

Victor Laskin's Blog

Programming, architecture and design (C++, QT, .Net/WPF, Android, iOS, NoSQL, distributed systems, mobile development, image processing, etc...)

Stackblur – C++ multi-threaded version of fast blur algorithm

I found nice algorithm for blurring images – Stackblur by Mario Klingemann. It could do the job relatively fast and gives decent quality. You can check it here – [web demonstration](#). As you can see it can be usable even in web projects.



As i wanted to include it to my cross-platform engine i found two c++ implementations:

- <https://gist.github.com/benjamin9999/3809142>
- http://www.antigrain.com/_code/include/agg_blur.h.html#stack_blur_rgba32

First is SSE friendly, second contains some division optimization via static tables. However, both are not using all cpu cores. I took second one as foundation for my implementation, as i expected my code to work on mobile devices with no SSE support. Single-core processing of 1920×1200 rgba image with 100 px radius took only 219 ms (Intel Q9550, Windows 7).

I improved stackblur code to multi-threaded version – on my quad-core cpu speed results as expected showed 4x improvement – 63ms for the same task. You can download the part my lib below and use it as a foundation to your needs.

Download multi-threaded 32bit-color (RGBA) version of StackBlur: [stackblur.cpp](#)

I believe algorithm can be optimized even further – any suggestions are welcome.

This entry was posted in C++, Design, Image processing, Kobald, Mobile development on 26 September, 2013 [<http://vitiy.info/stackblur-algorithm-multi-threaded-blur-for-cpp/>] .

ALSO ON VICTOR LASKIN'S BLOG

<p>Writing custom protocol for nanomsg</p> <p>6 years ago • 5 comments</p> <p>Nanomsg is next version of ZeroMQ lib, providing smart cross-platform sockets for ...</p>	<p>Immutable serialisable data structures in ...</p> <p>6 years ago • 4 comments</p> <p>As C++ is very flexible language there are a lot of ways to construct ...</p>	<p>C++11 functional decomposition – ...</p> <p>6 years ago • 12 comments</p> <p>This post is about making functional decomposition from perspective of ...</p>	<p>Encapsulation of asynchronous ...</p> <p>5 years ago • 3 comments</p> <p>This is an example how to create continuation-like monadic behaviour in ...</p>	<p>Templates a class citizer</p> <p>6 years ago • 12</p> <p>C++11 treats fi first-class citize gives us ability</p>
--	--	---	--	--



LOG IN WITH

OR SIGN UP WITH DISQUS **Dr-Emann** • 7 years ago

I can't seem to find MVThread. I've tried searching your blog, as well as the internet, but can't seem to find it.

 |  • Reply • Share ›**Victor Laskin** → Dr-Emann • 7 years ago

This is just simple thread class - contains virtual void run() function to do some job asynchronously. Has start(); and wait(); methods - one runs thread task and another waits until its finished. So you can use any thread implementation here (c++11 std::thread, boost::thread from boost lib, etc).

 |  • Reply • Share ›**Royi** • 7 years ago

Hi,

Is there a MATLAB implementation?

Or maybe Pseudo Code / C Cdoe with no optimization just to understand the method?

Thank You.

 |  • Reply • Share ›**Victor Laskin** → Royi • 7 years ago

I dont know about matlab implementation, but you can probably look into JS version as it expected to have more obvious code (<http://www.quasimondo.com/S....>)

 |  • Reply • Share ›



Royi → Victor Laskin • 6 years ago

<https://github.com/RoyiAvit...>

^ | v • Reply • Share ›



Elam • 7 years ago

Hi! I'm doing it on Android, our demands has some different. First, I must get a shot of screen, and then blur it, then show it immediately. I am write it with C program, and build with NDK. But I got some trouble, I must done with it lower than 200ms, but I didn't found a way to do it. So, can you give me some help or suggest? Thank you very much! Sorry for my english.

^ | v • Reply • Share ›



Victor Laskin → Elam • 7 years ago

Do you have trouble to perform blur fast enough or to get/show image? Blur part speed should depend on device resolution in your case. May be its even good idea to use every second pixel in case of full hd device resolution. For normal case algorithm described here should work under 200ms - how fast are your results? May be slow part is on java side when you provide image data to blur.

^ | v • Reply • Share ›