Efectos de espacio de imagen

Computación Gráfica Avanzada

Ingeniería en Computación
Facultad de Ingeniería – Universidad de la República

Gabriel Madruga

Introducción

Que vamos a ver:

- Capítulo 12 de Real-Time Rendering 4ta edición
  - Técnicas de procesamiento de imágenes
  - Métodos de reproyección
  - Técnicas basadas en muestreo
    - Lens Flare and Bloom (destellos de lentes y resplandor)
    - Depth of Field (profundidad de campo)
    - Motion Blur (desenfoque de movimiento)

Motivación

En procesamiento de imágenes nos vamos a concentrar en lo que podemos hacer teniendo como entrada una imagen.

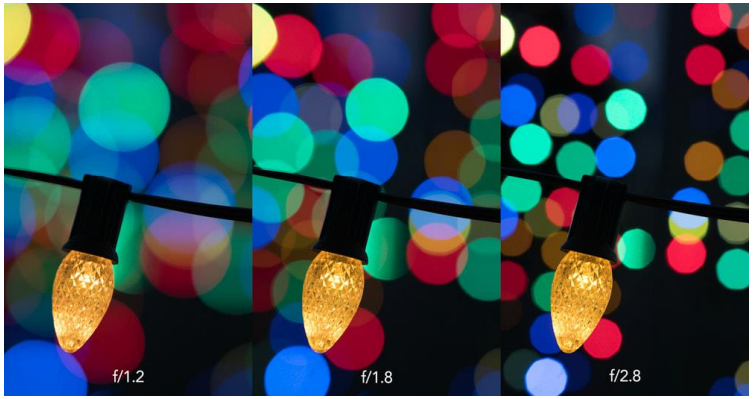
Hacer las imágenes más:

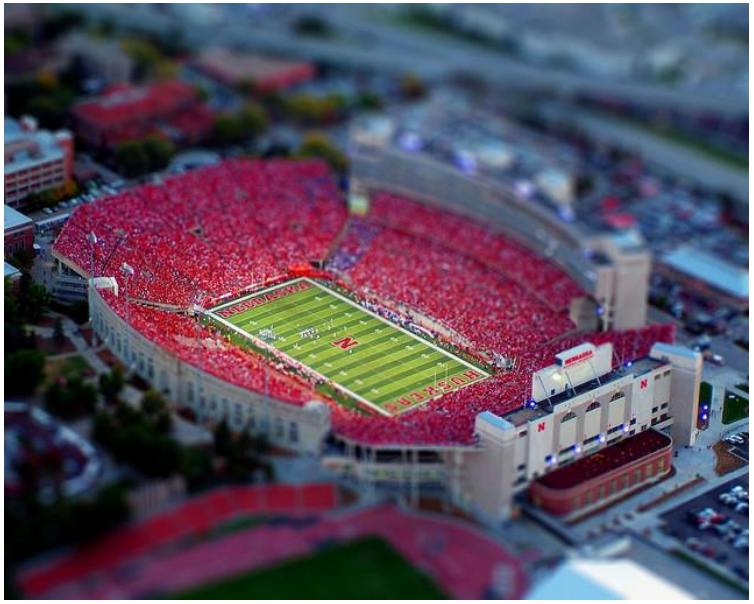
- reales, fotorealistas, o convincentes
- artísticas (i.e. mejorar la estética o comunicar)

Ejemplos

Algunos efectos:

- grano de película
- viñeteado
- destellos de lentes
- floración de luz o resplandor (bloom)
- profundidad de campo, bokeh
- desenfoque de movimiento
- Tilt-shift





Procesamiento de imágenes

Post-procesamiento: modificar la imagen después del renderizado.

Aprovechar la GPU

- Shaders programables
- Compute shaders

La mayoría de los efectos de procesamiento de imágenes se basan en recuperar la información de cada texel de la imagen en el píxel correspondiente y hacer algo con eso.

Procesamiento de imágenes

Como?

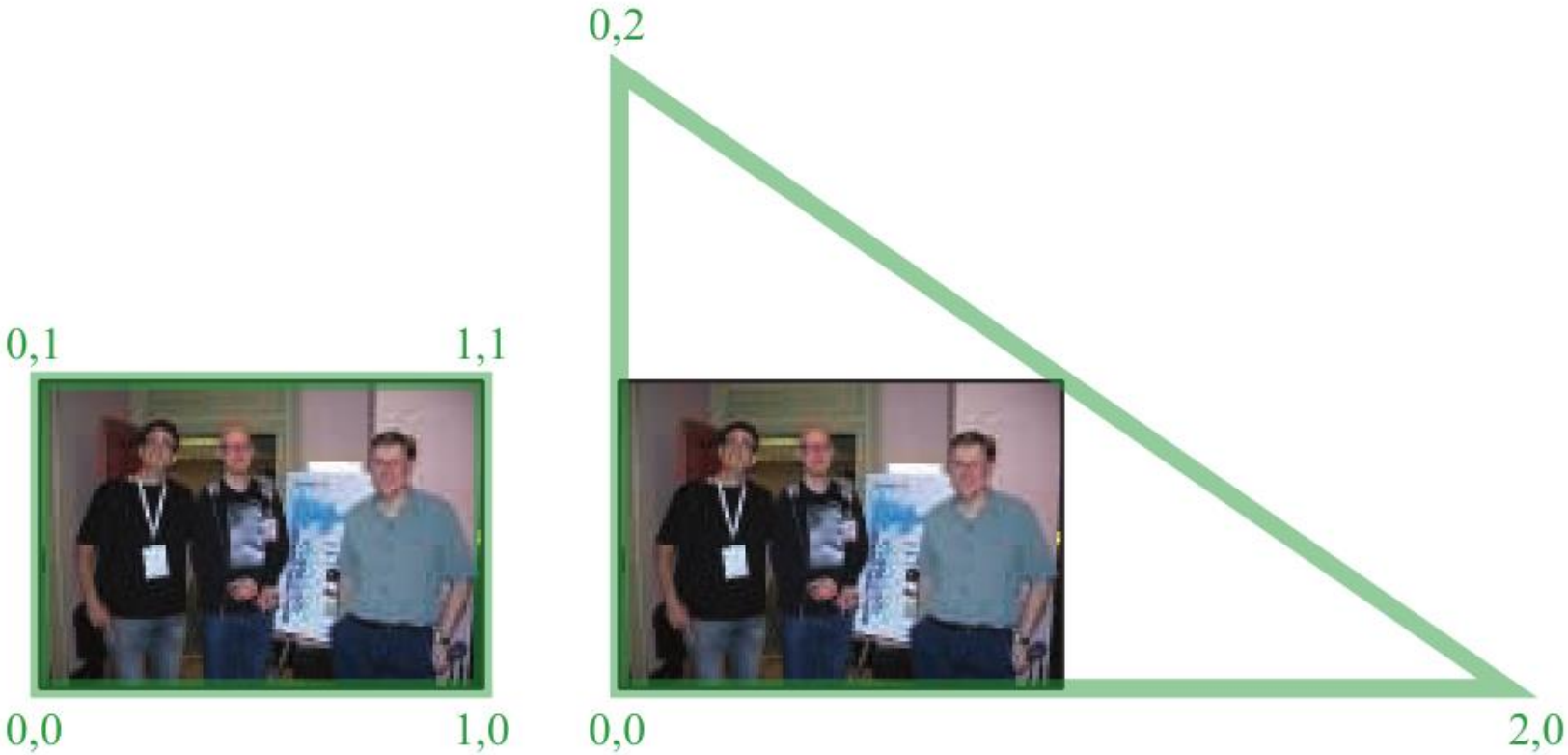
1. offscreen buffer (imagen en color o z-depth buffer)
2. se trata como textura y se aplica a un cuadrilátero (o triángulo)
3. se hace un **pase de pantalla completa** (full screen pass)
   - se renderiza para invocar al pixel shader para cada pixel

Procesamiento de imágenes

Cualquier operación de filtrado de pantalla completa intentará muestrear píxeles desde fuera de los límites de la pantalla.
Soluciones posibles:

- Ajustar al borde. Cuando se solicita un texel fuera de pantalla, se recupera el texel de borde más cercano.
  - A menudo no se notan
- Resolución ligeramente más alta que el área de visualización.
- Ignorar los faltos de información
- Duplicar los pixeles

Procesamiento de imágenes



Procesamiento de imágenes

- Hecho eso, shader de píxeles puede acceder a los datos de la imagen. Todas las muestras vecinas relevantes se recuperan y las operaciones se aplican a ellas.
- La contribución del vecino es ponderada por un valor dependiendo de su ubicación relativa del píxel que se está evaluando.
- El valor de cada texel se multiplica por su peso correspondiente y los resultados se suman, produciendo así el resultado final.

Procesamiento de imágenes

Teoría de Muestreo y Filtrado

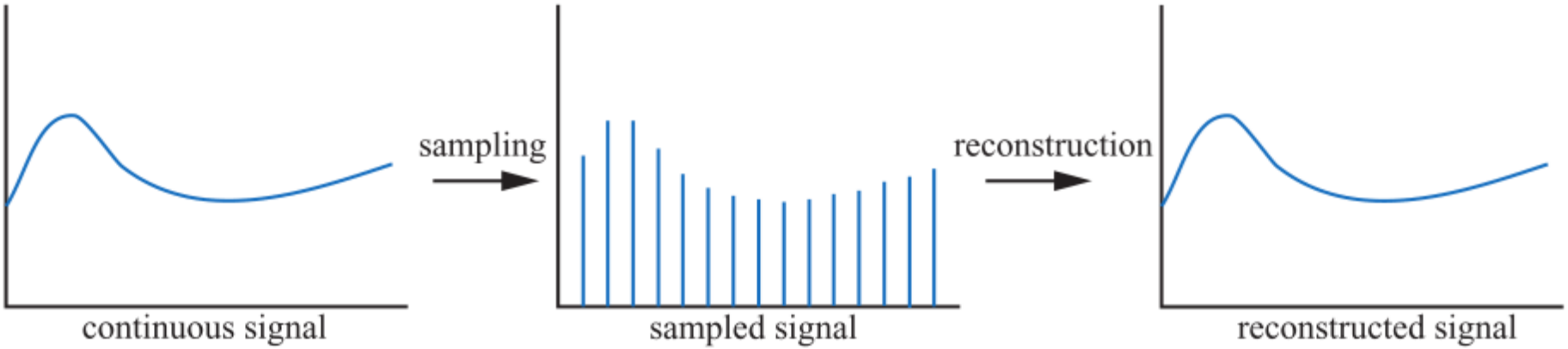
La generación de una imagen es el proceso de muestrear una escena tridimensional para obtener valores de color para cada píxel de la imagen (una matriz de píxeles discretos).

Para usar el mapeo de texturas, los texels se deben remuestrear para obtener buenos resultados en diferentes condiciones.

Para transformar una secuencia de imágenes en una animación, la animación a menudo se muestrea en intervalos de tiempo uniformes.

Procesamiento de imágenes

El objetivo de un proceso de **muestreo** es representar información digitalmente. Al hacerlo, se reduce la cantidad de información. Sin embargo, la señal muestreada debe **reconstruirse** para recuperar la señal original. Esto se hace **filtrando** la señal muestreada.



Procesamiento de imágenes

En procesamiento de imagen, un **núcleo, kernel**, matriz de **convolución** o máscara es una matriz pequeña que se utiliza para desenfoque, enfoque, realce, detección de bordes y más. Esto se logra realizando una convolución entre un núcleo y una imagen.

Más información sobre convoluciones:
http://www.songho.ca/dsp/convolution/convolution.html

Procesamiento de imágenes

Identidad	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Detección de bordes	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Procesamiento de imágenes

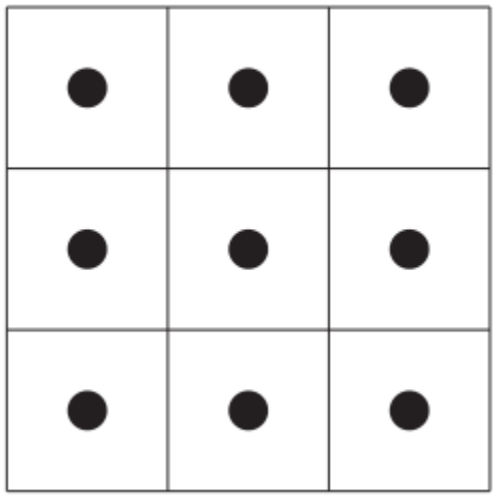
Enfocar	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Desenfoque de cuadro (normalizado)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Desenfoque gaussiano 3 × 3 (aproximación)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Desenfoque gaussiano 5 × 5 (aproximación)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

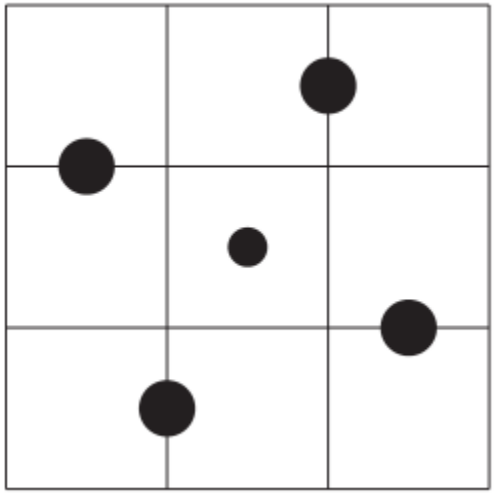
Procesamiento de imágenes

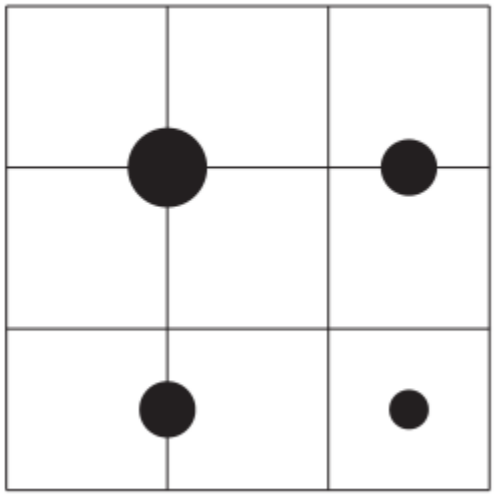
Una de las ventajas de usar la GPU es que el hardware de interpolación y mipmapping incorporado se puede usar para ayudar a minimizar la cantidad de texels a los que se accede.

Por ejemplo, digamos que el objetivo es usar un filtro de caja. Tomar el promedio de los nueve texels que forman una cuadrícula de 3 × 3 alrededor de un texel dado y mostrar este resultado borroso. Estas nueve muestras de textura serían luego ponderadas y sumadas por el shader de píxeles.

Procesamiento de imágenes







A la izquierda, se aplica un filtro de caja realizando nueve muestras de textura y promediando las contribuciones. En el medio, se utiliza un patrón simétrico de cinco muestras. A la derecha, en su lugar se utiliza un patrón de cuatro muestras. A cada muestra se le asigna un peso proporcional al número de texels que representa.

Procesamiento de imágenes

Algunos núcleos de filtrado son separables. Dos ejemplos son los filtros gaussianos y de caja.
Esto significa que se pueden aplicar en dos desenfoques unidimensionales separados.
Al hacerlo, se necesita un acceso a texel considerablemente menor que el que se necesita en general. El costo va de d^2 a 2d, donde d es el diámetro del núcleo o soporte.

Prueba matematica:
http://www.songho.ca/dsp/convolution/convolution2d_separable.html

Procesamiento de imágenes

(a)

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

(b)

0.0545	0.2442	0.4026	0.2442	0.0545
0.0545	0.2442	0.4026	0.2442	0.0545
0.0545	0.2442	0.4026	0.2442	0.0545
0.0545	0.2442	0.4026	0.2442	0.0545
0.0545	0.2442	0.4026	0.2442	0.0545

(c)

0.0545	0.0545	0.0545	0.0545	0.0545
0.2442	0.2442	0.2442	0.2442	0.2442
0.4026	0.4026	0.4026	0.4026	0.4026
0.2442	0.2442	0.2442	0.2442	0.2442
0.0545	0.0545	0.0545	0.0545	0.0545

(a) es un desenfoque gaussiano muestreando un área de 5 × 5, ponderando cada contribución y sumándolas.
(b) y (c) son dos desenfoques gaussianos unidimensionales, se realizan en serie, con el mismo resultado neto si multiplicamos los pesos.
En lugar de necesitar 25 muestras, como en (a), (b) y (c) utilizan cada una 5 por píxel, para un total de 10 muestras.

Procesamiento de imágenes

El **submuestreo** es otra técnica relacionada con la GPU que se usa comúnmente cuando se difumina.
La idea es crear una versión más pequeña de la imagen que se va a manipular. Se puede filtrar hacia abajo o crear con resolución más baja.

Disminuye considerablemente el número total de texels a los que se accede.

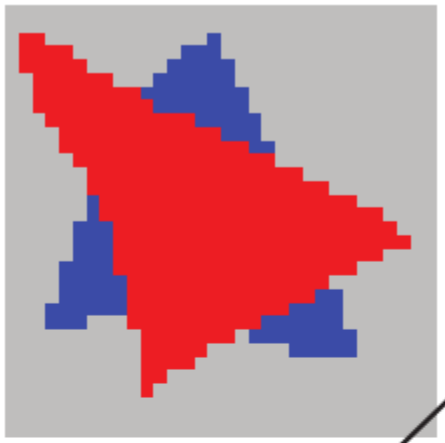
Para la mezcla con la imagen final, se usa interpolación bilineal lo que le da un efecto borroso adicional.

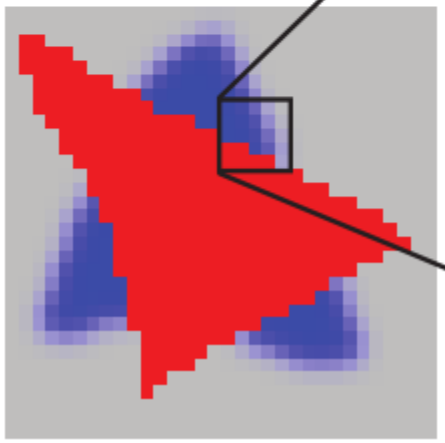
Procesamiento de imágenes

Los resultados de muestreos y otras operaciones de procesamiento de imágenes se pueden mejorar utilizando algún tipo de **filtro bilateral**. La idea es descartar o disminuir la influencia de las muestras que no parecen estar relacionadas con la superficie en nuestra muestra central.

El **filtro bilateral** es un filtro no lineal, preservador de bordes y de reducción de ruido y suavizado de imágenes.

Procesamiento de imágenes





0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
		0.0983	0.0596	0.0133
				0.0030

Procesamiento de imágenes

Una forma común de implementar un pipeline de postprocesamiento es usar buffers de **ping-pong**. La idea es aplicar operaciones entre dos búferes fuera de pantalla, cada uno usado para mantener resultados intermedios o finales.

Hacer que cada pase por separado realice un efecto particular es conveniente desde una perspectiva arquitectónica. Sin embargo, para mayor eficiencia, es mejor combinar tantos efectos como sea posible en una sola pasada

Procesamiento de imágenes

En capítulos anteriores, los shaders de píxeles que acceden a sus vecinos se utilizaron para:

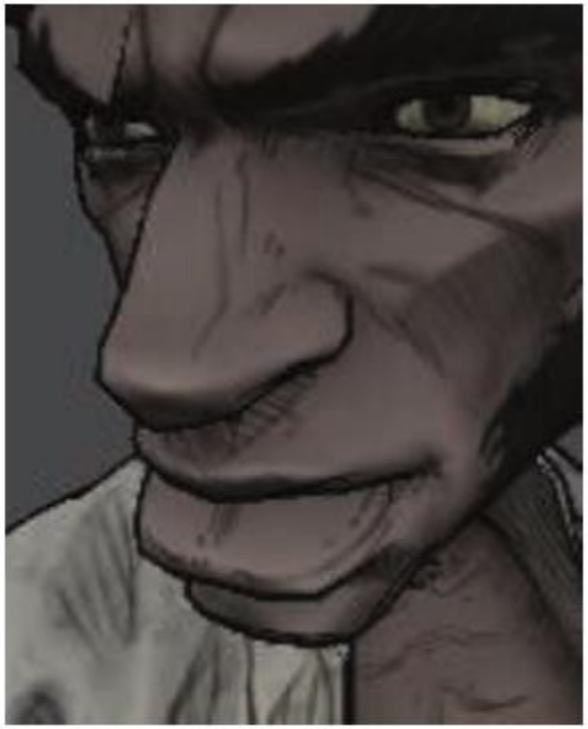
- Antialiasing morfológico,
- Sombras suaves
- Oclusión ambiental en el espacio de la pantalla
- Y otras técnicas

La sección 15.2.3 presenta algunas técnicas de procesamiento de imágenes utilizadas para la representación no fotorrealista.

Procesamiento de imágenes

Detección de bordes de Sobel modificado en el juego Borderlands.

Procesamiento de imágenes


Los bordes del contorno se generan utilizando detección de bordes con pesos variables.
Otros detalles están en la textura (arrugas)

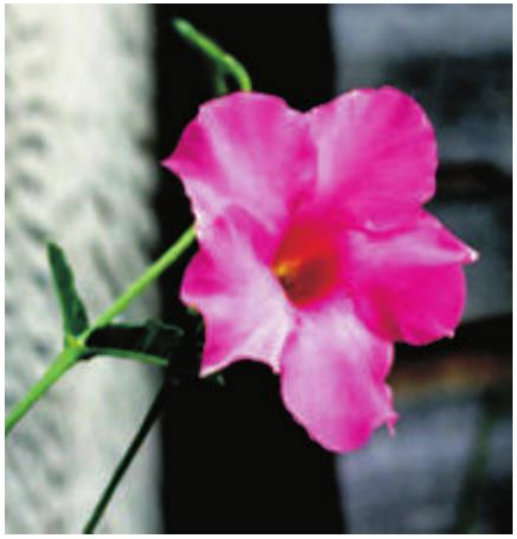
Procesamiento de imágenes

Los efectos de postprocesamiento generalmente se ejecutan en la imagen final, y pueden imitar:

- imágenes térmicas,
- reproducir el grano de la película
- aberración cromática
- realizar la detección de bordes
- generar reflejos de calor y ondulaciones
- posterizar una imagen
- ayudar a renderizar nubes y realizar una gran cantidad de otras operaciones.

Cada uno utiliza una imagen en color como la única entrada.

Procesamiento de imágenes



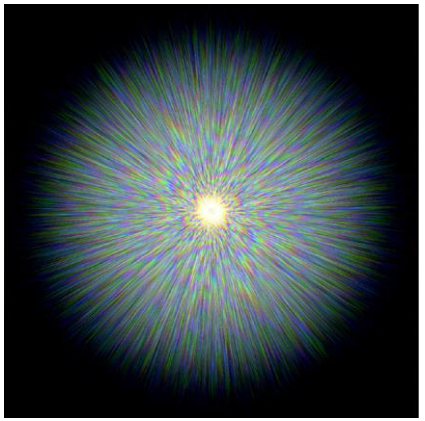
- Diferencia Gaussiana
- Detección de bordes
- Mezcla de la original con la de detección de bordes

Lens Flare and Bloom

El destello de la lente es un fenómeno causado por la luz que viaja a través de un sistema de lentes o el ojo por reflexión indirecta u otras trayectorias no deseadas.

Los destellos pueden clasificarse por varios fenómenos, de manera más significativa:

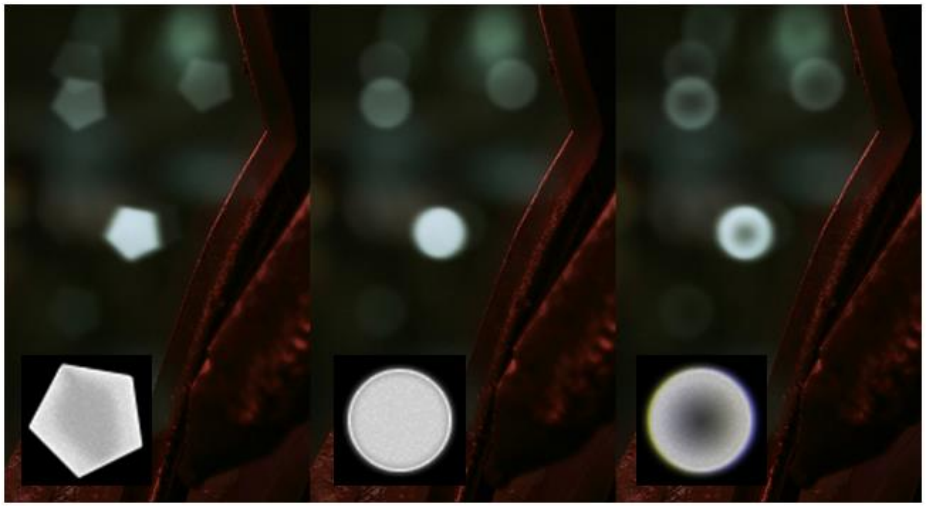
- halo
- corona ciliar



Lens Flare and Bloom

Aunque menos común con las mejoras en cámaras también pueden aparecer patrones poligonales.

Bokeh DOF Unreal Engine 4



Lens Flare and Bloom

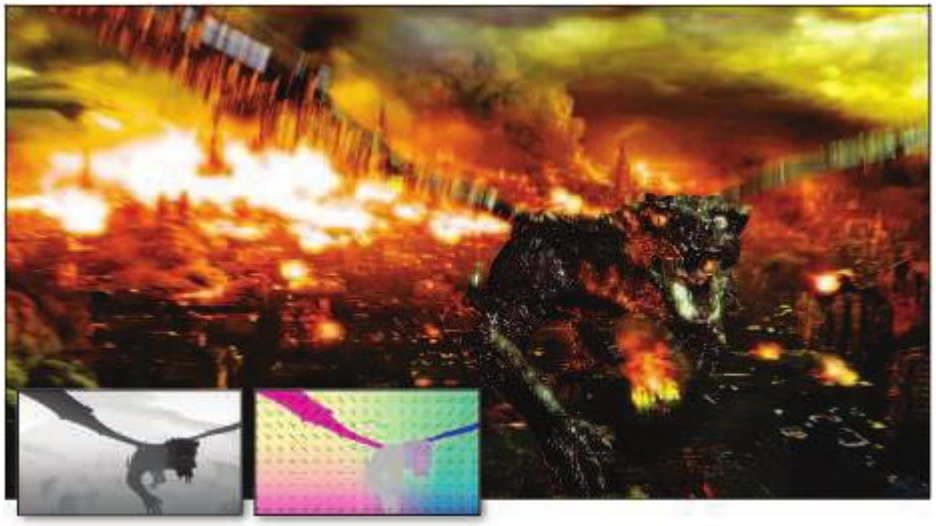
El resplandor (bloom) es causado por la dispersión en la lente y otras partes del ojo, creando un brillo alrededor de la luz y atenuando el contraste en otras partes de la escena. Se desborda el sensor a sitios vecinos.

Como clase, los halos, las coronas y la floración se llaman **efectos de deslumbramiento**.

Depth of Field

Para una cámara, hay un rango donde los objetos están enfocados, su profundidad de campo. Los objetos fuera de este rango son borrosos, mientras más afuera, más borrosos. En fotografía, este efecto borroso está relacionado con el tamaño de la abertura y la distancia focal.

Motion Blur



Métodos de reproyección

Que es?

- reutilización de muestras calculadas en cuadros anteriores
- nueva ubicación y orientación de visualización

Objetivos:

- amortizar el costo de representación en varios cuadros, explotar la coherencia temporal (e.g. suavizado temporal)
- obtener un resultado aproximado si la aplicación no puede con el marco actual a tiempo

Métodos de reproyección

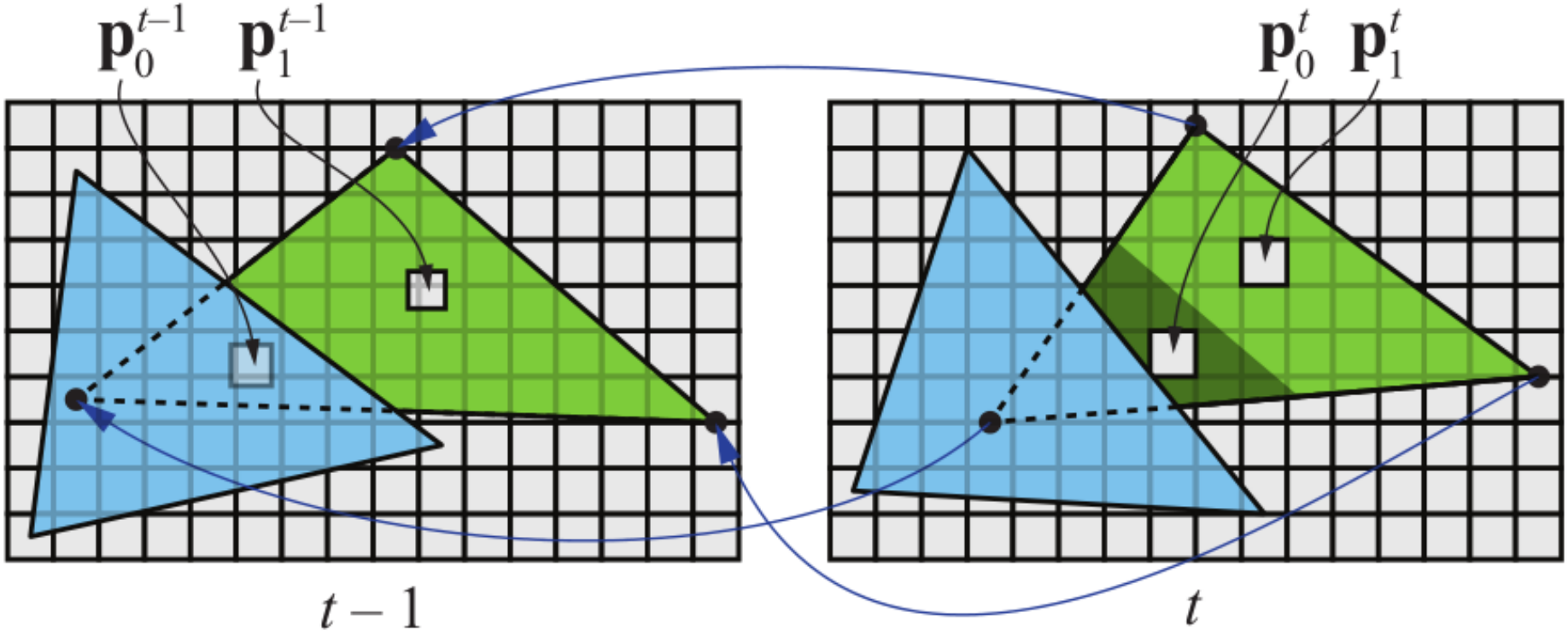
Dividido en dos tipos

- reproyección inversa (reverse)
- reproyección hacia adelante (forward)

Yang y Bowles presentan métodos de proyección de dos marcos para obtener uno intermedio.

- Al usar dos marcos tienen mejor chance de manejar las oclusiones de mejor manera
- Usado en juegos para pasar de 30 a 60 fps ya que el método ejecuta en menos de 1 ms

Métodos de reproyección



Ejemplo y por donde seguir

Hay un ejemplo que implementa DOF en WebGL:

https://github.com/tsherif/webgl2examples

FIN