



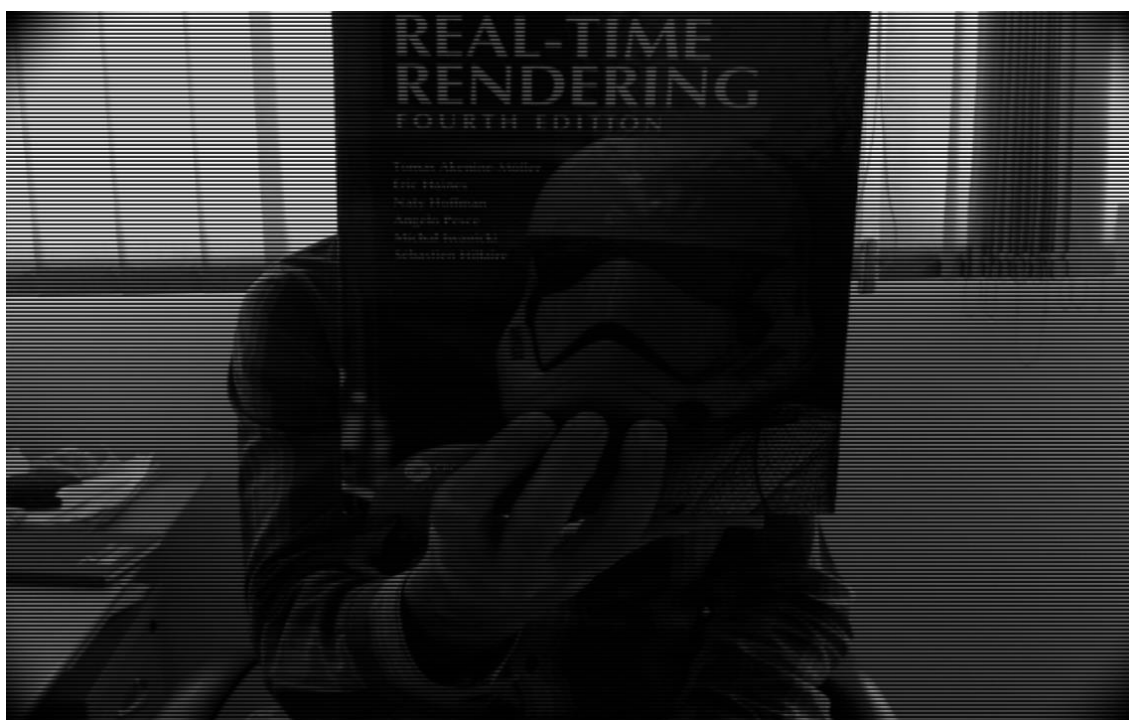
## Práctica 4

### Objetivo

El objetivo de esta práctica es estudiar la programación de la GPU usando shaders. Esta práctica se centra en el procesador de fragmentos.

### Tu trabajo

El objetivo de la práctica es conseguir el efecto de una TV antigua de blanco y negro para una escena dada. Dicho efecto emula el comportamiento de un aparato de televisión antiguo en un shader de fragmento, obteniendo un resultado como el que muestra la siguiente figura:



El programa que se proporciona al alumno (p4) captura una textura desde la primera cámara del sistema, y la muestra a ventana completa (si las razones de aspecto de la ventana y del vídeo cambian mucho, la imagen se verá distorsionada). Si no hay disponible ninguna cámara, se mostrará un vídeo pregrabado en bucle.

El programa proporcionado se encarga de cargar, compilar, enlazar e instalar los *shaders* sobre los que te tienes que basar (sólo deberías modificar el de fragmento). Inicialmente, dichos *shaders* se encargan de asignar al fragmento el color correspondiente del texel de la textura activa.

Deberás modificar el shader para conseguir los efectos mostrados en la figura anterior. Para simular una pantalla antigua, se han aplicado tres efectos:

- Conversión de color a blanco y negro
- Reducción de resolución espacial
- Vignetting.



Para implementar el primer efecto, primero deberás convertir el color del fragmento a escala de grises. Para pasar un color en RGB a escala de grises, tienes que sumar el 30% de la componente roja del color original, un 59% de la verde y un 11% de la azul. Dicho valor (escalar) se corresponde con la luminancia del color original. Para representar un color gris en una pantalla de color, tan sólo debes usar el mismo valor de luminancia en las tres componentes (p.e., el  $[0, 0, 0]$ , es negro,  $[1, 1, 1]$  es blanco, el  $[0.2, 0.2, 0.2]$  es un gris oscuro, etc).

Para implementar la pérdida de resolución espacial, deberás dibujar dos filas consecutivas de píxeles al 20% de su color gris correspondiente, luego otra al 80%, y por último otra con su valor de gris sin modificar. Se repetirá el proceso anterior con todos los píxeles de la imagen donde se dibuje un objeto. Puedes usar la variable especial de entrada `gl_FragCoord`, que contiene las coordenadas del centro del fragmento (en el sistema de coordenadas de la ventana, de tipo *float*), para decidir qué efecto aplicarle.

Por último, en la figura anterior se puede ver también el efecto de *vignetting*, común en las cámaras antiguas, y que consiste en un oscurecimiento de las esquinas. Para implementar este efecto, puedes usar la distancia entre el pixel que se está procesando y uno de los cuatro círculos:



Dado el radio de *vignetting*, puedes calcular si el pixel se corresponde con una de las cuatro esquinas, y puedes calcular un factor de oscurecimiento que tenga en cuenta la distancia desde el centro del círculo al píxel.

### **Extensiones opcionales**

Para conseguir puntuación extra puedes, por ejemplo:

- Implementar más efectos (p.e., ruido blanco, distorsión de la imagen manipulando las coordenadas de textura como, por ejemplo, distorsión de barril y cojín, desincronización de la señal horizontal...)
- añadir una GUI a tu programa para modificar distintos parámetros del efecto (por ejemplo, para simular monitores monocromo de otros colores, el radio del vignetting, etc.). Mira en el proyecto ej4-2 un ejemplo de cómo usar widgets vinculados a un uniform de tus programas.