

SEP # : 16

Title: anti-aliasing

Version: 1.0

Author: JB Silvy

Review:

Commented:

State: Draft

Scilab-Version: 5.2

Vote:

Created: Tuesday, 17 February 2009

Abstract

In this SEP, we discuss the introduction of full screen anti-aliasing (FSAA) to enhance Scilab Graphics image quality. When used in Scilab, anti-aliasing would allow to remove staircase artifacts that appears in Scilab Graphics, especially for polylines. For more information on anti-aliasing in general, see [1].

Background

All Scilab versions until Scilab 5.1 are not able to activate full screen anti-aliasing. However, most of software having visualization capabilities use it to enhance their graphics quality. They have consequently an advantage upon us. Matlab does not have such a feature by default but some toolboxes [2] provide it. Anti-aliasing has also been requested in our bug data base [3].

Rationale

FSAA enhances image quality but also decreases graphic performances. When FSAA is activated, it also comes with a quality setting. The higher the setting, the better the quality, but also the lower are the performances. Practically, there are nowadays 4 standard settings which are “2x”, “4x”, “8x” and “16x”. The “4x” and “8x” are the more commonly used settings. With “2x”, the difference with no anti-aliasing is barely noticeable. The difference between “8x” and “16x” is also barely noticeable. Consequently, “4x” represent a good compromise between speed and quality whereas “8x” stands for the best quality.

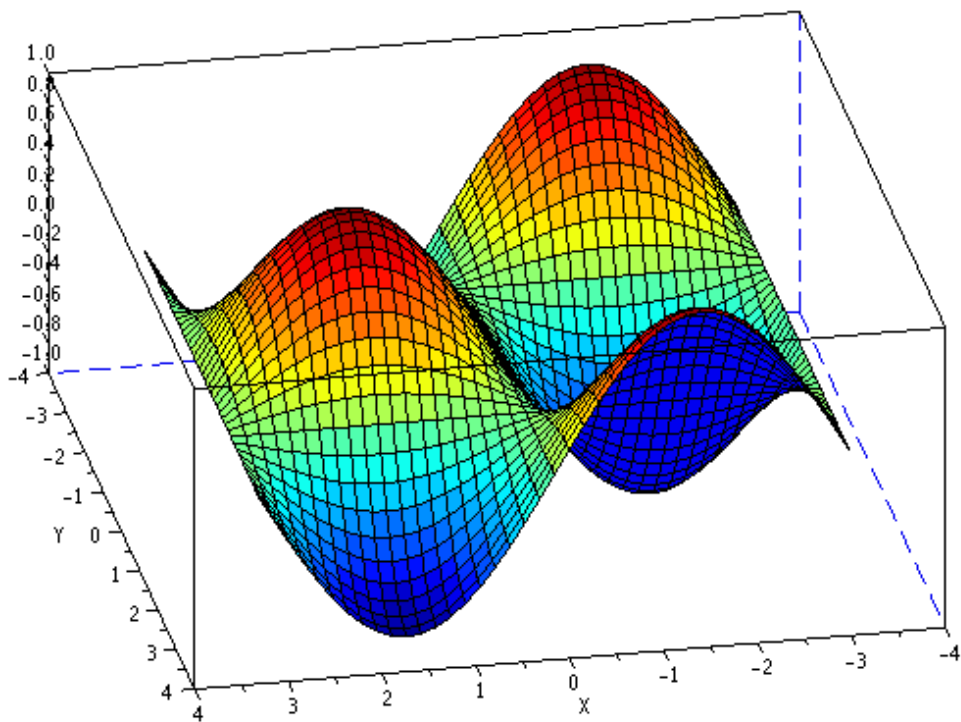


Figure 1: Result of plot3d() in Scilab without anti-aliasing.

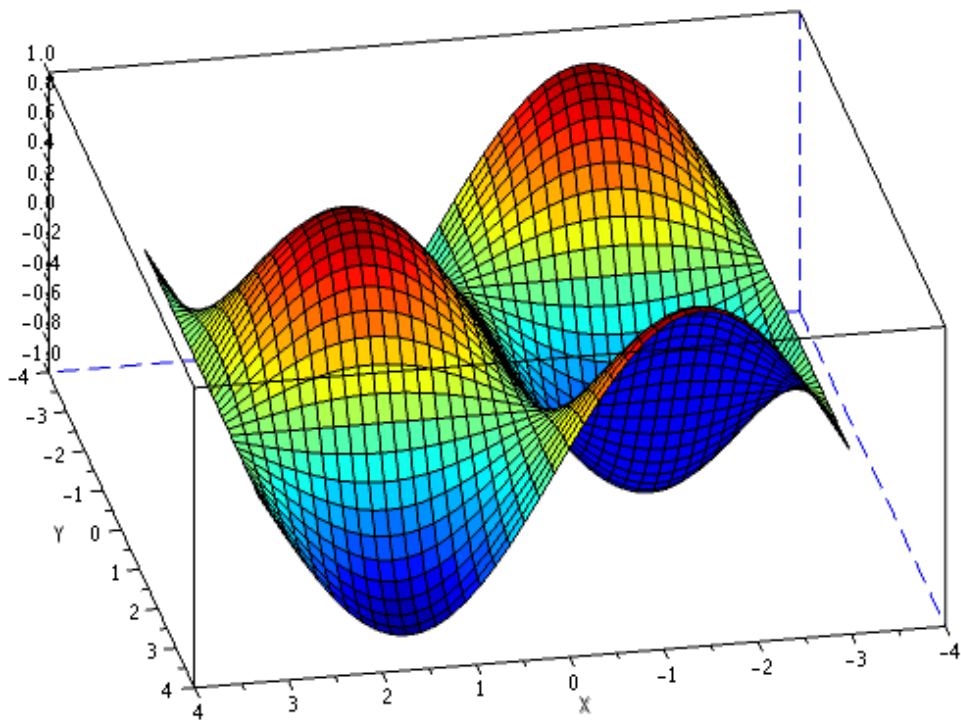


Figure 2 : Result of plot3d() in Scilab with anti-aliasing 4x.

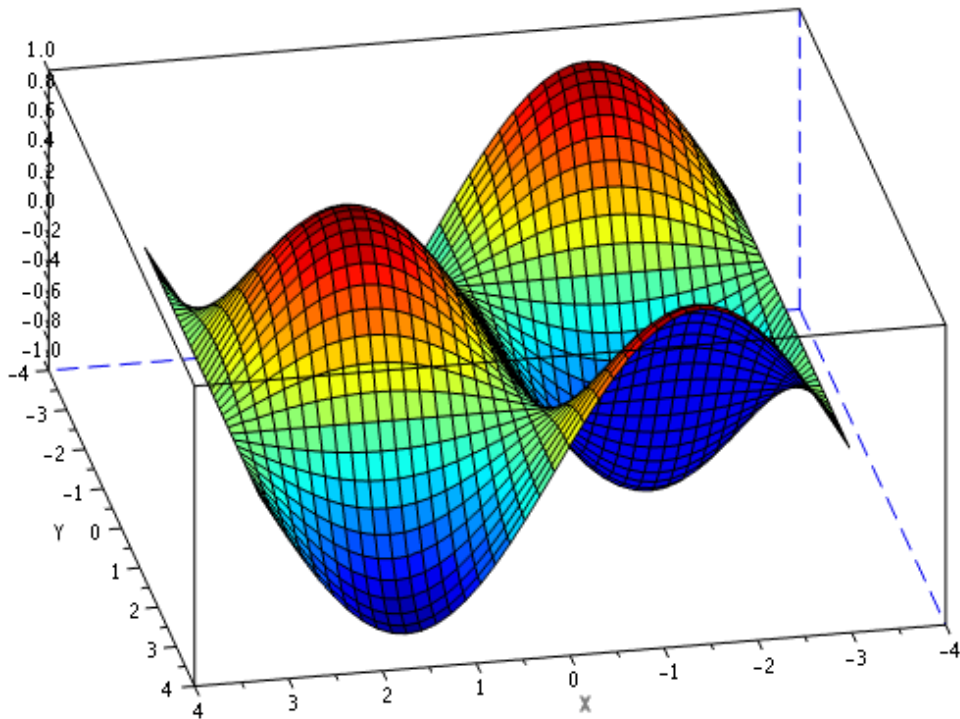


Figure 3 : Result of plot3d() in Scilab with anti-aliasing 8x.

From Scilab, the anti-aliasing will be controlled per figure. A property “anti_aliasing” will be added to figure handles. This property value can be one of the following: “off”, “2x”, “4x”, “8x”, “16x”. If the property is set to “off”, anti-aliasing is disabled. Otherwise, the value stands for the anti-aliasing quality.

An other approach would be to have only two or three possible values, more understandable by the user. The value could be chosen between “off” or “on” (actually setting quality to “4x”) or between “off”, “performance” (quality “4x”) and “quality” (quality “8x”).

By default, the anti_aliasing will be turned off to avoid speed loss in already existing Scilab codes (such as Scicos). However, the default can be changed by modifying the default figure.

Limitations

As any OpenGL/JOGL development, this feature should be widely tested on different architectures. This will avoid bad surprises on low end computers.

Moreover, anti-aliasing might not be supported by a few graphic configurations. In this case a warning will be send to the user that anti-aliasing won't work on his configuration.

Example Usage

```
// setting anti-aliasing by default
```

```
df = gdf();
df.anti_aliasing = "4x";
plot();

// changing anti_aliasing quality
f = gcf();
f.anti_aliasing = "16x";
f.anti_aliasing = "off";
```

Changelog

1.0 – Initial Revision

Copyright

This document has been placed under the license CeCILL-B.

Bibliography

- [1] Wikipedia – Anti-aliasing entry, <http://en.wikipedia.org/wiki/Anti-aliasing>
- [2] Myaa - My Anti-Alias for Matlab,
<http://www.mathworks.com/matlabcentral/fileexchange/20979>
- [3] S. Ledru - Scilab Bug 2984 , http://bugzilla.scilab.org/show_bug.cgi?id=2984