

## Procesadores gráficos y Aplicaciones en Tiempo Real

# Mapeo Tonal

### Descripción

El mapeo tonal es un proceso que consiste en aplicar cambios de contraste local para resaltar de manera zonal las texturas de cada zona de una fotografía. El resultado es una imagen en la que se ha reducido el contraste global, y se ha obtenido a cambio una mayor riqueza de contrastes locales. La utilización correcta del mapeo tonal puede dar lugar a un mayor realismo en las imágenes ya que modela la capacidad del ojo de adaptarse a los distintos grados de luminosidad que presenta una misma escena.

En otras cosas, el mapeo tonal sirve para lograr un registro completo de todos los niveles de brillo de la escena original en fotografías HDR (High Dynamic Range). Las imágenes HDR permiten un mejor rango dinámico de luminancias entre las zonas más claras y las más oscuras de una imagen del que las técnicas de imagen digital estándar o métodos fotográficos pueden ofrecer. En concreto, una imagen HDR contiene una mayor variación de la intensidad y del color de lo permitido por el formato RGB, que utiliza únicamente 1 byte por canal. Para almacenar esta información adicional se utiliza coma flotante de precisión simple para cada canal en vez de *unsigned char* como en RGB. Esto permite una gama extremadamente amplia de valores de intensidad.

Esto permite obtener imágenes más acordes a las visualizadas por el ojo humano, que cuando observa el mundo que nos rodea es capaz de distinguir detalles en zonas con una diferencia de iluminación muy superior a la soportada por otros formatos como la película o los formatos de imágenes comprimidas. Este rango dinámico más extenso permite a las imágenes HDR representar con más exactitud el extenso rango de niveles de intensidad encontrados en escenas reales, que van desde luz solar directa hasta la débil luz de las estrellas.

El problema es que, aunque tenemos cámaras capaces de grabar en HDR la amplia gama de intensidad que existe en el mundo real, nuestras pantallas habituales no son capaces de mostrarla. El mapeo tonal transforma las intensidades en la imagen de manera que podamos representarla y ver tanto la diferencia entre los valores más brillantes como los menos brillantes.

Por ejemplo, en la imagen base de la práctica, el interior de la iglesia con luz que entra por las vidrieras, los valores de coma flotante de entrada sin procesar de la imagen HDR están en un rango de 0 a 275. Sin embargo la mayoría son menores a 3 y la media se establece en 0.41. Esto significa que ciertas áreas (las ventanas) son muy brillantes en comparación con cualquier otro sitio. Si linealmente mapeamos el rango de entrada [0-275] en el rango RGB [0-255] que puede representar un monitor la mayoría de los valores se asignarán a 0, pudiendo ver solamente las áreas más brillantes (ej: vidrieras) apareciendo el resto en tono negro. El mapeo tonal permite que los valores más brillantes no se encuentren tan lejos de la media. Así cuando mapeamos los valores en el rango RGB [ 0-255 ] se puede observar las diferencias de brillo y se ve la imagen completa.

Hay muchas maneras de realizar este proceso y es tanto un arte como una ciencia. Una posible solución pasa por modificar la forma de representar el color.

El espacio RGB que hemos estado utilizando para representar las imágenes se puede considerar como un posible conjunto de ejes que abarcan un espacio tridimensional de color. Otra posible manera de representar una imagen en color es separar la información de color a partir de la información de brillo. Hay varios métodos diferentes para hacer esto, y uno muy común durante los días de televisión analógica era crominancia-luminancia o YUV. El modelo YUV define un espacio de color en términos de una componente de luminancia (Y) y dos componentes de crominancia (U,V). A partir de las imágenes se puede conseguir fácilmente el canal de luminancia y posteriormente realizar una compresión de su gama a [0, 1]. Para ello, se requiere la distribución acumulada de los valores de luminancia que se puede realizar en varios pasos:

1. Obtención de valores de luminancia de entrada: [2 4 3 3 1 7 4 5 7 0 9 4 3 2]
2. Cálculo de min / max / rango: 0 / 9 / 9
3. Cálculo de histograma (3 bins): [4 7 3]
4. Cálculo de la distribución acumulativa (exclusive scan): [4 11 14]

Cómo se puede observar todos los pasos se pueden realizar utilizando patrones básicos de paralelismo en CUDA y solo afecta al código en *funcHDR.cu*. Luego se puede realizar un mapeo tonal recomblando los nuevos valores de la intensidad con la información de color para formar la imagen final, lo cual viene realizado en el código de apoyo mediante la implementación de Mike's IPython notebook (basándose en la práctica de *tone mapping* de Udacity del curso *Introduction to Parallel Programming*).

### Objetivos parciales

- Encontrar el valor máximo y mínimo de luminancia de forma paralela, estableciendo el rango (diferencia entre ambos) a representar (2 puntos).
- Generar de forma paralela un histograma de todos los valores de luminancia en los diferentes *bins* existentes (2 puntos). El número de bins será un parámetro establecido y el cálculo de cada *bin* viene dado por
$$\text{bin} = (\text{Lum}[i] - \text{lumMin}) / \text{lumRange} * \text{numBins};$$
- Realizar de forma paralela un *exclusive scan* en el histograma para obtener la distribución acumulada de los valores de luminancia (2.5 puntos).
- Correcta adecuación del número de bloques y de threads por bloque para diferentes tamaños de imagen. No es necesario tener en cuenta la arquitectura concreta de la tarjeta, pero si se debe especificar las limitaciones sobre la arquitectura en concreto sobre la que se ha ejecutado (1 punto).
- Utilización de memoria compartida y cooperación entre los diferentes threads para mejorar el rendimiento (2.5 puntos).

**Nota:** las puntuaciones para cada objetivo parcial son las puntuaciones máximas que se pueden obtener si se cumplen esos objetivos. No se debe hacer un programa separado para cada objetivo, sino un único programa genérico que cumpla con todos los objetivos simultáneamente.

#### **Parte optativa (para los que tienen la asignatura completa)**

- Análisis de la utilización de librerías externas como THRUST para la realización de alguna de las funciones realizadas en el cálculo de la luminosidad mostrando sus beneficios e inconvenientes.

#### **Entrega de prácticas**

La entrega de prácticas se hará a través del Campus Virtual en las fechas anunciadas. Se debe entregar un único archivo *funcHDR.cu* con el código de toda la práctica, debidamente comentado y una memoria de la misma en formato pdf. La memoria debe incluir:

- Índice de contenidos
- Autores.
- Descripción del código: incluyendo descripción de las principales funciones implementadas
- Análisis de las mejoras introducidas, especificando tiempos relativos al uso de memoria compartida, tamaños de bloque, etc...
- Comentarios personales: incluyendo problemas encontrados, críticas constructiva, propuesta de mejoras y evaluación del tiempo dedicado.
- No incluir código fuente
- NO DESCUIDE LA CALIDAD DE LA MEMORIA DE SU PRÁCTICA. Aprobar la memoria es tan imprescindible para aprobar la práctica, como el correcto funcionamiento de la misma. Si al evaluar la memoria se considera que no alcanza el mínimo admisible, la práctica se considerará SUSPENSA.

#### **Autoría de la práctica**

La práctica se debe realizar en grupos de 2 personas como máximo.

El hecho de detectar copia en las prácticas expondrá a los alumnos a la posibilidad de una apertura de expediente disciplinario y expulsión. En caso de detectar copia, los alumnos afectados serán suspendidos en la TOTALIDAD de la asignatura. Una práctica será considerada copia en caso de que contenga la totalidad o una parte de la práctica de otro alumno. Se considerará copia en caso de que contengan:

- Archivos con la totalidad o fragmentos de código que no sean del propio autor
- Memorias con la totalidad o fragmentos de frases que no sean de los autores