Cómo realizar una correcta gestión del color en sistemas de realidad virtual

Halina Cwierz[1], Francisco Díaz-Barrancas[1], P. J. Pardo[1], A. L. Pérez[2*] y M. I. Suero[2]

[1] *Departamento de Ingeniería de Sistemas Informáticos y Telemáticos, Universidad de Extremadura, Centro Universitario de Mérida, C/Santa Teresa de Jornet 38, Mérida E06800, España*

[2]*Departamento de Física, Universidad de Extremadura, Facultad de Ciencias, Avda. de Elvas s/n, Badajoz E06071, España*

http://grupoorion.unex.es

1. Introducción

La realidad virtual (VR) intenta representar en tiempo real escenas de la realidad, que se reconstruyen en los dispositivos informáticos mediante diferentes plataformas. Pero ¿cómo gestionan estas plataformas, internamente el color? Y ¿hasta qué punto son un fiel reflejo de la realidad?

Los sistemas de gestión del color establecen una serie de transformaciones colorimétricas, que permiten trasladar las coordenadas de un estímulo cromático de los espacios de color independientes del dispositivo (CIE XYZ, CIE Lab) a los espacios de color dependientes del dispositivo (RGB, CMYK) y viceversa. Todas estas transformaciones matemáticas requieren un tiempo de cálculo a menudo demasiado largo, ya que los valores de resolución y frecuencia de actualización de los dispositivos de realidad virtual son tales que la gestión del color se vuelve inviable desde el punto de vista técnico, ya que es necesario vincular varias transformaciones colorimétricas. En este trabajo se analiza cómo realizar una correcta gestión del color en un sistema de VR usando las posibilidades que nos ofrece *Unity* [1] para trabajar en entornos virtuales, junto con la última versión comercial de las gafas de realidad virtual *Oculus Rift* (CV1). Este dispositivo de visualización cabeza-arriba (HUD: Head-Up Display) está equipado con dos pantallas personalizadas, una por lente, fabricada por Samsung Display Co. Estas pantallas son del tipo AMOLED con una resolución nativa de 1200 x 1080 píxeles, 3.51" de tamaño diagonal y una densidad de píxeles resultante de 456 ppp.

El objetivo es incorporar un Sistema de Gestión de Color (CMS) en el software de Unity, cuyo espacio de color nativo es RGB. Partiremos de la caracterización cromática del dispositivo de visualización HUD mencionado anteriormente y la definición del perfil colorimétrico asociado a él. El cálculo de este perfil ICC se presentó como póster en la XIV Conferencia del Color [2]. Se usará la biblioteca de gestión de color *LittleCMS* [3] y mediante transformaciones colorimétricas calcularemos los valores triestímulos de diferentes fuentes de luz, caracterizadas por su distribución espectral de potencia. Posteriormente, aplicaremos una transformación de color desde el espacio de color CIE XYZ 1931 al espacio de color RGB nativo de Unity, a través de los perfiles ICC y del sistema de gestión del color. Con todo esto pretendemos mejorar la sensación de realismo en los entornos de realidad virtual, mejorando la reproducción del color en el entorno de VR de Unity.

2. Metodología

Esta comunicación tiene una doble finalidad, compartir los resultados obtenidos en nuestros trabajos de investigación a la comunidad científica, y servir como tutorial a la misma comunidad científica para poder iniciarse en la gestión del color en dispositivos de realidad virtual. Por lo tanto, en esta sección explicaremos paso a paso, cómo se ha construido la escena en Unity, el software necesario para poder gestionar el color en un entorno virtual y una explicación detallada del uso de las funciones de la biblioteca *LittleCMS*.

* *e-mail*: aluis@unex.es

Pasos a seguir para realizar una correcta gestión del color en un entorno virtual con *Unity*:

1. Descargar de la página de Little cms la última versión del código. Descomprimir y compilar la última solución propuesta (Projects\VC2017\lcms2.sln) en Visual Studio, para las dos arquitecturas, x86 y x86_64. De esta compilación obtendremos dos archivos lcms2.dll. Estas librerías son fundamentales para que en nuestro proyecto de Unity podamos utilizar las funciones para gestionar el color que proporciona *Little CMS*.
2. Crear un nuevo proyecto 3D en Unity, incluir el *Asset "Oculus Integration*" para poder trabajar con el soporte avanzado de renderizado, social, plataforma, audio y desarrollo de Avatares para dispositivos *Oculus* VR y algunos dispositivos Open VR soportados. Este *Asset* se puede descargar de forma gratuita de la página de *UnityAssetStore*.
3. El perfil ICC de *Oculus Rift* hay que incluirlo en la carpeta Resources\Sprites de Assets, si no existiera, es necesario crearla. Con estos pasos previos, ya tendríamos el soporte software básico para que podamos usar los métodos que proporciona la librería *Little CMS* y un entorno virtual para *Oculus Rift*.

En nuestro trabajo de investigación, hemos simulamos una cabina de iluminación LED *Just Normlicht*. Esta cabina incorpora 12 puntos de luz LED distribuidos en dos filas, que han sido simuladas en la versión virtual de la cabina, como se puede ver en la Fig.1

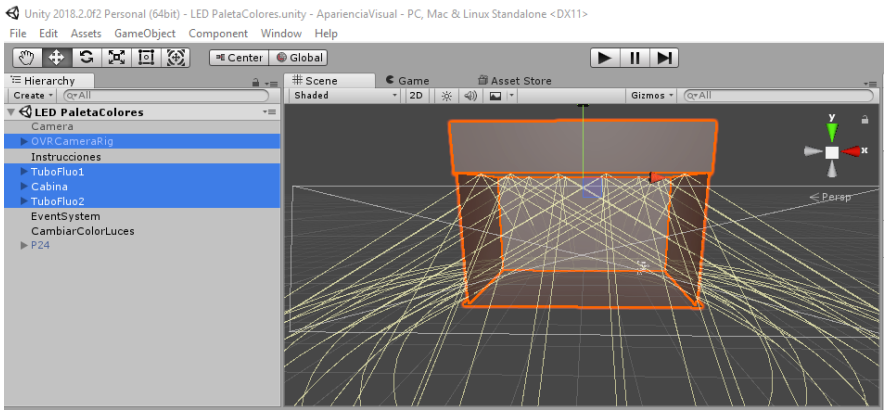

Figura 1: Simulación de cabina y focos en Unity

Se ha creado un nuevo script "*CambiarLuz.cs*" en lenguaje C# desde el menú Crear en la parte superior izquierda del panel Proyecto o seleccionando *Assets* > *Create* > *C# Script* desde el menú principal. En este script se ha cambiado el comportamiento de las luces de la cabina, para adaptarlos a nuestra gestión del color, usando las funciones que nos proporciona la librería Little CMS. Toda la documentación sobre las funciones y tipos de datos de esta librería, se puede consultar en los archivos PDF que se obtienen cuando se descarga el software.

Para poder usar cualquier función de esta librería en un script de *Unity*, hay que escribir antes de la función una llamada a la importación: [*DllImport ("lcms2")*] y, a continuación, la función que se usará en el script. Podemos ver un ejemplo en la Fig. 2.

```
//crear un perfil por defecto de tipo XYZProfile
[DllImport ("lcms2")]
private static extern IntPtr cmsCreateXYZProfile();
//crear un perfil por defecto de tipo _sRGBProfile
[DllImport ("lcms2")]
private static extern IntPtr cmsCreate_sRGBProfile();
```

Figura 2: Ejemplo para usar funciones de LittleCMS en un script de Unity

A continuación, se detallan las funciones implementadas más relevantes, relacionadas con la gestión del color. Hay que aclarar que todas las funciones de esta librería se pueden sobrecargar y adaptarlas a los tipos de datos con los que vayamos a trabajar en nuestro proyecto.

Funciones para acceder a los perfiles ICC

Abrir un perfil ICC desde un fichero
cmsHPROFILE *cmsOpenProfileFromFile* (const char *ICCProfile, const char *sAccess);
private static extern IntPtr cmsOpenProfileFromFile (string perfilicc, string lectura);

```
//usamos el perfil de Oculus
rutaicc2 = Application.dataPath + "/Resources/Sprites/Oculus.icc";
hInProfile = cmsOpenProfileFromFile (rutaicc2, "r");
```

Figura 3: Ejemplo para usar al perfil de Oculus

Crear un perfil XYZ por defecto
cmsHPROFILE *cmsCreateXYZProfile* (void);
private static extern IntPtr cmsCreateXYZProfile();

```
//Crear un perfil XYZ por defecto
XYZProfile = cmsCreateXYZProfile ();
```

Figura 4: Ejemplo para crear un perfil XYZ por defecto

Funciones para realizar transformaciones del color

Crear una transformación de color
cmsHTRANSFORM *cmsCreateTransform* (cmsHPROFILE Input, cmsUInt32Number InputFormat, cmsHPROFILE Output, cmsUInt32Number OutputFormat, cmsUInt32Number Intent, cmsUInt32Number dwFlags);
private static extern IntPtr *cmsCreateTransform* (IntPtr hInProfile, uint tipo1, IntPtr hOutProfile, uint tipo2, uint INTENT_PERCEPTUAL, uint cero);

Aplicar una transformación de color
void *cmsDoTransform* (cmsHTRANSFORM hTransform, const void * InputBuffer, void * OutputBuffer, cmsUInt32Number Size);
private static extern void *cmsDoTransform* (IntPtr Transform,ref cmsCIEXYZ InputBuffer,byte [] OutputBuffer,int size);

```
TYPE_XYZ_DBL= 4784152;
TYPE_RGB_8=262169;
xyzToRGBtransform = cmsCreateTransform (XYZProfile, TYPE_XYZ_DBL, hInProfile, TYPE_RGB_8, 3, 0);
cmsDoTransform (xyzToRGBtransform, ref xyz, rgb2, width * height);
```

Figura 5: Ejemplo de creación y aplicación de una transformación del color de XYZ a RGB

Con estas funciones implementadas en el código, ya es posible aplicar las transformaciones colorimétricas necesarias para realizar una correcta gestión del color

3. Resultados

Para comprobar la funcionalidad de nuestro sistema, se ha implementado el script "*CambiarLuz.cs*" para calcular los valores XYZ de varias fuentes de luz, a partir de su distribución espectral de potencia y el

espacio de color triestímulo CIE 1931. Usando las funciones que proporciona *LittleCMS* se han convertido los valores XYZ a RGB y se han aplicado a los focos de luz de la escena, Fig. 6.

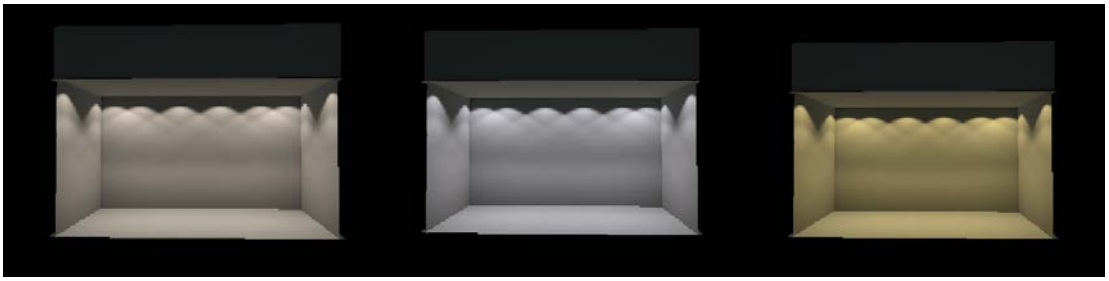

Figura 6: Ejecución del Script "CambiarLuz.cs" para la fuente luminosa TL84 (izqda), simulador D50 (centro) y simulador A (dcha).

El objetivo principal de estas transformaciones de color es comparar la fidelidad de la reproducción del color en la escena virtual iluminada con diferentes fuentes de luz mediante un sistema de gestión del color. Para comprobar su eficacia, se ha introducido en la escena de realidad virtual un *Color Checker* virtual al que se le ha cambiado la fuente luminosa. En la Fig. 7 se muestra el *Color Checker* bajo las distintas fuentes luminosas y se pueden apreciar las variaciones de color originadas por el cambio de fuente.

Figura 7: Efecto del cambio de fuente luminosa sobre el Color Checker para la fuente luminosa TL84 (izqda), simulador D50 (centro) y simulador A (dcha).

4. Conclusiones

Las pruebas realizadas han confirmado que es posible realizar una correcta gestión del color en Realidad Virtual mediante la inclusión de un sistema gestor de color, lo que redunda en una mejoría en la fidelidad en la reproducción del color en escenas de realidad virtual y la mejora de la sensación de realismo.

Agradecimientos: Este trabajo ha sido realizado gracias a las ayudas IB16004 y GR18131 de la Junta de Extremadura y ha sido parcialmente financiado por el Fondo Europeo de Desarrollo Regional.

Bibliografía

[1] "Unity Technologies," 2019. [Online]. Available: https://unity.com/es. [Accessed 2 April 2019].

[2] F. Díaz-Barrancas, P. J. Pardo, M. I. Suero and A. L. Perez, "Is it possible to apply color management technics in Virtual Reality devices?," in *XIV Conferenza del Colore*, Firenze, 2018.

[3] M. M. Saguer, "Little CMS," 2018. [Online]. Available: http://www.littlecms.com/. [Accessed 10 Feb 2019].