Boosting graphics-rendering performance in Windows Forms

Christopher Ireland
Steema Software
September 2010

Abstract

The performance of GDI+ in graphics-intensive Windows Forms applications can greatly reduce their responsiveness. Here I demonstrate that the use of Direct2D via SlimDX can substantially reduce some of these bottlenecks.

Keywords: Windows Forms, performance, GDI+, Direct2D, SlimDX, TeeChart

Introduction

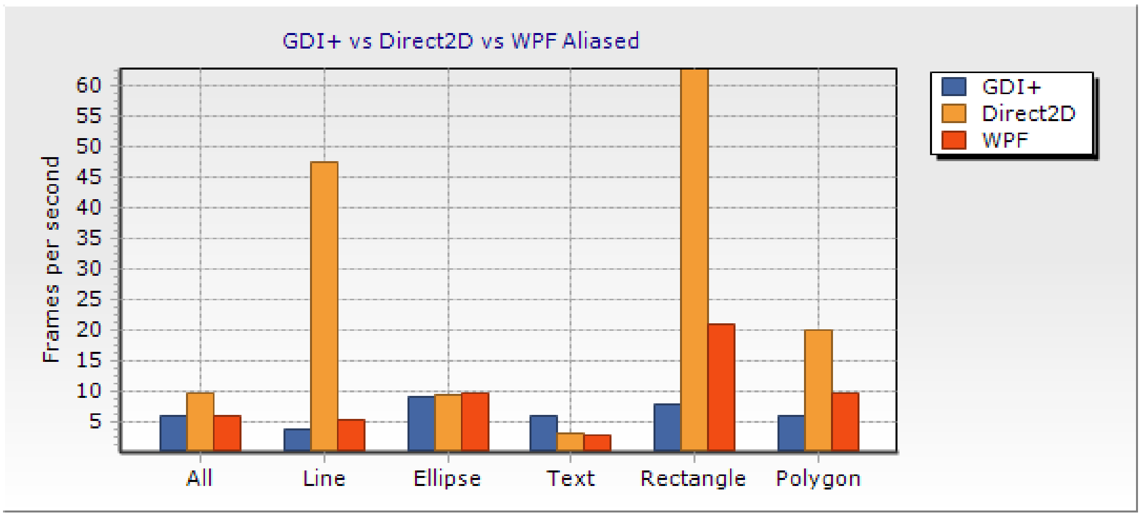
If you find that the graphics-rendering performance of Windows Forms is a problem for you, a newly available graphics API may be the solution you are looking for. While WPF is presented as a higher-performance alternative for.NET developers willing to re-design their existing applications, there is a technology gap for those with large codebases based on Windows Forms.

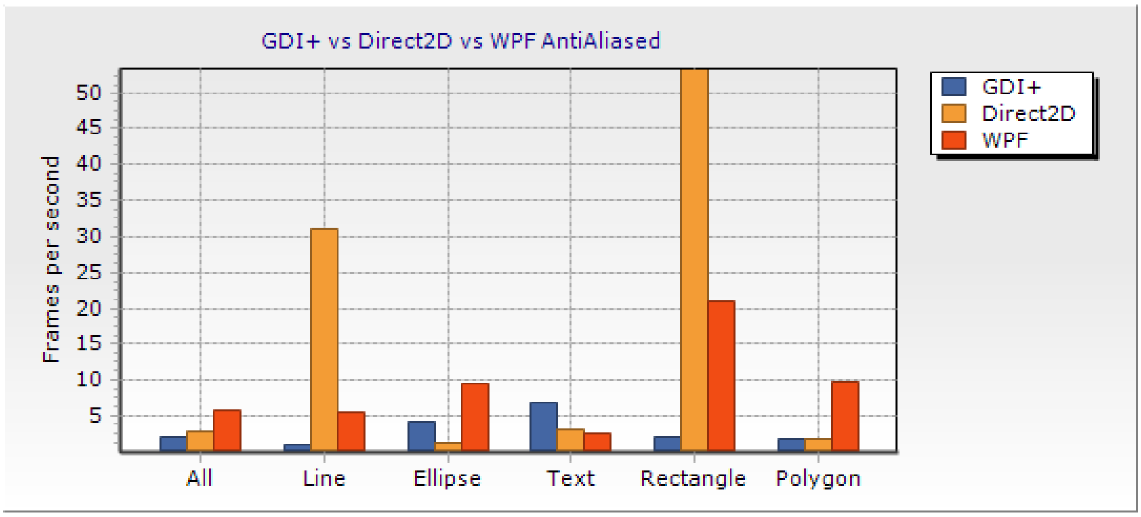
Microsoft's Direct2D is a hardware-accelerated, immediate-mode 2-D graphics API that provides high performance and high quality rendering for 2-D geometry, bitmaps, and text. SlimDX is a free open source framework designed to be an efficient, simple, and lean wrapper that exposes Direct2D to managed code. Together, this new graphics API can lever your existing Windows Forms code to create applications that render faster than either GDI+ or even WPF.

Graphics performance in Windows Forms

The clearest way to see the effects of different graphics API on performance is by the use of a sample application. Such an example can be downloaded here. To run this example you will need to install SlimDX and be working on a Direct2D compatible operating system (Windows Vista, Windows 2008, Windows 7).

The following results were obtained on a Windows 7 machine with an ATI HD4650 graphics card:





These results suggest:

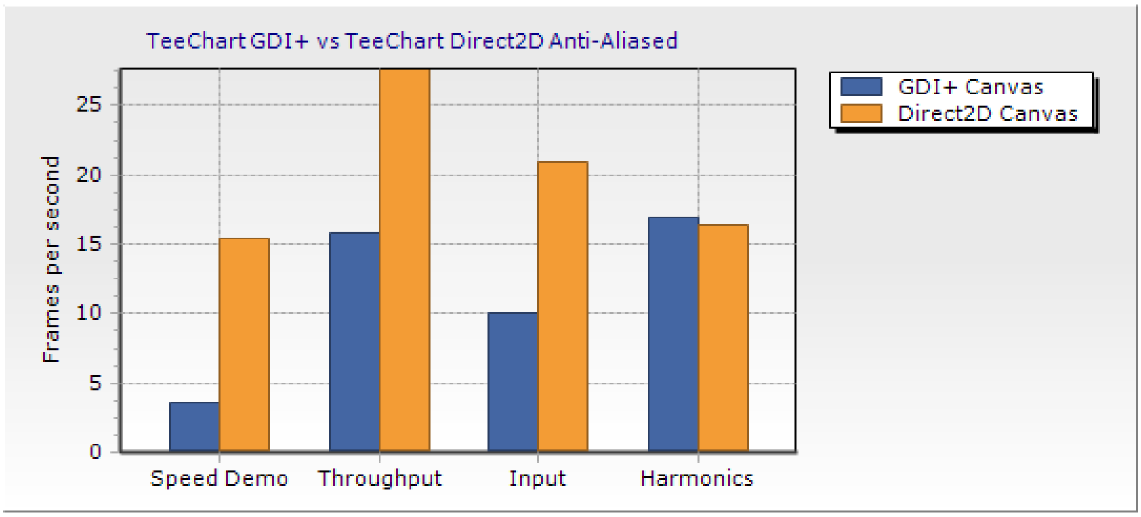
- Direct2D outperforms GDI+ both aliased and anti-aliased in all scenarios other than Text and anti-aliased Ellipse
- Aliased Direct2D outperforms WPF in all scenarios other than Ellipse*
- WPF outperforms GDI+ both aliased and anti-aliased in all scenarios other than Text*
- Anti-aliased Direct2D Line and Rectangle are several magnitudes of time faster than their WPF equivalents*

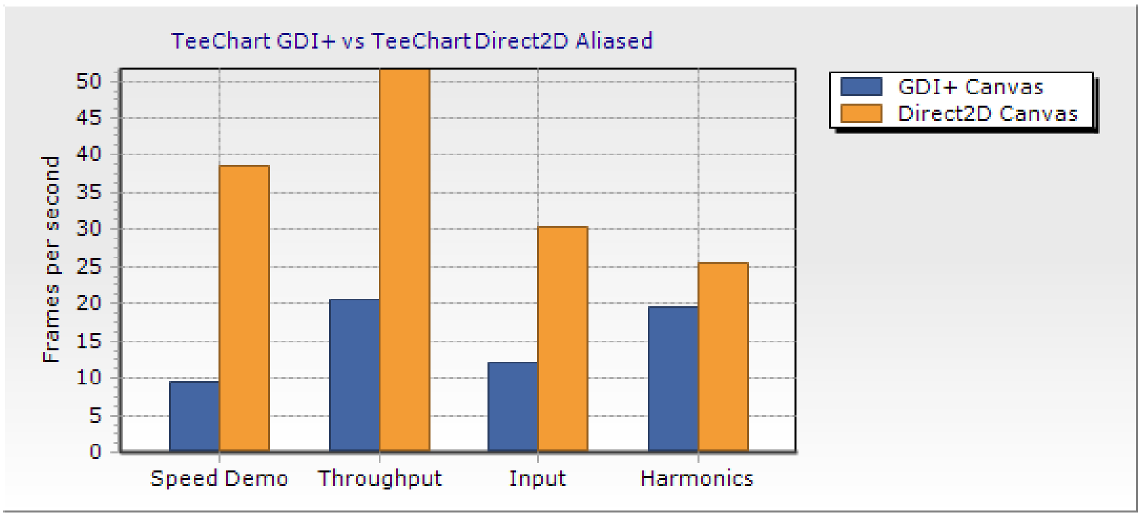
*WPF does not have an aliased mode as such. Please note that the published figures for WPF frames per second appear to be substantially higher than the frequency at which the application repaints.

Graphics performance in production code

Steema Software has taken advantage of this new technology to create a new Charting Component designed specifically for high performance rendering in Windows Forms. To see for yourself the performance gains enjoyed by using the Direct2D API in Windows Forms, you can download a free, fully functional evaluation version of TeeChart for .NET Digital Signal Processing from here .
This takes charting into new territory for Windows Forms applications whereby continuous real-world analogue signals can be faithfully rendered.

The following results were obtained by running the different demonstration applications available in the evaluation version of TeeChart for .NET DSP on a Windows 7 machine with an ATI HD4650 graphics card:





These results suggest:

- As expected, using the Direct2D API in the production code of graphics-intensive applications gives greatly increased performance in Windows Forms.
- Bar one of the above demos, using Direct2D increases the performance in both Aliased and Anti-Aliased scenarios.
- The Speed Demo was outputting 10,000 lines per frame, that is, using Direct2D Aliased up to 386,400 lines per second were being drawn, over two and a half as many points as in the equivalent GDI+ demo.

Conclusion

Using Direct2D within the SlimDX wrapper can give significant performance gains for graphics-intensive Windows Forms applications. This is good news especially for those developers looking to represent real-time data in a Chart Component such as TeeChart for .NET DSP.