Zeta Shader

Render of the Strike Gundam using the Zeta Shader

I. Introduction

This shader was created in order to reproduce the typical shading of metal in mecha anime in realtime.

Hizack from the opening of Zeta Gundam. Note the color variation on his shoulder.

This effect was inspired by Transformers Devastation from Platinium Games. This game uses a similar shader, called the Wakame Shader to render faithfully the style from those anime.

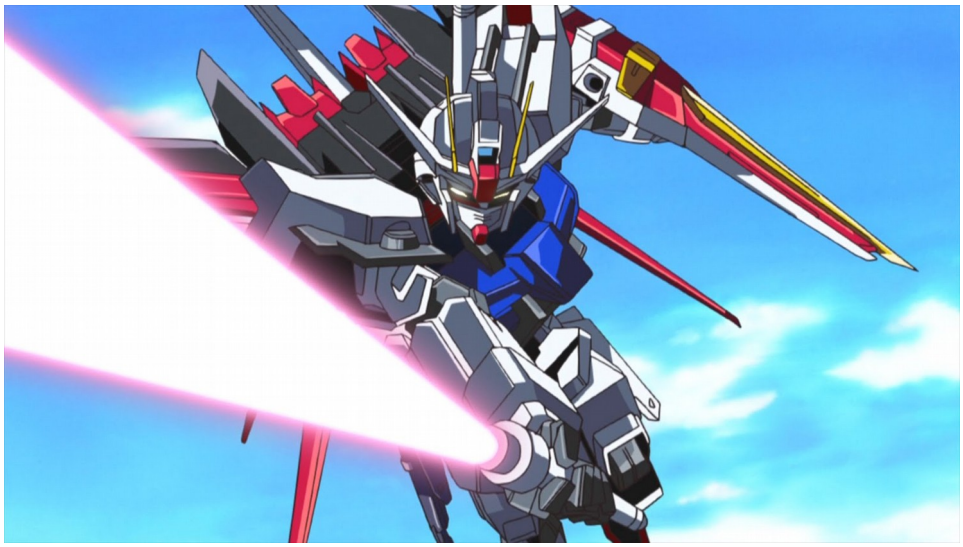
Transformers Devastation has pretty much nailed the style of shading of the Transformers Series

This shader was created in response to the Wakame shader in order to address its shortcomings. It works using several textures, and choses which one to show depending on the light. This brings two flaws :

- The lighting doesn't change much with motion and lightsource movment, because all the details are prerendered

- Creating those textures requires a lot of work

The Zeta Shader instead requires no additional effort and reacts well with motion, at the cost of a less fidelity to the old style of anime. However, it provides higher fidelity to the modern style of mecha anime.



The Strike Gundam as depicted in Gundam Seed

The Strike Gundam rendered using the Zeta Shader

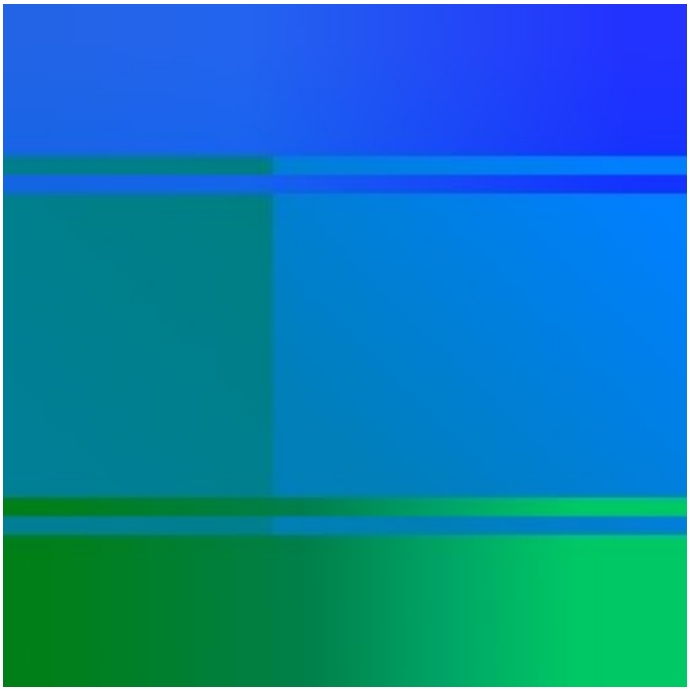
II. Internal Workings

The Zeta Shader is an extended toon shader, based on the Phong Shader found in Gratin.

Due to the nature of mecha designs, it works best with models using sculpted detail and simple textures.

It uses a two dimensional toon ramp. The X-axis is mapped to the diffuse part, and the Y-Axis to the specular one.

As such, it is important to have a wide specular for the shader to work properly



The Toon Ramp used in this instance

The toon ramp uses all channels aviable in order to convey more information :

Red – Bloom :

Allows for a shinier look in bright spots

Green – Saturation :

This makes possible to adjust the saturation of the final color. This allows more highlights, without making the rest of the picture dull. (0 → x0 ; 127 → x1 ; 255 → x2)

Blue – Intensity :

The basic toon ramp, becomes the new diffuse component.

This is the basis of the shader, and one of the most important parts. The saturation variable makes it possible to have “whiter” zones for lighted parts, and vibrant colors everywhere else.

Application of the Toon ramp on the Strike Gundam (no Bloom)

This is one of the fastest parts of the rendering pipeline. Sticking only to this allows for an already good result, as the specular tends to naturally form around edges and enhance them.

However, for a better look, we want to add solid edges to the final image.

In order to do this, we use several edge detection methods :

- Z Buffer difference. This first one is set to find only big differences in depth, in order to find only the silhouette of the robot. It is then dilated to add attention to the global shape of the robot.

- Z Buffer difference. This second one is set to find small differences in depth. This allows for a better representation of the fine details.

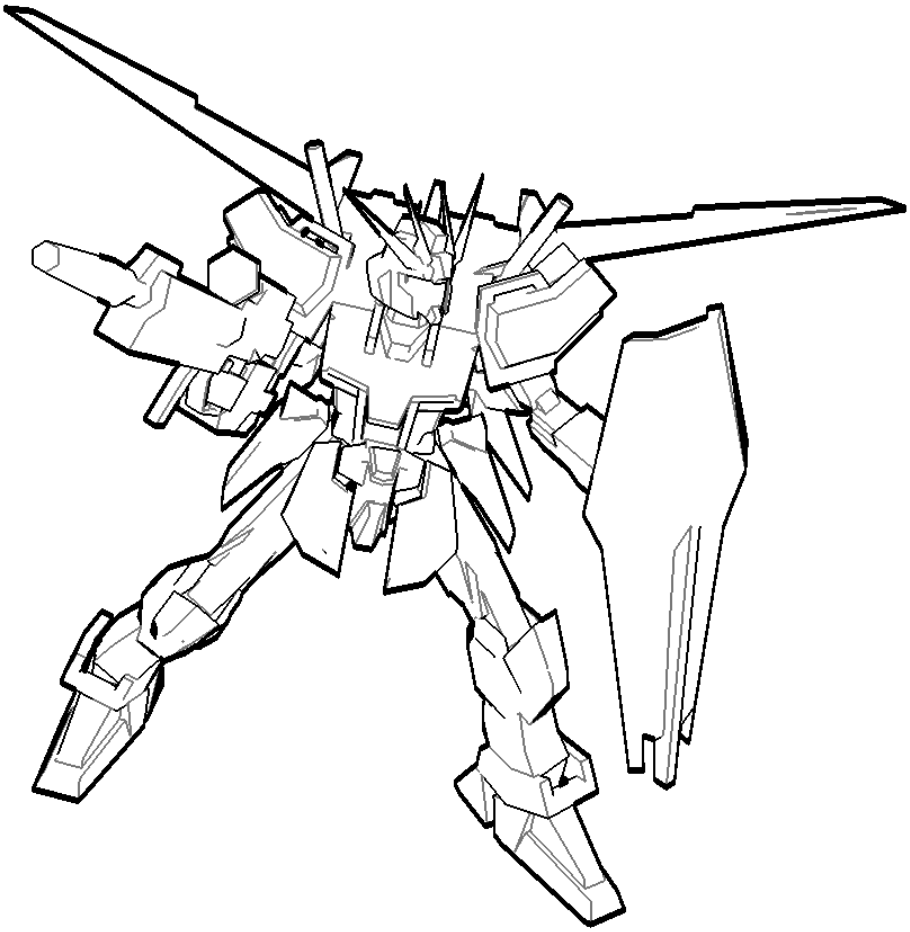
- Normal Buffer difference. This last one allows us to enhance the small sculpted details. It is overwhelming, so we reduce its impact prior to composition.

This capitalizes on the fact that most detail on mecha doesn't come from the texture, but from the shape. So, in order to make it look good, the edges need to be considered carefully, as a better appreciation of the sculpted detail will make for an overall better picture.

We then combine those images and put them on top of the first render.

It is possible (though not implemented in this version) to accelerate this step by combining normal and Z Buffer info in one output (likely rgb for the normals and alpha for depth) and find out all the edges at once.

From left to right : Z Buffer (Silhouette) ; Z Buffer (Fine) ; Normal Buffer



Combining the edges gives this images, which exposes a lot of the details present on the model

At this point in time, the shader is very faithful to modern mecha anime, while being fast to compute and requiring very little additional work on models, making it suitable for video games and other realtime applications.

The next and final step is the most costly but can really enhance the final image. It is bloom.

Bloom is setup through an illumination map unique to the 3D model and through red channel of the toon ramp.

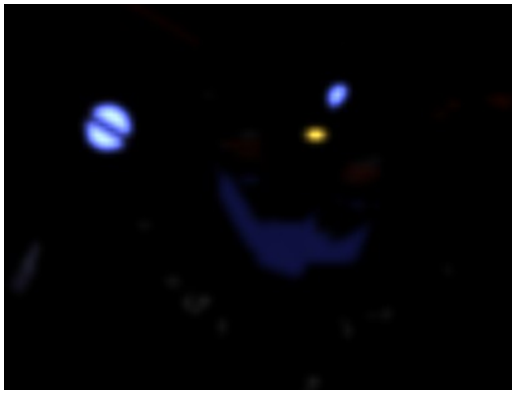
In this version of the shader, it corresponds to the blue channel of the fxTexture. It is possible to

dedicate the whole texture to bloom, allowing for better control of the color (for instance, a white eye blooming green, as used later for the beam effect).

Illumination is put in a separate buffer with the color of the underlying texture. We add this buffer to the original image, allowing for an illumination effect even without using bloom.

A portion of the illumination buffer. You can faintly see a blue trace on where would be the torso.

It is then reduced in size (for faster computation), blurred, multiplied (in order to show it more) and scaled back to original size.



The Bloom effect to be added to the final scene

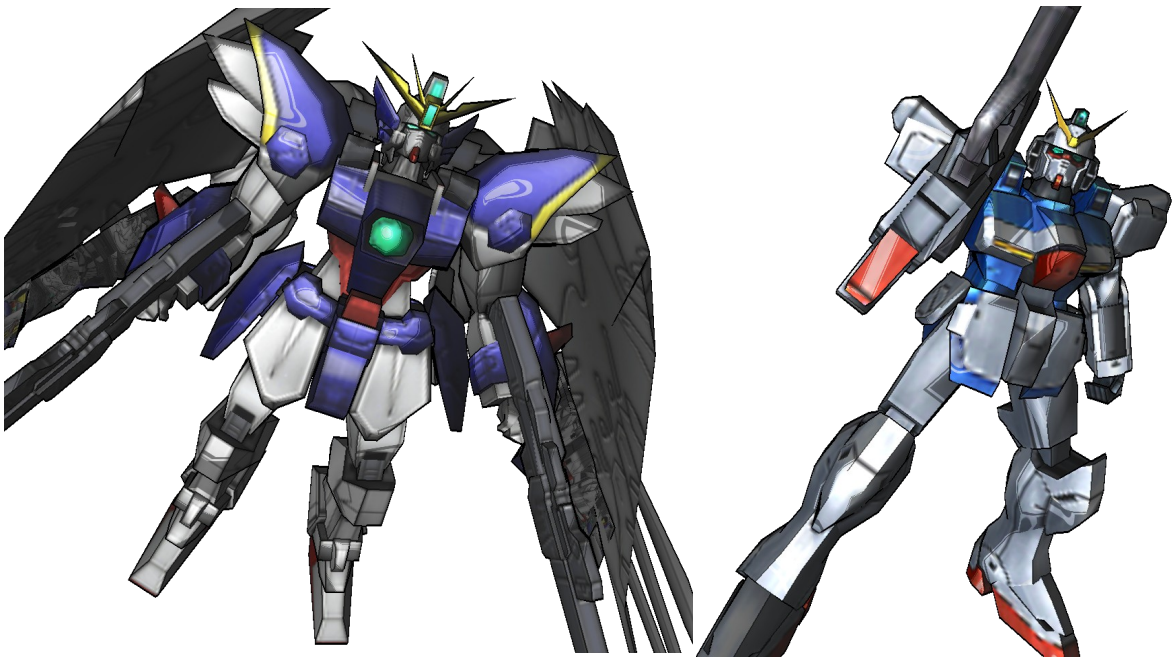
We then add this to the render from last step

The bloom effect in the final render.

III. Limitations & Adjustements

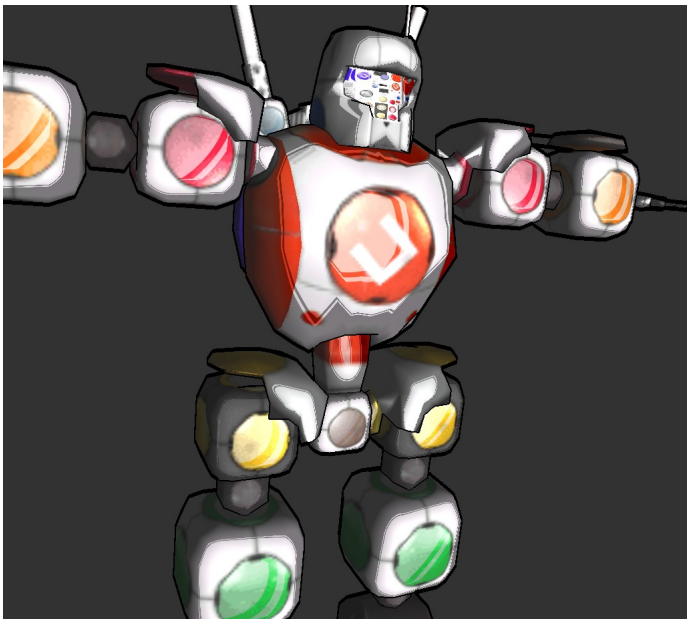
As stated earlier, the Zeta shader works best with models with simple textures and sculpted details.

Below are some suboptimal models rendered with the Zeta Shader.



The Wing Zero Custom and the Victory Gundam rendered with the Zeta Shader.

These models have lighting precomputed on the texture, so the shader itself is less visible. The Strike Gundam texture has been simplified manually.

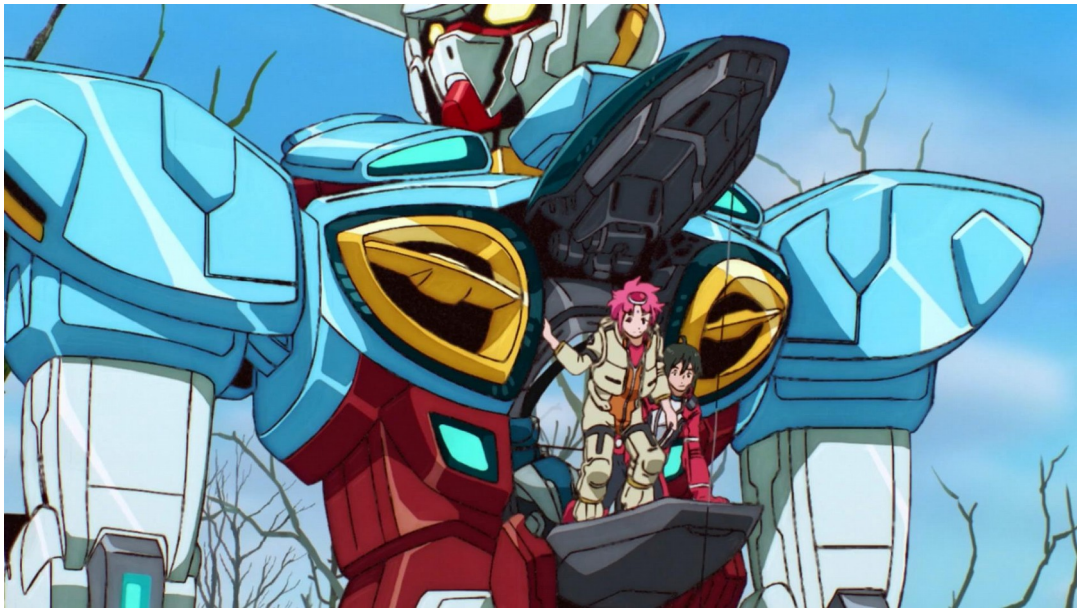


Cubix rendered with the Zeta Shader. The simple texture showcases the capability of the shader, but the lack of sculpted detail doesn't show the edge detection

While the Zeta Shader works well out of the box, some adjustments may be required depending on the scene.

The current toon ramp works pretty well for space scenes, but not so well for in-atmosphere ones.

Picture of the Gundam Barbatos from Gundam Iron Blooded Orphans (2015/2016). We can see that the output is similar to the one of the Zeta Shader



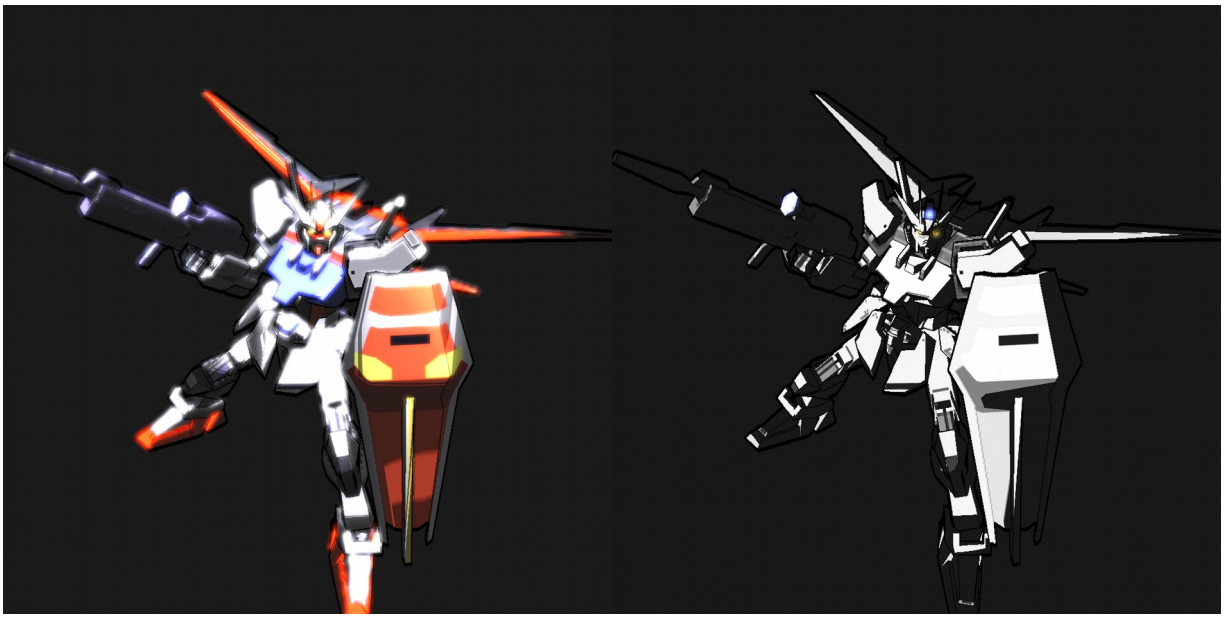
The Gundam G-Self from G no Reconguista (2014/2015). We can see that there is less contrast and more highlights than is the space scene.

The toon map supplied with the Zeta shader works well with shiny robots (due to the mix between the ramps) in a space-like environment, like in a space colony.

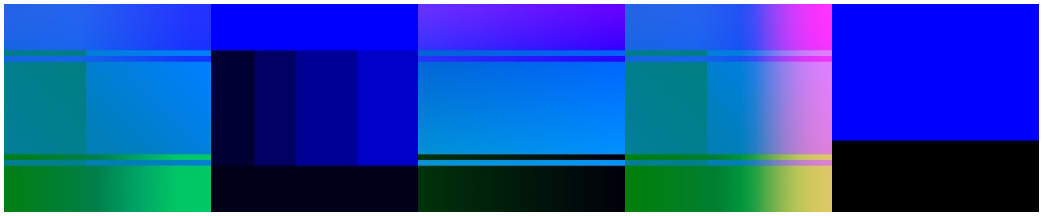
Adjustments can be made to get closer to the feel wanted in a scene. For a space scene, the ramp should be weighted more heavily towards low specular regions, and have a bigger contrast. In atmosphere scenes, the ramp should be balanced, without a lot of contrast. This supposes a single lighting source, though several ones will yield a similar result.

You can adjust the toon ramp for more specific ambiances. (Ramps below in order)

Grayscale Ramp ; High Contrast Ramp



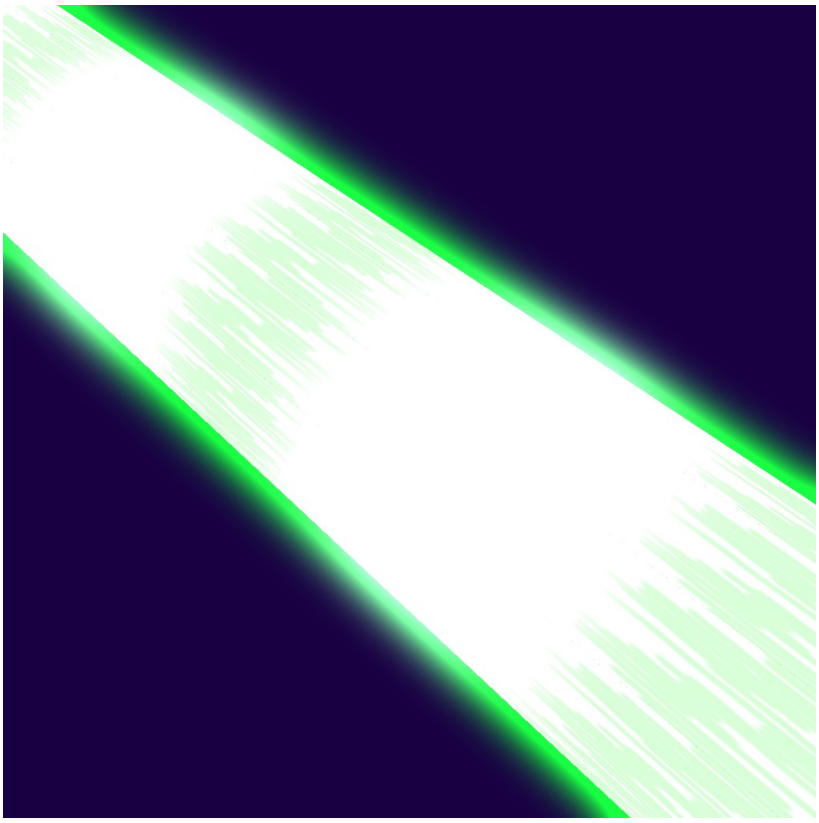
Shiny Ramp ; Black & White Ramp



IV. Beam Shader

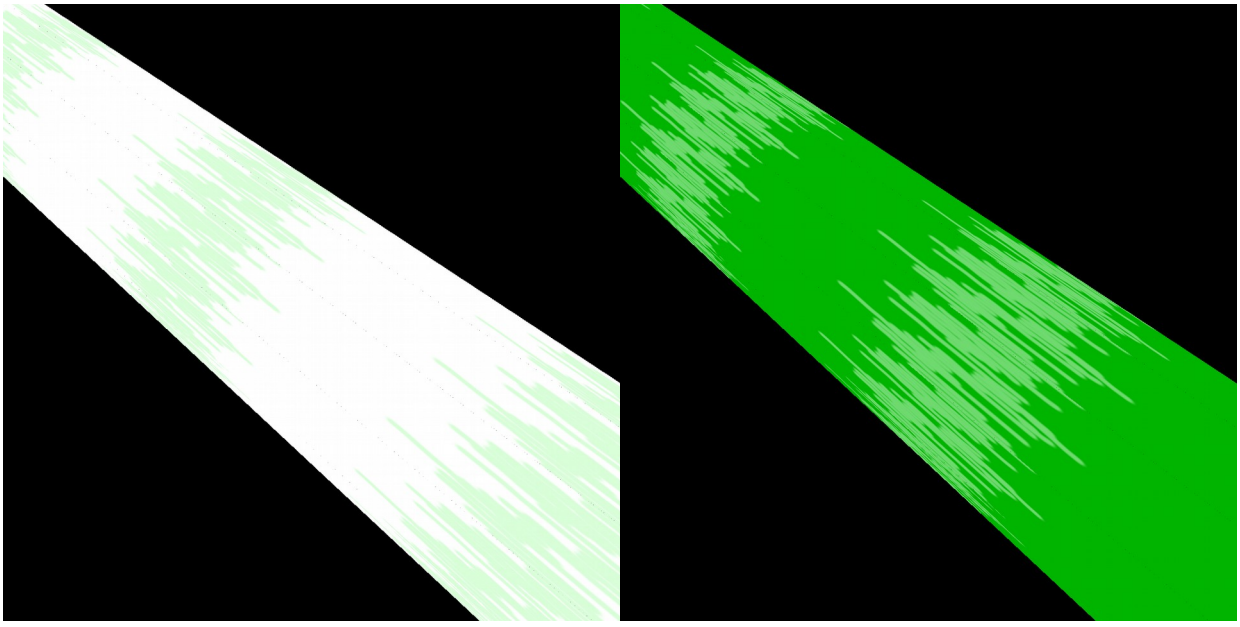
One of the recurring themes in mecha anime are laser/plasma beams.

In order to complete the Zeta Shader, here is a shader for beams (as in fired from a beam rifle), that can also be used for effects such as propulsion, explosions and beam sabers.



A beam rendered with the beam shader. It is animated (see it in Gratin)

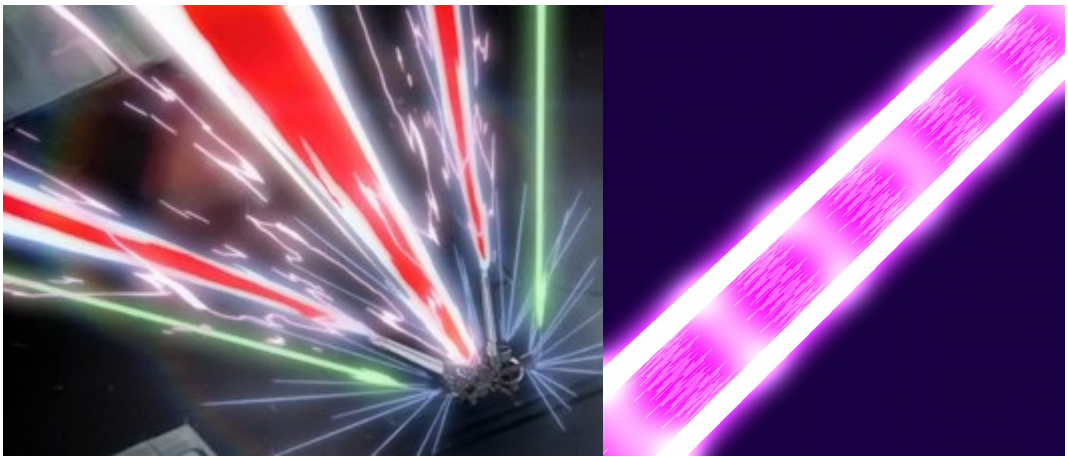
It is made with two textures, one for the base of the beam, and one for bloom.



Left : Base ; Right : Bloom (before blur)

Another type of beams are beams with one color outside and one inside. In order to reproduce this, we use a 3D model composed of two cylinders : one inside, and one bigger outside with its normals flipped.

Using backface culling, the smaller one will show through. Gratin doesn't support this so the 3D model has been cut in order to show the effect.



Left : Meteor firing in Gundam Seed ; Right : Beam shader reproducing a similar beam

Propulsion of the Star Build Strike. It is very similar to the beam effect, and should pose no problem when trying to reproduce it.

Left : Fenice Rinascatta posing with its beam saber in front of an explosion. The beam saber and yellow part of the explosion car be reproduced with the beam shader. Right : Gundam X, using a strange-looking beam saber, reproducible using an appropriately shaped model

Beam sabers can be reproduced out of the box, but should be coupled with a particle emitter for additional meaningful effects, like an energy pulsion or a trail following the saber's movments.

A trail left behind a beam saber slash. This is a common effect to suggest motion, and can be made using particle emitters at the end of the beam.

Left : The ∀ Gundam's beam saber Right : The G-Self's beam saber.

Thin beam sabers are more accurately made using a particle emitter, as it allows for better reproduction of the fluctuation present in their respective anime.

V. Future Work

This work can be expanded upon in different ways.

First of all, simply making different toon ramps for different environments would allow the shader to accurately reproduce more situations.

Implementing the shader outside of Gratin, for instance in a game engine, or in Blender, can allow us to use it in motion, and implement the particle beam saber effects.

Other minor adjustments can be made for better animation, like reflexions (even though they aren't direct usually in anime, and are simplified before being reflected), and adjusting it to be used on characters.

More control over the final render could be useful, but will require more work on the part of the artists in the end. It has been partially implemented in the green channel of the fxMap.

For that purpose, a tool could be made to allow artists to “paint” on the toon ramp from a normal render, using the rampChosen buffer in Gratin. This would allow artists to design ramps more easily by using methods familiar to them.

Some optimizations can always be made, like combining all the edge detectors together to divide by 3 the texture access needed.

VI. References

Images used (except for *Transformers Devastation* one) are from different Gundam series, belonging to Sunrise.

Transformers Devastation has been made by Platinum Games.

Models found on Models Resource, from the game *Kidou Senshi Gundam: Gundam vs. Gundam NEXT PLUS* on the PSP. Cubix model from the game *Cubix Robots for Everyone : Showdown* on the Gamecube.

Glow made using the method in this article :
http://www.gamasutra.com/view/feature/130520/realtime_glow.php

Shader made during an internship at INRIA in Bordeaux.

Author : panagiotis.tsiapkolis@etu.u-bordeaux.fr

Responsable de Stage : Pierre Bénard (http://www.labri.fr/perso/pbenard/)

Gratin is made by the Manao Team (INRIA)