the green to blue FoV.

As the user moves to the blue FoV, the placement score drops below a defined threshold of 80% of the original score, and a new *optimal* placement is suggested to the user using a bounding box shown in Fig. 7(c). The volume is updated to the new position upon performing a pinching gesture.

A. User Study

For our user study, we recruited 16 participants: 11 males and 5 females, aged between 19 and 35 (Mean: 27.6±4.7). No participants were color-blind. Since this work is specifically for volume rendering visualization, we required the participants to have an understanding of volume rendering and TFs.

a) Experiment Design: The study was carried out in two sequential parts. First, each participant was asked to place the Engine volume in a simulated 3D scene using one of two predefined objective combinations (**Part I**). The Engine dataset was modified to contain cube and sphere volume primitives (explained later in this section). We divided the setup, so two participants had the same objective combination. Moreover, we ensured that the 3D scene and objective combination resulted in similar difficulty for each setup.

Next, we computed an adjusted TF for the user-configured placement and a VoxAR placement using the same objectives. As a result, we generated four scenarios:

S1 User-configured placement + input TF, (UP+OTF)
S2 User-configured placement + its VoxAR TF, (UP+VTF)
S3 VoxAR placement + input TF, (VP+OTF)
S4 VoxAR placement + VoxAR TF, (VP+VTF)

For each scenario, participants performed two tasks (**Part II**):

T1 Count the number of volumetric primitives of type X.
T2 Count the number of volumetric primitives of type Y that have an intensity value in the range I.

Part I: Each participant was shown a photogrammetry-reconstructed [40] 3D scene in Unity and asked to find an optimal placement for a volume that is either on (1) a surface area or (2) anywhere in the view 2.5m from a given point, and for both, placed as close to the center of the FoV as possible. To aid the participants with the objectives, we provided 3D visuals in the scene as shown in Fig. 8 (a).

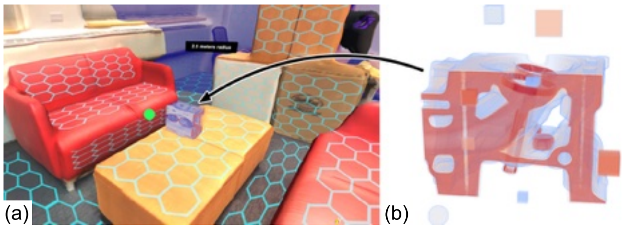

Fig. 8. Screenshot of the visual cues provided to the participants for **Part I**. The transparent 'honeycomb' texture marks valid surface areas, the purple region indicates out-of-bound, and the green dot indicates screen center.

Part II: We conducted the study using HoloLens2 and adopted a within-subject design with two independent variables. Specifically, for each participant, we presented four scenarios, **S1** to **S4**, and asked them to complete tasks **T1** and **T2** for all the scenarios. To avoid learning, the Engine volume was modified for each scenario to include cube and sphere volumes of randomized frequency (with a total cube+volume count of 10), positions, sizes ranging between $10 \times 10 \times 10$ and $20 \times 20 \times 20$ voxels, and intensity values between 0 - 255 (see Fig. 8 (b) for an example). We used the diverging TF for **S1** and **S3**, and for ease of identifying intensity ranges in **T2**, we binned the TF colors into five uniform-sized bins. Moreover, for counterbalancing our findings, each participant was presented **S1** to **S4** in a random order.

Before starting **Part II**, we first performed eye and display calibration of the HoloLens, followed by a warm-up session to help the participants familiarize themselves with using the HMD. During warm-up, they were shown a different volume and TF and were asked to practice the hand-gesture-based rotation interactivity. At the start of each trial, the participants were seated where the pre-defined FoV was measured for **Part I** and were asked to respond to the tasks *"as accurately and efficiently as possible."* Based on our current implementation of the VoxAR system, the participants were only allowed to rotate the volume. After every trial, the participants were asked a series of qualitative questions.

b) User Study Results: We present an evaluation of VoxAR by analyzing the quantitative and qualitative responses for **T1** and **T2**, for all scenarios, based on Absolute Error and Task Completion Time. For our analysis, we define **S1** (UP + OTF) as the baseline condition and use it to compare with (UP+VTF), (VP+ OTF), and (VP+VTF).

Absolute Error: We define absolute error as the average measure of how much the answers of the participants differ from the correct answer. Fig. 9 shows a plot of this measure for each scenario, averaged over the total number of trials. Based on this result, we can see that the end-to-end VoxAR technique, placement optimization followed by TF adjustment, significantly decreases the mean absolute error. That is to say, for both tasks, the participants were able to perform the data analysis tasks more accurately. Compared to UP+OTF, VoxAR reduces the mean absolute error by 65.8% and 69.6% for **T1** and **T2**, respectively. An application of Friedman's test confirmed that there is a significant effect on the recognition of an element of volumetric objects: Q=16.9; p<.001 for **T1** and (Q=12.8; p<.01 for **T2**. Pairwise comparisons using the

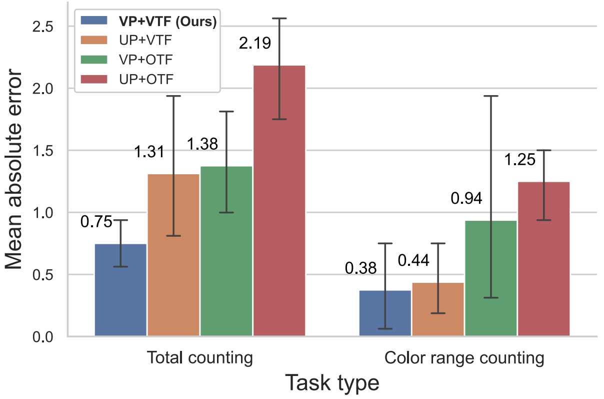

Fig. 9. Comparison of mean Absolute Error between **T1** and **T2** for all scenarios (95% confidence interval).

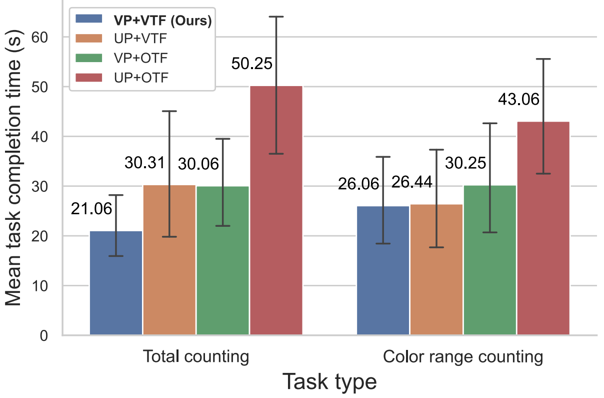

Fig. 10. Comparison of mean task completion time between **T1** and **T2** for all scenarios (95% confidence interval).

Nemenyi post-hoc test indicate that the difference between the baseline (UP+OTF) and ours (VP+VTF; VoxAR) in both of the tasks is significant (p=0.001; p=0.021, respectively). In observing the reasons for the difference, we noticed that, as shown in Fig. 11, the background blending made the participants prone to missing smaller primitives (as marked by the blue ring). Moreover, given the similarity of the red TF color with the background, most users misconceived the hole in the volume as a primitive (marked by the white ring). However, using the VoxAR TF, participants could deduce that the appearance of the background color represents a hole. The findings also confirm that using either of the VoxAR components, placement optimization or TF adjustment, can improve data perception in OST-AR.

Task Completion Time: Next, we measure the task completion time of the two tasks over the total number of trials, as shown in Fig. 10. The results show that VoxAR reduces the time taken to complete each task by 58.1% and 39.5%, on average, respectively. An application of Friedman's test shows that there is a significant effect on the completion time (Q=24.5; p¡.001 for **T1**) (Q=13; p¡.005 for **T2**). Pairwise comparisons using the Nemenyi post-hoc test indicate that the difference between UP+OTF and ours, VP+VTF and VoxAR, is significant (p=0.001; p=0.007, respectively).

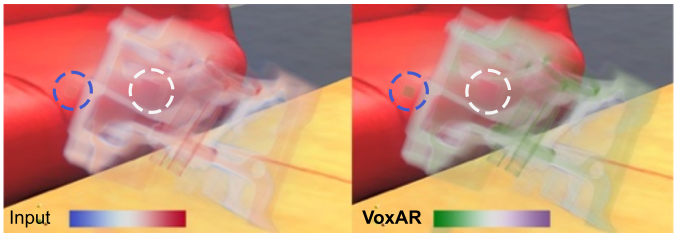

Fig. 11. An example of instances where VoxAR TF aided users to better perceive data, compared to the input TF.

c) Subjective Feedback: To collect findings for perceived performance, effort, and the certainty of the two tasks, we asked qualitative questions based on the Semantic differential scale [41], at the end of each task. Each question consisted of ratings ranging from 0 to 5, and was anchored by bipolar adjectives. A higher rating indicated that the participant was more confident in their abilities or had a higher positive response towards the condition. The results did not show a uniform tendency across questions. However, the lowest rated condition among all participants was consistently UP+OTF (Mean: 3.3 ± 1.2). The mean ratings of our full method (VP+VTF) was 3.67, and the other two conditions (UP+VTF and VP+OTF) were rated similarly (3.65, 3.79, respectively).

B. System Performance

We used an Intel Xeon Bronze 3106 CPU with 64GB of RAM and an Nvidia Quadro RTX 6000 as the remote render server. On average, VoxAR placement achieved framerates above the HoloLens target framerate of 60Hz [42]. Considering surface placement alone, VoxAR achieves $>$ 80fps, dependent on placeable surfaces in view. For 3D placement, we achieve 70-80fps; considering both, we achieve 53-65fps. The performance of VoxAR regarding surface placement is more stable, given that the system only needs a single placement map. Performance regarding 3D placement is more variable as it is negatively correlated with the resolution of the 3D space considered. The VoxAR TF adjustment is CPU-based and has an $O(n)$ time complexity, depending on the number of TF control points. The diverging TF with 3 control points took 5s, whereas the Viridis TF, which is a continuous TF and has the maximum number of control points (255) took 15s to optimize. As TF adjustment occurs only once, the system can perform in real-time after initial placement.

VIII. Conclusion, Limitations, and Future Work

This paper presented VoxAR, a two-step approach for enhancing volume rendering visualization in OST-HMDs. Our method combines spatial and environmental constraints with user preferences to find an optimal placement for the volume at runtime. Once placed, it adjusts the input TF to improve its visual distinctiveness with the real-world background, while also maintaining the perceptual mapping between the data attributes and the input TF colors. Furthermore, we have provided a solution to extend the VoxAR method for changing FoV, thus supporting a key utility of AR applications to allow interactivity. To evaluate VoxAR, we have demonstrated potential applications and compared our technique with recent work in OST-AR color enhancement. We also carried out a user study, and its findings suggest that VoxAR facilitates effective and efficient user performance when conducting volume rendering-related comprehension tasks in OST-AR.

In solving a novel challenge for AR-based scientific visualization, VoxAR faces several limitations. Some are hardware-related. For instance, OST-HMDs cannot project dark colors, such as black, thus limiting the lower luminance range of TF color options. Additionally, many OST-AR devices do not contain a dedicated GPU, thus affecting our shader-driven performance and needing to offload our volume rendering pipeline. However, recently, Magic Leap released their second generation OST-AR HMD [43], which allows global (the entire view) and segmented (area where digital content appears) visor dimming. According to the manufacturers, this utility is designed to improve visibility of virtual content across bright ambient light conditions, such as outdoors or operating rooms. To this end, for future work, we intend to examine how the visor-dimming feature can optimally aid in color enhancement.

At the technique level, although we address changing FoV, further challenges for dynamic scenes need to be addressed. Specifically, converging to a solution becomes difficult as the color spectrum in the background surrounding the initial placement broadens. As such, we aim to investigate the possibility of dynamically adapting the TF colors for immediate yet changing backgrounds, such that the TF update would minimally affect the perceptual mapping of its colors to the data attributes. This also implies studying the impact of such changes on data analysis and reasoning during real-time and dynamic visualization adaptation.

In its current state, VoxAR focuses primarily on alleviating color blending and does not account for the relationship between color and alpha TF, with respect to the background. We plan to investigate this relationship to ensure consistent and robust visualization of semi-transparent volumes and nested structures in surface-like shading.

Acknowledgments

This research was supported in part by NSF awards IIS2107224, IIS2107328, IIS1901030, NCS-FO2124179, OAC1919752, ICER1940302, and an IBM-SUNY grant 2106.

References

[1] M. Mathur, J. M. Brozovich, and M. K. Rausch, "A brief note on building augmented reality models for scientific visualization," *Finite Elements in Analysis and Design*, vol. 213, p. 103851, 2023.

[2] Unity Technologies, "Advanced workflows for AR developers: Unity MARS," https://unity.com/products/unity-mars, 2023.

[3] J. a. M. Evangelista Belo, M. N. Lystbæk, A. M. Feit, K. Pfeuffer, P. Kán, A. Oulasvirta, and K. Grønbæk, "AUIT – the adaptive user interfaces toolkit for designing XR applications," in *ACM Symposium on User Interface Software and Technology*, 2022.

[4] Y. Zhang, R. Wang, Y. Peng, W. Hua, and H. Bao, "Color contrast enhanced rendering for optical see-through head-mounted displays," *IEEE Transactions on Visualization & Computer Graphics*, vol. 28, no. 12, pp. 4490–4502, 2022.

[5] Y. Itoh, M. Dzitsiuk, T. Amano, and G. Klinker, "Semi-parametric color reproduction method for optical see-through head-mounted displays," *IEEE Transactions on Visualization & Computer Graphics*, vol. 21, no. 11, pp. 1269–1278, 2015.

[6] K.-H. Lee and J.-O. Kim, "Visibility enhancement via optimal two-piece gamma tone mapping for optical see-through displays under ambient light," *Optical Engineering*, vol. 57, no. 12, p. 123104.

[7] T. Fukiage, T. Oishi, and K. Ikeuchi, "Visibility-based blending for real-time applications," in *IEEE International Symposium on Mixed and Augmented Reality*, 2014, pp. 63–72.

[8] C. Weiland, A.-K. Braun, and W. Heiden, "Colorimetric and photometric compensation for optical see-through displays," in *Universal Access in Human-Computer Interaction: Intelligent and Ubiquitous Interaction Environments Conference.* Springer, 2009, pp. 603–612.

[9] J. David Hincapié-Ramos, L. Ivanchuk, S. K. Sridharan, and P. P. Irani, "SmartColor: Real-Time Color and Contrast Correction for Optical See-Through Head-Mounted Displays," *IEEE Transactions on Visualization & Computer Graphics*, vol. 21, no. 12, pp. 1336–1348, 2015.

[10] A. Erickson, K. Kim, G. Bruder, and G. F. Welch, "A Review of Visual Perception Research in Optical See-Through Augmented Reality," in *International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, F. Argelaguet, R. McMahan, and M. Sugimoto, Eds. Eurographics, 2020.

[11] K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, and B. H. Thomas, *Immersive Analytics*. Springer, 2018, vol. 11190.

[12] D. Lindlbauer, A. M. Feit, and O. Hilliges, "Context-aware online adaptation of mixed reality interfaces," in *ACM Symposium on User Interface Software and Technology*, 2019, p. 147–160.

[13] E. Marino, F. Bruno, and F. Liarokapis, "Color harmonization, deharmonization and balancing in augmented reality," *Applied Sciences*, vol. 11, no. 9, 2021.

[14] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Springer, 2006, pp. 75–102.

[15] R. Du, E. Turner, M. Dzitsiuk, L. Prasso, I. Duarte, J. Dourgarian, J. Afonso, J. Pascoal, J. Gladstone, N. Cruces, S. Izadi, A. Kowdle, K. Tsotsos, and D. Kim, "DepthLab: Real-Time 3D Interaction with Depth Maps for Mobile Augmented Reality," in *ACM Symposium on User Interface Software and Technology*, 2020, p. 829–843.

[16] R. Gal, L. Shapira, E. Ofek, and P. Kohli, "Flare: Fast layout for augmented reality applications," in *IEEE International Symposium on Mixed and Augmented Reality*, 2014, pp. 207–212.

[17] Y. Cheng, Y. Yan, X. Yi, Y. Shi, and D. Lindlbauer, "SemanticAdapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections," in *ACM Symposium on User Interface Software and Technology*, 2021, p. 282–297.

[18] P. Fleck, A. Sousa Calepso, S. Hubenschmid, M. Sedlmair, and D. Schmalstieg, "RagRug: A toolkit for situated analytics," *IEEE Trans Vis Comput Graph*, vol. 29, no. 7, pp. 3281–3297, 2023.

[19] Microsoft, "Microsoft Mixed Reality Toolkit v2.8.3," https://github.com/Microsoft/MixedRealityToolkitUnity/releases, 2022.

[20] K. Kiyokawa, Y. Kurata, and H. Ohno, "An optical see-through display for mutual occlusion of real and virtual environments," in *IEEE and ACM International Symposium on Augmented Reality*, 2000, pp. 60–67.

[21] O. Cakmakci, Y. Ha, and J. P. Rolland, "A compact optical see-through head-worn display with occlusion support," in *IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2004, pp. 16–25.

[22] Y. Itoh, T. Langlotz, J. Sutton, and A. Plopski, "Towards indistinguishable augmented reality: A survey on optical see-through head-mounted displays," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–36, 2021.

[23] J. L. Gabbard, J. E. Swan, J. Zedlitz, and W. W. Winchester, "More than meets the eye: An engineering study to empirically examine the blending of real and virtual color spaces," *IEEE Virtual Reality Conference*, pp. 79–86, 2010.

[24] T. Langlotz, M. Cook, and H. Regenbrecht, "Real-time radiometric compensation for optical see-through head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 11, pp. 2385–2394, 2016.

[25] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, S. Heining, B. Ockert, E. Euler, and N. Navab, "Real-time volume rendering for high quality visualization in AR," in *Augmented Environments for Medical Imaging Including AR in Computer-aided Surgery*, 2008, pp. 104–113.

[26] Y. Chu, X. Li, X. Yang, D. Ai, Y. Huang, H. Song, Y. Jiang, Y. Wang, X. Chen, and J. Yang, "Perception enhancement using importance-driven hybrid rendering for AR based endoscopic surgical navigation," *Biomedical Optics Express*, vol. 9, no. 11, pp. 5205–5226, 2018.

[27] W. Luo, A. Lehmann, H. Widengren, and R. Dachselt, "Where should we put it? Layout and placement strategies of documents in augmented reality for collaborative sensemaking," in *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 2022.

[28] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer, "Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment," *IEEE Transactions on Visualization & Computer Graphics*, vol. 27, no. 02, pp. 1171–1181, 2021.

[29] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the art in transfer functions for direct volume rendering," *Comput Graph Forum*, vol. 35, no. 3, pp. 669–691, 2016.

[30] ISO/CIE 11664-4, "Colorimetry—Part 4: CIE 1976 L*a*b* colour space," *European Standard*, 2019.

[31] M. R. Luo, G. Cui, and B. Rigg, "The development of the cie 2000 colour-difference formula: Ciede2000," *Color Research & Application*, vol. 26, no. 5, pp. 340–350, 2001.

[32] M. D. Fairchild, *Color Appearance Models*. John Wiley & Sons, 2013.

[33] M. Stone, D. A. Szafir, and V. Setlur, "An engineering model for color difference as a function of size," in *Color and Imaging Conference*. Society for Imaging Science and Technology, 2014, pp. 253–258.

[34] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss, "Colorgorical: Creating discriminable and preferable color palettes for information visualization," *IEEE Transactions on Visualization & Computer Graphics*, vol. 23, no. 1, pp. 521–530, 2017.

[35] M. Mahy, L. Van Eycken, and A. Oosterlinck, "Evaluation of Uniform Color Spaces Developed after the Adoption of CIELAB and CIELUV," *Color Research & Application*, vol. 19, no. 2, pp. 105–121, 1994.

[36] D. A. Szafir, M. Stone, and M. Gleicher, "Adapting color difference for design," in *Color and Imaging Conference*, vol. 2014. Society for Imaging Science and Technology, 2014, pp. 228–233.

[37] Unity Technologies, "Unity3D," https://unity3d.com/unity/, 2023, online; accessed 1-March-2023.

[38] Kitware, "Holographic Remoting: Stream VTK to the HoloLens 2," https://www.kitware.com/stream-vtk-to-the-hololens-2.

[39] Microsoft, "Holographic remoting — MRTK2," https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/tools/holographic-remoting?view=mrtkunity-2022-05.

[40] M. Queisner, M. Pogorzhelskiy, C. Remde, J. Pratschke, and I. M. Sauer, "VolumetricOR: A new approach to simulate surgical interventions in virtual reality for training and education," *Surgical Innovation*, vol. 29, no. 3, pp. 406–415, 2022.

[41] J. G. Snider and C. E. Osgood, *Semantic Differential Technique; a Sourcebook*. Aldine Publishing Company, 1969.

[42] "App quality criteria Mixed Reality," https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/app-quality-criteria-overview/, accessed 1-March-2023.

[43] Magic Leap Inc., "Magic Leap 2," https://www.magicleap.com/magic-leap-2, 2023, online; accessed 1-March-2023.

Saeed Boorboor is a Principal Research Scientist at the Center for Visual Computing, Stony Brook University. He received his Ph.D. in Computer Science from Stony Brook University and Bachelor's degree in Computer Science from School of Science and Engineering, Lahore University of Management Sciences, Pakistan. His research interests include scientific visualization, immersive analytics, and medical imaging.

Matthew Castellana is currently pursuing a Ph.D. degree in Computer Science at Stony Brook University. He received his B.S. in Computer Science and B.E. in Computer Systems and Engineering from Rensselaer Polytechnic Institute. His research interests include virtual and augmented reality, novel VR/AR peripherals and interfaces, remote collaboration, graphics, and related areas.

Yoonsang Kim is currently pursuing a Ph.D. degree in Computer Science at Stony Brook University. He received his Bachelor's degree in Computer Science from Soongsil University, South Korea. His research interests include multi-user scenarios, platform/device-agnostic scenarios in XR, AR privacy and security, and computer graphics.

Chen Zhu-Tian is a postdoc at the Visual Computing Group at Harvard University. Before joining Harvard, he was a postdoc at University of California San Diego, and a Ph.D. student at Hong Kong University of Science and Technology. His interests are in Information Visualization, Human-Computer Interaction, and Augmented Reality.

Johanna Beyer received the PhD degree in computer science with the University of Technology Vienna, Austria in 2009. She is currently a research associate with Visual Computing Group, Harvard University. Before joining Harvard, she was a postdoctoral fellow with Visual Computing Center, KAUST. Her research interests include scalable methods for visual abstractions, large-scale volume visualization, and immersive analytics.

Hanspeter Pfister is currently An Wang professor of computer science with the John A. Paulson School of Engineering and Applied Sciences, Harvard University. His research interests include visual computing, including intersection of scientific visualization, information visualization, computer graphics, and computer vision and spans a wide range of topics, including biomedical image analysis and visualization, image and video analysis, and visual analytics in data science.

Arie E. Kaufman is a Distinguished Professor of Computer Science, Director of Center of Visual Computing, and Chief Scientist of Center of Excellence in Wireless and Information Technology at Stony Brook University. He served as Chair of Computer Science Department, 1999-2017. He has conducted research for >40 years in visualization, VR and graphics, and published >350 refereed papers. He was the founding Editor-in-Chief of IEEE TVCG, 1995-98. He is an IEEE Fellow, ACM Fellow, National Academy of Inventors Fellow, recipient of IEEE Visualization Career Award (2005), and inducted into the IEEE Visualization Academy (2019). He received his Ph.D. in Computer Science from Ben-Gurion University, Israel (1977).