Post Processing Effects

By Michael Michuki
What is Post processing?

Post Processing is the process of adding additional Effects after production to tweak the overall look and feel of a scene.

Examples of elements and effects include

- Lens Flares
- Motion blur
- Colour Grading
- Bloom (HDR blooming effect on bright objects)
- Depth of field
- Gaussian blur:
- Ambient occlusion
- Vignette
- Material effects, which are custom materials run on the final scene image.
Why do we need Post processing?

Electronic games often use various visual effects that are applied after the scene is rendered by the game engine. These are used to make the game look better or just create a particular look.

Uses:
- Emphasize on the objects in the game
- Emphasize on the environment and how it interacts with objects in a scene
- Guide the attention of the audience to a particular point.
Lens Flare
Lens Flare

- It occurs when a bright object, usually the sun, is in the shot. The light causes a glare off every piece of glass it passes through on the way to the film or optical receiver. This causes a little ghostly chain of circles, on an imaginary line from the object through the centre of the frame.

- The **Lens Flare** effect is an image-based technique that simulates the scattering of light when viewing bright objects due to imperfections in camera lenses.
When

- A lens flare is often deliberately used to invoke a sense of drama. A lens flare is also useful when added to an artificial or modified image composition because it adds a sense of realism, implying that the image is an un-edited original photograph of a "real life" scene.
Anamorphic lens flares

- This is relatively simple – light reflection on the glass-air contact surface can happen in many places in the physical lens. It can happen both before and after the anamorphic lens components. Therefore extra light transmitted and producing a lens flare will be ghosted as if the image was not-anamorphic and had regular, not squished aspect ratio.
Motion Blur

DX9 motion blur
Motion Blur

- Seen in still images and movies.
- In cameras the shutter is open for a short period of time to give light to the film media.
- Fast moving objects appear blurred.
Motion Blur

-In the human eye motion blur is perceived because the sensors in the human eye also have some reaction over time (the image fades over time). Motion blur is mostly visible in movie like frame rate (20 - 30) and on screen shots.

- Gives the illusion an object is moving faster than it really is, disguises screen tearing and low frame rates, and makes a game look like it's being filmed with a camera.
Application

- A forward moving player perceives motion blur mostly near the image corners.
- You can see the full resolution image softly blends with half resolution motion blur image.
- Note that rotating the player view results in motion blur. That is expected but might be not what the player likes to see. Normally the eye would focus on some interesting points in the image and follow them.
- For the eye the motion blur would disappear.
Application

- The gather operation is fast and blurs image content in the motion direction but without further effort image content would leak into areas that are static and are in front of the moving object (e.g. the 3rd person player).

- To fix this we mask out those cases and only blur within the moving object.
Color Grading
Colour Grading

A varied and intentional colour palette in video can inform the audience in a multitude of ways. The mood of a narrative is enhanced by the use of colour grading.
Colour Grading

They use the term to literally mean, “Correcting problems of the underlying image.” Some examples:

- Fixing exposure problems
- Fixing white balance problems
- Repairing excessive noise from aggressive ISO settings
- **Expanding contrast** from LOG- or Flat- recorded images
- “Developing” the image from RAW recordings
- Setting the initial black-, white- and gamma points
Colour Grading

What else is color grading?
After correcting the initial image problems, colorists move into the realm of the color grade. Some examples include:

**Shot matching**: Ensuring the editor’s “invisible edit” isn’t revealed by shots that look different as the timeline plays down

**Removing distractions**: Isolating and manipulating annoying elements that prevent shots from matching each other

**Controlling the viewer’s eye**: Using shape masks (or other techniques), attracting the eye to the focal point of interest

**Creating looks**: Stylizing an image to indicate a flashback, dream sequence, or re-creation—or simply to give the entire project a unique feel

*The LUT texture to use as a lookup table for color correction.*
Colour Grading
The Tone Mapping

- Map the wide range of HDR (high dynamic range) colours into the small LDR (low dynamic range) so a monitor can display the colour.

- Done after normal rendering during post processing.

- A global tone mapper is a function with three inputs (RGB) and three outputs (RGB).

- A local tone mapper also takes into account the neighbourhood of the pixel but is much more computationally intensive (means slower).

- A good tone mapper function tries to preserve the colour of a pixel even if the colour is very bright.

\[ \text{GammaColor} = \frac{\text{LinearColor}}{\text{LinearColor} + 0.187} \times 1.035; \]
Colour Correction/Colour grading (Same)
Antialiasing
Antialiasing

Anti-aliasing (AA) aims to cut down on the pixellated, jagged edges you see in the game. It's one of the more popular graphics tweaks you'll find, and helps smooth out objects when you can't increase the resolution any further.

How It Affects Performance: Anti-aliasing can affect your performance pretty significantly, just like raising the resolution of your game.
Types of Antialiasing

No AA  FXAA  MSAA  32x CSAA
SSAA (also known as FSAA): Super sampling anti-aliasing was the first type of anti-aliasing available. It's useful on photorealistic images, but isn't very common in games anymore, because it uses so much processing power.
**MSAA**

**MSAA**: Multi-sample anti-aliasing is one of the more common types of anti-aliasing available in modern games. It only smooths out the edges of polygons, not anything else—which cuts down on processing power compared to SSAA, but doesn't solve pixelated textures. (MSAA still uses quite a bit of power, though.)
CSAA and EQAA: These types of anti-aliasing (used by newer NVIDIA and AMD cards, respectively) are similar to MSAA, but at a fraction of the performance cost.
**FXAA**

**FXAA:** Fast approximate anti-aliasing has a very small performance cost, and smooths out edges in all parts of the image. However, it usually makes the image look blurry, which means it isn't ideal if you want crisp graphics.
TXAA: Temporal anti-aliasing only works on certain newer graphics cards, but combines lots of different techniques to smooth out edges. It's better than FXAA, but still has some blurriness to it, and uses a bit more processing power.
Bloom

No Bloom

Bloom
Bloom

- Is a real world light phenomena that can greatly add to the perceived realism of a rendered image at a moderate render performance cost.

- Bloom can be seen by the naked eye when looking at very bright objects that are on a much darker background.

- Even brighter objects also cause other effects (streaks, lens flares), but those are not covered by the classic bloom effect.
Bloom

- Because our displays (e.g. TV, TFT, ...) usually do not support HDR (high dynamic range), we cannot really render very bright objects.

- Instead we simulate the effects that happen in the eye (retina subsurface scattering), when light hits the film (film subsurface scattering), or in front of the camera (milky glass filter).

– The effect might not always be physically correct but it can help to hint the relative brightness of objects or add realism to the LDR (low dynamic range) image that is shown on the screen.
Bloom

**Left top:** Without bloom the brightness of the object is not shown well.

**Right top:** A single Gaussian works well for the left object but not so well for the right one.

**Left bottom:** Using two Gaussians works well on both objects.

**Right bottom:** Using three Gaussians also adds a subtle nice large scale glow.
Bloom
Depth of Field
Depth of Field

- Depth of Field (DoF) applies a blur to the scene based on distance in front of or behind a focal point.

- Near or far objects blur away, depending on where you're focusing.

- This simulates what happens in real-world cameras.

- The effect can be used to draw the viewer's attention and to make the rendering appear more like a photograph or like a movie.

- The first is to reinforce the illusion of depth in the scene you're watching.
You are Hurt. Get to Cover!
Vignette

- Reducing the brightness or saturation of the corners while emphasizing and drawing interest to the center of the image. This is starting to replace bloom as the most common game effect.

- It is purposefully added in post-processing in order to draw the viewer’s eye away from the distractions in the corner, towards the center of the image.

- Depending on the type and cause of vignetting, it can be gradual or abrupt.
Ambient Occlusion
Ambient Occlusion - Details

- Originally developed by Industrial Light and Magic (ILM) and used in the 2001 film Pearl Harbor.

- Ambient occlusion is a lighting model that calculates the brightness of a pixel in relation to nearby objects in the scene.

- More specifically, it determines when certain pixels are blocked from the environmental light by nearby geometry, in which case, its brightness value is reduced. It accounts for the general dimming effect when two evenly lit objects are brought close to each other.
Ambient Occlusion - Details

- The Stanford dragon below is rendered in an evenly lit environment. There are a few darker and lighter areas on the model, but the lighting is mostly uniform. Despite having fairly intricate geometry, the dragon appears flat and without clear depth perception.

- Ambient occlusion can be thought of as an approximate form of global illumination (GI). GI calculates the color of each pixel based on the light contributed from the surrounding hemisphere.
Material Effects
Material Effects

- The Material Effect is used to insert a Material into the post processing chain. It references a Material whose Emissive output is passed on to the next node in the chain as the rendered scene.

- The Material referenced in a Material Effect will usually sample the current rendered scene and then modify it in some way, passing the result on through its Emissive channel.

- Materials used in post processes must use the MLM_Unlit Lighting Model and in order to sample the scene texture or scene depth the material must use a transparent Blend Mode (Translucent, Additive, Modulate, AlphaComposite).
Scene Texture Expression

- Instead of using a Texture Sample as the primary render source for the material, the Scene Texture expression is used when creating a Material for use in the post processing system.

- This expression is used in exactly the same manner as any Texture Sample except that it outputs the currently rendered scene instead of a regular texture.

- This is visualized in the image below with an added depth-based desaturation to show the output of the Scene Texture expression is being modified by the material.
Scene Depth Expression

- The Scene Depth expression in the Material Editor can be used to sample the scene depth texture.

- Essentially, this expression outputs the depth from the camera to the geometry in the world at the current pixel.

- Scene Depth expression requires additional math to be performed to scale and bias the depth results to the value range desired. For example, dividing the Scene Depth by a value (Scene Depth / 1024) will provide a normalized distance value that increases from 0 to 1 as the depth goes from 0 to 1024.
Scene Tinting

- The Scene Depth expression in the Material Editor can be used to sample the scene depth texture.

- Essentially, this expression outputs the depth from the camera to the geometry in the world at the current pixel.

- Scene Depth expression requires additional math to be performed to scale and bias the depth results to the value range desired. For example, dividing the Scene Depth by a value (Scene Depth / 1024) will provide a normalized distance value that increases from 0 to 1 as the depth goes from 0 to 1024.
References


