

VR ImagePlot: Visualización 3D de galerías de imágenes

Leonardo Bareño

Resumen—Se presenta un Sistema Interactivo en Tiempo Real (RIS) de Realidad Virtual para visualización tridimensional de imágenes pictóricas, mencionando el origen de la propuesta, estructura de la solución, proceso de desarrollo, descripción de las pruebas de usuario y resultados obtenidos.

Palabras Clave—Realidad Virtual, Visualización de Datos, Humanidades Digitales, Unity3D, Samsung Gear VR

1 MOTIVACIÓN

La base conceptual de VRIP está en los trabajos de Lev Manovich [1] (autor reconocido en los ámbitos de cultura digital, nuevos medios y humanidades digitales) en donde él y sus colaboradores utilizaron una variante de la visualización 2D *scatterplot* con el fin de obtener lo que él denomina “Espacios de Estilo”, consistentes en una descripción cuantitativa de rasgos que permitirían distinguir un determinado estilo pictórico en un conjunto de imágenes, a partir de características visuales de cada imagen digitalizada (promedios de tono, saturación y brillantez, entropía, etc). En el marco de estos estudios, se elabora ImagePlot [2]: un complemento para el software ImageJ escrito en lenguaje Java y empleando librerías de interfaz hacia IJ, que permite generar tales visualizaciones teniendo como entradas un conjunto de imágenes y una tabla con la información precalculada de características. El resultado es una imagen 2D con dimensión lateral entre 2000 y 20000 píxeles que escala y posiciona cada minatura en un plano coordinado XY. Algunos otros de los resultados obtenidos se pueden apreciar en la cuenta de Flickr del grupo de investigación [3].

En el ámbito de los estudios teóricos de visualización de datos, por varios años expertos como Tamara Munzner han recomendado enfáticamente la no utilización de visualizaciones 3D [4], entre otros motivos, debido a que la inmersión se realiza a costa de una menor resolución de imagen; esta última siendo una característica más valiosa. No obstante, tras mejores prestaciones tecnológicas que han llegado con dispositivos como Oculus Rift o HTC Vive en aspectos tales como la resolución, han surgido posturas recientes [5] que son más favorables a este respecto, y que exhortan a una reevaluación de lo afirmado por Munzner.

Con este software de VR se pretende explorar el alcance, ventajas y desventajas de desplegar tales visualizaciones en una representación tridimensional, además de brindar al usuario funcionalidades de navegación inmersiva a través de las imágenes, y de selección del conjunto de datos que desea explorarse.

2 SOLUCIÓN

VRIP es un software en plataforma Android 6.0 (API 23) elaborado con Unity v. 2017.4.19 (no se eligió la última versión existente en 2019 por motivos de mantener compatibilidad con varias de las librerías que serán referidas a continuación) para ser usado con la combinación de dispositivos: HMD Samsung Gear VR, teléfono Samsung Galaxy S7 y controlador Samsung Gear VR Controller. Iniciado su desarrollo a partir de código fuente de demostración suministrado por Oculus, incorpora librerías para manejo del controlador, gestor de base de datos SQLite, e interfaz gráfica de usuario (GUI). El contenido original elaborado por el suscrito se concentra principalmente en el directorio `./assets/LB` (a excepción de los scripts `CanvasVRModifier.cs` y `ColorConverter.cs`), así como en algunas modificaciones realizadas a nivel de scripts de GUI.

• Email: contact.lb.apps@gmail.com

2.1 Conjunto de datos

Tras el estudio de varias alternativas, p ej: Met Museum [6] o MoMA [7] las cuales ofrecen abiertamente en Github las bases de datos completas de sus contenidos digitales pero que a diferentes niveles imponen restricciones tanto en acceso directo a los archivos de imágenes digitales como en términos y condiciones de uso, o BAM Dataset [8] que requiere de una autenticación previa que impondría demoras al preparar la descarga masiva; se optó por usar imágenes procedentes de la página húngara Web Gallery of Art (WGA), la cual es una colección de imágenes de bellas artes y arquitectura comprendidas entre los s. III a XIX iniciada desde 1996 [9] y que aún conserva un esquema de diseño web que facilita su exploración programática. WGA también ofrece mediante un archivo Excel la lista completa de su colección, junto con URLs e información adicional de cada obra [9].

De una cantidad inicial de 46713 ítems de obras, manejadas en 2 conjuntos: *art* (imágenes de resolución completa con dimensiones entre 700 y 1500 píxeles de lado, tamaño total: 6.91 GB) y *detail* (resolución reducida a 400 píxeles de lado, tamaño total: 937 MB), fueron descartados 2001 ítems de imágenes en escala de grises (8 bits) para mantener homogeneidad en el formato de la imagen (RGB, 24 bits) durante la etapa de procesamiento; por lo cual la cantidad final neta para uso en el aplicativo es de 44712 ítems. Eventualmente en la fase de pruebas en alfa, fue generado a partir del conjunto *detail* un nuevo conjunto derivado, con reducción de dimensiones al 20% (tamaño total: 63.4 MB) para generar un menor consumo de memoria RAM en el despliegue de girasoles.

Para cada imagen en la colección *art* se realizó un procesamiento en lote con la librería `imagej.jar`, en el que se obtuvieron 12 valores numéricos escalares respecto de todos los píxeles de color que la componen:

- Promedio de tono
- Promedio de saturación
- Promedio de brillantez
- Mediana de tono
- Mediana de saturación
- Mediana de brillantez
- Desviación estándar de tono
- Desviación estándar de saturación
- Desviación estándar de brillantez
- Entropía de tono
- Entropía de saturación
- Entropía de brillantez

Vale aclarar que para algunos casos al generar las visualizaciones, en [1] se recomienda el uso de la mediana en vez del promedio, por ser este primero el menos afectado por valores atípicos (*ouliers*).

Un decimotercer valor ingresado posteriormente a los datos de los ítems es denominado *date_num*: datos derivados del campo *date* (año estimado de elaboración de la obra) que fueron sometidos a ajustes de formato para homogeneizarlos como numéricos. Estas 13 características de la imagen son utilizadas en el cuadro de *Settings*, en los Dropdowns de X, Y y Z para seleccionar 3 de ellas, y

emplearlas luego en el posicionamiento espacial de cada girasol/esfera.

Posterior a esto, se ingresaron los datos al interior de un archivo de base de datos de SQLite (./Assets/StreamingAssets/wga_hu_bd.sqlite), cuya clave principal es una cadena con la ruta relativa y nombre del archivo de imagen.

2.2 Controlador y navegación

El GearVR Controller es un dispositivo conectado al Galaxy S7 vía Bluetooth que no puede capturar datos de su posición sino únicamente los de su rotación (es decir, es 3DOF). Esto, además de que cuenta con un menor número de botones que el Oculus Touch o HTC Vive Controller, implicó realizar una disposición no convencional que aprovechara al máximo las reducidas capacidades del dispositivo.

Para realizar las funciones de navegación y de selección de elementos, en el Gear VR Controller se configuraron:

- Movimiento horizontal/vertical sobre el touchpad sin presionar su botón: La cámara se mueve orbitando esféricamente alrededor del punto central del área de interés, permaneciendo su rotación siempre orientada hacia dicho punto.
- Movimiento vertical sobre el touchpad manteniendo presionado su botón: Aumentar o disminuir el radio esférico. Este movimiento permite la inmersión dentro del conjunto de girasoles/esferas desplegados.
- Botón de trigger: Selección de elementos del menú (settings), y de girasoles/esferas para abrir panel de detalles.

2.3 Elementos de espacio virtual

2.3.1 Espacio de despliegue de ítems

Se dispuso con un fondo de color negro uniforme, debido al efecto conocido de la influencia en percepción visual que un color ejerce sobre otro cuando lo rodea. (Otra configuración observada en imágenes producidas por ImagePlot fueron fondos con tonos grises como RGB(70,70,70) o RGB(136,136,136)). A fin de orientar al usuario sobre la pose que actualmente tiene, se colocaron también 3 flechas refiriéndose a los ejes coordenados X, Y y Z (en el lenguaje visual de la herramienta Unity en donde X está en rojo, Y en verde y Z en azul). En la parte inferior izquierda se encuentra un botón de GUI para abrir el menú (settings).

Dos tipos de elementos pueden ser desplegados para cada ítem:

- **Girasoles:** Imágenes en miniatura colocadas sobre un cuadrilátero plano, cuyo vector normal está modificándose permanentemente mediante código para estar siempre orientado hacia la posición de la cámara. Al utilizar texturas 2D para desplegar los mapas de bits de cada imagen, el total de girasoles están consumiendo una importante cantidad de memoria RAM que impone una limitación sobre el número máximo de girasoles que pueden ser cargados.

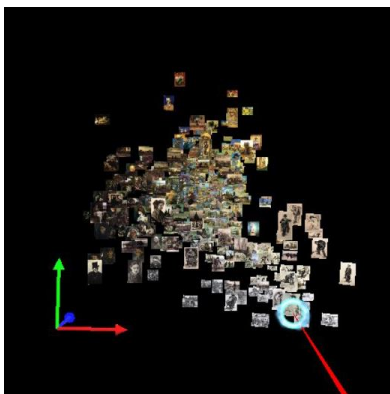


Fig. 1. Ejemplo de despliegue con girasoles.

- **Esferas:** A semejanza a los puntos del gráfico *scatterplot* originario, cuyo color equivale al promedio de color de la imagen. Al no estar consumiendo texturas, esto permite que puedan ser colocadas en pantalla un mayor número de esferas que de girasoles, en caso de ser necesario.

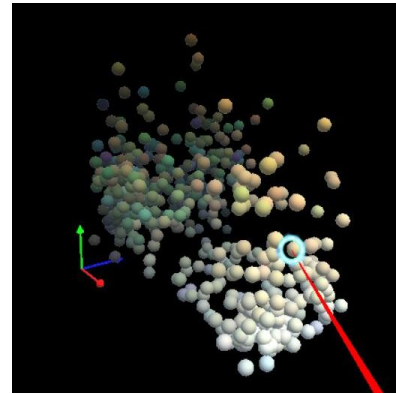


Fig. 2. Ejemplo de despliegue con esferas.

2.3.2 Interfaz gráfica Settings

Permite filtrar el número de imágenes desplegadas, establecer cuáles valores numéricos se utilizarán en X, Y, Z, y configurar otros parámetros de despliegue.

Por ser este software de propósito experimental, no se impuso una validación de configuraciones más allá de los valores permitidos (es decir, son admitidas configuraciones que generan un query con una cantidad de resultados que eventualmente podrían colapsar la ejecución debido a agotamiento de RAM).

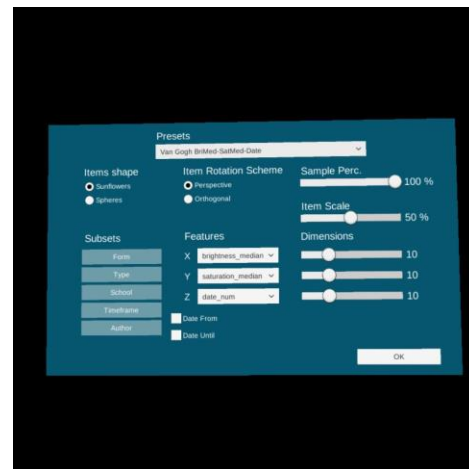


Fig. 3. Interfaz gráfica Settings.

En su construcción, para elementos como botones, sliders, RadioButtons y CheckBoxes se utilizaron los controles por omisión ofrecidos por Unity, los cuales funcionaron correctamente en VR. Para el caso de los elementos Listbox y Dropdown, estos requirieron de algunas modificaciones desde código fuente. Para estos 2 elementos se utilizó el paquete *third-party* "UI Tools for Unity" obtenido del Asset Store de Unity. En Listbox se agregó código para crear funcionalidad de selección múltiple de elementos (utilizado en los campos *form*, *type*, *school*, *timeframe* y *author*). Los Dropdown requirieron de un mayor tiempo de *troubleshooting* para que funcionaran adecuadamente en VR:

- Se hallaron elementos instanciados sólo en tiempo de ejecución que también requerían de ajustes en su raycaster,

tal como es recomendado en [11] para aplicar sobre elementos de existencia permanente en la lista *Hierarchy*.

- Listbox y Dropdown instancian en tiempo de ejecución un elemento de UI llamado “Blocker” que sirve para inhibir la detección de elementos ajenos a los de la lista temporalmente desplegada (sobre todo, los elementos que quedan ocultos debajo de dicha lista). Esta característica no pudo ser correctamente adaptada a VR por lo cual se optó por una contingencia.

Es importante también señalar el hecho de que Unity Technologies ha publicado en Bitbucket para libre uso y consulta el código fuente de su módulo de UI [16], un hecho bastante inusual considerando que Unity sólo revela el código fuente de su motor de juegos a compañías de gran volumen con las que tiene acuerdos especiales, bajo circunstancias muy puntuales (p.ej. optimizaciones o problemas de rendimiento) y en países con legislaciones favorables de protección de propiedad intelectual. [17] “UI Tools for Unity” es de hecho un *fork* de este módulo, con algunas mejoras y modificaciones.

2.3.3 Cuadro de detalle de imágenes

Puede ser accionado tanto para girasoles como para esferas. Permite identificar y conocer detalles adicionales de cada imagen en pantalla. La imagen mostrada a la izquierda es procedente de la colección *art*, aplicando escalado de dimensiones para ajustarlo dentro del panel.

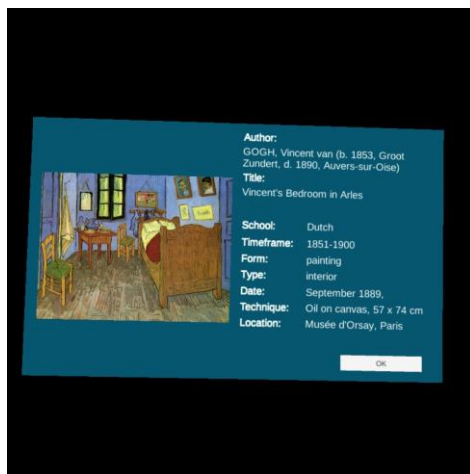


Fig. 4. Panel de detalles de imagen.

3 DESCRIPCIÓN DE PRUEBAS

Para el aplicativo, las pruebas de usuario estuvieron diseñadas para tener un indicio de cuán intuitiva fue para los usuarios la comprensión de las funcionalidades del aplicativo.

Una dificultad que había que ser resuelta consistió en el hecho de no poder contar con una forma de poder ver en PC en una pantalla replicada lo que el usuario estaba viendo en el Gear VR. Se ensayaron softwares *third-party* que transmitían streaming de la pantalla del celular hacia el PC (Samsung Sidesync, Samsung Flow, MirrorOp, Vysor, y funcionalidades incorporadas dentro de los SOs Windows 10 y Android [13]) que funcionaban bien para las pantallas de home de Android y la mayoría de apps, pero que perdían la conexión cuando se iniciaban VRIP u otras aplicaciones de Oculus. Y una solución sugerida para el Oculus Go [14] de realizar un streaming vía USB no pudo hacerse con el Gear VR al no permitir este HMD la conexión entre el teléfono y el PC, aun cuando existen casos divulgados de que por el puerto de carga del Gear VR se podían enchufar diversas clases de accesorios USB [15].

Ante esto, se realizaron modificaciones menores para reproducir clips de audio ante 5 eventos: inicio de aplicación, entrada a *Settings*, entrada a detalles, salida de settings/detalles, y uso de la acción de navegación “Movimiento vertical sobre el touchpad manteniendo presionado su botón” descrito anteriormente. Además de ello, en cada sesión de prueba se grabaron estos sonidos con un dispositivo externo.

Se realizaron un total de 7 pruebas de usuario, de las cuales 3 fueron descartadas por quedar incompletas o no haber seguido adecuadamente el procedimiento indicado.

El procedimiento inicialmente diseñado fue:

- Se les explicó verbalmente el modo de uso del controlador, y luego ingresaron a la aplicación.
- Ingresar a *Settings*, escoger preset “Van Gogh BriMed-SatMed-Date” y pulsar OK.
- Dejar que el usuario explorara el ambiente por un tiempo indeterminado.
- Ingresar a *Settings*, permitir que el usuario escogiera libremente los valores, y pulsar OK
- Dejar que el usuario explorara el ambiente por un tiempo indeterminado.

Posteriormente, en cada grabación se determinó el tiempo entre la salida de *Settings* (que marca el inicio de la generación de girasoles/esferas) y la primera entrada del usuario para ver los detalles de una imagen de su preferencia (esta última acción no le fue indicada a los usuarios que expresamente la hicieran, sino que se dejó que la intentaran como parte de la navegación libre).

En esto último, hay que tener en cuenta un comportamiento heredado del demo de Oculus de que cuando el Line Renderer intersecta a un objeto interactuable, aparece un cursor *Gaze Pointer* animado en frente del objeto. Este cursor desaparece cuando se señala hacia un punto vacío.

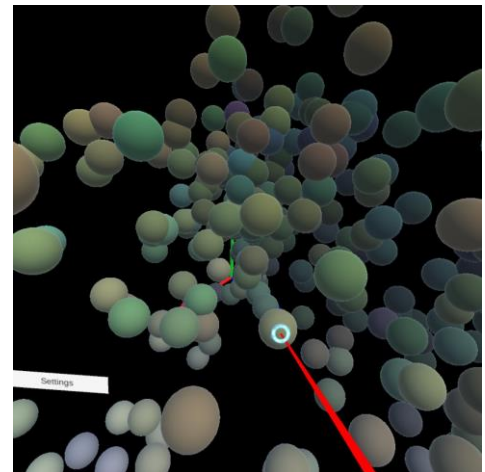


Fig. 5. Se aprecia el cursor *Gaze Pointer* (redondo, en azul) apareciendo sobre un objeto cuando es señalado con el controlador.

4 RESULTADOS

De las 4 pruebas válidas, 1 de ellas fue suspendida por manifestar el usuario una sensación de mareo; no obstante, en esta alcanzó a capturarse la información de interés.

5 TRABAJO FUTURO

El trasfondo expresado en la motivación de este software justifica la elaboración de pruebas dentro del ámbito de visualización, dirigidas a determinar la eficacia de interfaces 3D, tales como las mencionadas en las referencias citadas por Munzner

en [12]. Ahí mismo, la autora advierte sobre lo “truculento” que puede llegar a ser el diseñar experimentos controlados para hacer tales demostraciones.

En el dataset hace falta también un proceso de selección supervisada de imágenes, debido a que no todas corresponden enteramente a imágenes pictóricas. Por otra parte, existen algunos cuadros que no son rectangulares sino ovalados, y su espacio faltante en la periferia es rellenado de color blanco, lo cual altera enormemente los valores alejándolos de la realidad. Ante esto, podría optarse por una solución basada en máscaras que permitan seleccionar o descartar los píxeles que verdaderamente pertenecen a la obra.

A nivel de desarrollo, quedaron pendientes algunas características que hubiera sido interesante apreciarlas:

- Al igual que en uno de los ejemplos exhibidos en [1], usar texturas semitransparentes para expresar densidad cuando se acumulan imágenes unas detrás de otras.
- A fin de mitigar algunos efectos negativos de la distorsión perspectiva ([4], pág 122), ensayar una configuración de cámara de proyección ortogonal (más el comportamiento de esquema ortogonal de rotación de ítems ya implementado en *Settings*).

REFERENCIAS

- [1] Manovich, Lev. Style Space: How to compare image sets and follow their evolution. 6 de agosto de 2011. http://manovich.net/content/04-projects/073-style-space/70_article_2011.pdf
- [2] Código fuente de ImagePlot: <https://github.com/culturevis/imageplot> (accesado mayo 2019)
- [3] Software Studies. ImagePlot Gallery: <https://www.flickr.com/photos/culturevis/albums/72157627389144448> (accesado mayo 2019)
- [4] Munzner, Tamara. Visualization Analysis & Design. Boca Raton: CRC Press, 2015. Págs 117-130
- [5] Marriot, Kim (et al). Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualization. En: Immersive Analytics. Cham: Springer Nature Switzerland AG, 2018. Pág 26
- [6] The Metropolitan Museum of Art Open Access CSV. <https://github.com/metmuseum/openaccess> (accesado mayo 2019)
- [7] The Museum of Modern Art (MoMA) Collection. <https://github.com/MuseumofModernArt/collection> (accesado mayo 2019)
- [8] The Behance Artistic Media Dataset. <https://bam-dataset.org/> (accesado mayo 2019)
- [9] Web Gallery of Art, searchable fine arts image database. <https://www.wga.hu/index.html> (accesado mayo 2019)
- [10] Web Gallery of Art Catalogue. <https://www.wga.hu/frames-e.html?database/download/index.html> (accesado mayo 2019)
- [11] Unity's UI System in VR. <https://developer.oculus.com/blog/unitys-ui-system-in-vr/> (accesado mayo 2019)
- [12] Empirical Evidence. En sección 6.3 “No Unjustified 3D” en Munzner, Tamara. Visualization Analysis & Design. Boca Raton: CRC Press, 2015. Págs 129-130
- [13] How to Screen Mirror Android to Windows PC 2018. <https://www.youtube.com/watch?v=1bYBBYdUtqw> (accesado mayo 2019)
- [14] Oculus Go Wired Mirroring: How To Guide. <https://developer.oculus.com/blog/oculus-go-wired-mirroring-how-to/> (accesado mayo 2019)
- [15] Everything you can do with the USB port on your Samsung Gear VR. <https://www.vrheads.com/everything-you-can-do-usb-port-your-gear-vr> (accesado mayo 2019)
- [16] Unity Technologies – UI / Bitbucket. <https://bitbucket.org/Unity-Technologies/ui/> (accesado mayo 2019)
- [17] Apartado “How can I license or use Unity's source code?” en página: “FAQ: Licensing & activation.” <https://unity3d.com/unity/faq/2491> (accesado mayo 2019)