Libretro Forums > RetroArch > RetroArch - Shaders > ScaleNx - Artifact Removal and Algorithm Improvement

PDA

View Full Version : ScaleNx - Artifact Removal and Algorithm Improvement

Sp00kyFox 2014-11-02, 9:15

Download:
https://anonfiles.com/file/e3ae5a2d167f ... b887d5f79f (https://anonfiles.com/file/e3ae5a2d167f7356f7b42cb887d5f79f)

I noticed that the scaleNx filters are missing from the shader repository. Even if they are outdated by more superior edge interpolation algorithms today I still find them useful for low resoultion games to remain that pixelated picture while still give it more refinement.

I implemented both scale versions (2x, 3x) based on the pseudocodes presented here:
http://scale2x.sourceforge.net/algorithm.html

But there were two thing I wasn't satisfied with. First, both versions are creating weird little artifacts which are especially annoying with a low resolution source image (well known problem of this algorithm). And second, while scale2x smoothes out 45° edges just fine the 3x variant has its problem with them. You can see both issues is this comparison shot:

https://i.imgur.com/NlIKVEY.png

So I created a post processing shader (ScaleNx-Cleaner) which looks for these artifacts and removes them by replacing those pixels with the surrounding ones. And I made an advancement to the rule set of the 3x variant called Scale3xEX which should make the original obsolete:

https://i.imgur.com/1XeGjQq.png

gallery:
https://imgur.com/a/BEJn7

hunterk 2014-11-02, 17:55

Wow, big improvements. Great job! You need me to push these up to the common-shaders repo?

Sp00kyFox 2014-11-02, 18:05

yeah, that would be nice. thx!

hunterk 2014-11-07, 19:33

Finally pushed these up to the repo. Sorry for the delay.

Sp00kyFox 2014-11-10, 13:59

no problem. It's like a bad habit but I'm already working on an improved version. I wasn't really satisfied with removing the artifacts afterwards instead of avoiding them right away. so I'm gonna up a new version soon. this will make the cleaner shaders obsolete.

Sp00kyFox 2015-01-08, 20:28

ok, got a new version, no need for the extra cleaner pass anymore. I also modified the algorithm so it properly works with equality thresholds. here is the download:

https://mega.co.nz/#!yYohhDCA!jgEe41OL5rueTLzSP3GMYIBaGgc7sjyfy95-8hb8KXc

for the interested reader I'm gonna explain both algorithms:

Scale2xSFX

if anyone used the original scale2x or EPX one can clearly notice single outstanding pixels here and there which are very annoying and the only drawback in my opinion for this method which otherwise pretty much keeps the pixelart style intact.

ok, why does these artifacts (see the first post in this thread for an example screen) come up? first, let's give those pixels some names:

```
J
A B C E0 E1
K D E F L E2 E3
G H I
M
```

where E is the central pixel and E0 to E3 are the 4 subpixels of E (and A0 to A3 are the subpixels of A and so on).
well, let's say E0 is the artifact pixel. what happened is that the condition to replace E0 is met simultaneously with the ones of B2 and D1. this implies (look at the original algorithm for that) that the surrounding pixels of E0 (which are E1, E2, B2 and D1) are all of the color E and therefore E0 itself is an outstanding pixel because it isn't E anymore.

the way to fix this is obvious. just check if the conditions for B2 and D1 are also met, if that's the case then don't touch E0. well here is the original and the improved algorithm in pseudocode for comparison.
note: to allow the algorithm to work correctly with threshold comparisons the so called parent condition is slightly altered but still aquivalent to the original one in the case of true equality checks.

Scale2x

```
E0 = B=D & B!=F & D!=H ? 0.5*(B+D) : E
E1 = B=F & B!=D & F!=H ? 0.5*(B+F) : E
E2 = D=H & B!=D & F!=H ? 0.5*(D+H) : E
E3 = F=H & B!=F & D!=H ? 0.5*(F+H) : E
```

Scale2xSFX

```
E0 = B=D & B!=F & D!=H & (E!=A | E=C | E=G | A=J | A=K) ? 0.5*(B+D) : E
E1 = B=F & B!=D & F!=H & (E!=C | E=A | E=I | C=J | C=L) ? 0.5*(B+F) : E
E2 = D=H & B!=D & F!=H & (E!=G | E=A | E=I | G=K | G=M) ? 0.5*(D+H) : E
E3 = F=H & B!=F & D!=H & (E!=I | E=C | E=G | I=L | I=M) ? 0.5*(F+H) : E
```

the actual shader code looks a bit different due to subterm optimization.

Scale3xSFX

the cause of artifacts and how to prevent them is analogue to the case of Scale2x. the interesting flaw from Scale3x is that it doesn't properly handle 45° lines, on those lines Scale3x only adds pixels but doesn't "shave" them off. so to fix this we have to add another condition.
but first let's have a look at the subpixels of E at scale factor 3:

```
J
A B C E0 E1 E2
K D E F L E3 E4 E5
G H I E6 E7 E8
M
```

again.. A0 to A8 are the subpixels of A and so on.
okay, let's say the pixels E, C and G are building a 45° line. then the condition for B8 and D8 is met but now at scale factor 3 E0 is standing out like a sore thumb. so to fix this we again use a very simple rule, just check if the condition for B8 or D8 is satisfied. if this is the case then "remove" E0 (replace it with B=D).
and here is the pseudocode comparison:

Scale3x

```
E4 = E

E0 = B=D & B!=F & D!=H ? 0.5*(B+D) : E
E2 = B=F & B!=D & F!=H ? 0.5*(B+F) : E
E6 = D=H & B!=D & F!=H ? 0.5*(D+H) : E
E8 = F=H & B!=F & D!=H ? 0.5*(F+H) : E

E1 = (B=D & B!=F & D!=H & E!=C) | (B=F & B!=D & F!=H & E!=A) ? B : E;
E3 = (B=D & B!=F & D!=H & E!=G) | (D=H & B!=D & F!=H & E!=A) ? D : E;
E5 = (B=F & B!=D & F!=H & E!=I) | (F=H & B!=F & D!=H & E!=C) ? F : E;
E7 = (D=H & B!=D & F!=H & E!=I) | (F=H & B!=F & D!=H & E!=G) ? H : E;
```

Scale3xSFX

```
E4 = E

E0 = (B=D & B!=F & D!=H & (E!=A | E=C | E=G | A=J | A=K)) | (B=D & C=E & C!=J & A!=E)
| (B=D & E=G & A!=E & G!=K) ? 0.5*(B+D) : E
E2 = (B=F & B!=D & F!=H & (E!=C | E=A | E=I | C=J | C=L)) | (B=F & A=E & A!=J & C!=E)
| (B=F & E=I & C!=E & I!=L) ? 0.5*(B+F) : E
E6 = (D=H & B!=D & F!=H & (E!=G | E=A | E=I | G=K | G=M)) | (D=H & A=E & A!=K & E!=G)
| (D=H & E=I & E!=G & I!=M) ? 0.5*(D+H) : E
E8 = (F=H & B!=F & D!=H & (E!=I | E=C | E=G | I=L | I=M)) | (F=H & C=E & C!=L & E!=I)
| (F=H & E=G & E!=I & G!=M) ? 0.5*(F+H) : E

E1 = (B=D & B!=F & D!=H & (E!=A | E=C | E=G | A=J | A=K) & E!=C)
| (B=F & B!=D & F!=H & (E!=C | E=A | E=I | C=J | C=L) & E!=A) ? B : E;
E3 = (B=D & B!=F & D!=H & (E!=A | E=C | E=G | A=J | A=K) & E!=G)
| (D=H & B!=D & F!=H & (E!=G | E=A | E=I | G=K | G=M) & E!=A) ? D : E;
E5 = (F=H & B!=F & D!=H & (E!=I | E=C | E=G | I=L | I=M) & E!=C)
| (B=F & B!=D & F!=H & (E!=C | E=A | E=I | C=J | C=L) & E!=I) ? F : E;
E7 = (F=H & B!=F & D!=H & (E!=I | E=C | E=G | I=L | I=M) & E!=G)
| (D=H & B!=D & F!=H & (E!=G | E=A | E=I | G=K | G=M) & E!=I) ? H : E;
```

for a more detailed explanation of the original algorithms go visit:
http://scale2x.sourceforge.net/

hunterk 2015-01-08, 21:07

Awesome write-up :D

I'll get the repo updated with your latest code ASAP.

Sp00kyFox 2015-09-06, 17:39

Hi there, just wanna announce that I revisited my modification of scaleNx. I guess my original version was a bit hard to understand since I cramped everything into one step. Now I seperated it into multple passes and added a few enhancements. This should not only make it easier to understand and modify but also faster because of fewer redundant calculations.

here is a little preview, 2 x scale3xSFX (click for full resolution):
http://i.imgur.com/2oVBxW0m.png (https://i.imgur.com/2oVBxW0.png) http://i.imgur.com/kn9kEHOm.png (https://i.imgur.com/kn9kEHO.png)

I still need to clean up the code, download will be ready in the next few days.

hunterk 2015-09-06, 19:25

Looks great. It does an awesome job with curves :)

Sp00kyFox 2015-10-04, 0:28

if anyone is wondering, just wanna mention that I'm still working on it. I saw the potential for a more sophisticated interpolation, so it isn't just an exercise without any real purpose.
here is a wip screenshot, I added lvl3 edge detection (you can see it on the left side of shantae's hair)

https://i.imgur.com/jWX8VjY.png

Hyllian 2015-10-04, 0:40

It's looking interesting, Sp00kyFox.

Is it intended only for low color games (8 and 16-bit systems) or for most modern anti-aliased ones? Do you think it would be hard to add any anti-aliasing treatment to the algorithm, or it isn't your goal?

BlockABoots 2015-10-04, 15:15

Should rename this shader 'waterpaint'

Amazing the results that can be achieved though

Sp00kyFox 2015-10-07, 9:05

sorry to keep you waiting. I spend the time pushing it forward to a release status and it's ready!

download:
https://mega.nz/#!HNImzbDD!LTys9sZAVI1qaIpMmUfz2pleCZ5y5FLg6DQ3rEU pdK0

screenshot album:
https://imgur.com/a/lzBPe

there is a user-defined threshold value. the 9x-cgp can be rather demanding on your system, so be warned.
screenshots or suggestions for shader combos are welcome.

It's looking interesting, Sp00kyFox.
Is it intended only for low color games (8 and 16-bit systems) or for most modern anti-aliased ones? Do you think it would be hard to add any anti-aliasing treatment to the algorithm, or it isn't your goal?
thx. well I developed it with gameboy and NES games in mind. the edge detection is general enough so that it also deals with anti-aliased graphics but it doesn't differentiate. it does an ok job on it I guess but it doesn't compete in that regard with specialized filters, the optimal usage is certainly with pixelart. to be honest with you I don't really know how one would deal with anti aliased material so I would need to study the topic first before I can answer that question. my algorithm itself is not 'too complex' though, so feel free to take a look. I'll try to explain it in further detail in the near future.

Hyllian 2015-10-07, 9:56

It reminds me of xbr-noblend filter in that gallery. It indeed treats anti-aliasing some way, so you have some kind of color distance measure.

Sp00kyFox 2015-10-07, 10:25

It reminds me of xbr-noblend filter in that gallery. It indeed treats anti-aliasing some way, so you have some kind of color distance measure.
indeed, xbr-noblend looks similar and in my test runs I used it for comparisons. one of the goals of the original idea was to modify the scaleNx algorithm so that it can deal with color gradients (where the original one only checks for equality). I experimented with a few metrics but finally stayed with the one I used in some of my previous shaders which I think yields better results than the often used YUV-method:

http://www.compuphase.com/cmetric.htm

Sp00kyFox 2016-01-25, 19:01

I'm back with a new version. I slightly improved the algorithm by changing the way the color threshold works. until now it could actually could get in the way of itself if multiple valid but contradicting edge candidates are available. now it prioritizes those with minimal distance value. as a result you can pretty much crank up the color threshold value as much as you like without destroying the picture or lessen the edge interpolation.
also calling it ScaleFX now (SFX - Sp00kyFox, you see what I did there? ^^)

I also worked on combining the reverse AA filter with ScaleFX. came up with a solution by doing the edge detection on the original picture but taking the subpixels in the last pass from rAA. this works surprisingly well. since I didn't incorporate the algorithm itself the neat thing is that one can replace rAA with any other preshader as long as it outputs a 3x-version of the original frame.

download:
https://mega.nz/#!bAhTXKSY!EoeEy3Xt6yusa4ebEuG9LHiF9lH4KmRwF9K3qlz vMIc

gallery:
https://imgur.com/a/UatLR

original
https://i.imgur.com/kDZSKbI.png

ScaleFX-hybrid (rAA)
https://i.imgur.com/0VtZxQY.png

Hyllian 2016-01-25, 19:25

Great update, Sp00kyFox! It seems to be a good alternative to xbr-hybrids.

Hyllian 2016-01-30, 1:21

I just added your newest version to common-shaders.

Sp00kyFox 2016-02-19, 17:18

tl;dr:
new version, supports lvl6 edges, way more performant.
here is the download and and in the spirit of the SF5 release a gallery with some street fighter games:

download:
https://mega.nz/#!7BhiSSjK!yyfxcPp7VwCxPkwV6O5FlyRAzZPP0q-DVr-6C1Nqg74

gallery:
https://imgur.com/a/X71Bw

Development Notes

back with another major update. I was working on the edge detection and the shader now supports up to level 6 (which is 6 pixels between each stair). the super mario title screen is a good example:

old version:
https://i.imgur.com/nABZTaY.png

new version:
https://i.imgur.com/Mi4o1QI.png

during implementing it I noticed a problem. until now I determined the edge level by counting the pixels beginning from the stair into one direction. this led to improper sloping at the end of a staircase pattern. this is also fixed now by looking into both directions. compare the hud of shantae.

original, old version, new version:
http://i.imgur.com/10GnEWI.png
http://i.imgur.com/ZZFvTNA.png
http://i.imgur.com/f4i7RgD.png

adding level 6 support and looking into both directions costs a lot more texture lookups though. lucky enough there was potential for a huge performance gain. until now I did the edge analysis in the last pass which runs at scale factor 3. so every texture lookup and calculation is done again and again for each of the nine subpixels. I added a fifth pass which only takes the result from the last pass by doing one lookup. therefore the previously last pass can now run at scale factor 1 and runs a lot faster because of this.
especially in the case of using ScaleFX twice (to get a 9x filtered image) you should notice a big improvement in performance.

regarding file and folder structure. there is an overlap between the normal and hybrid variant. in fact only pass0 and pass1 are different, so I removed the other ones and changed the cgp files accordingly.

Hyllian 2016-02-19, 18:23

It's looking great, Sp00kyFox!

The stair at lvl6 is amazing!

So, looking at both directions and one of them can be shorter than the other. Then you choose the level based on the shorter one? (it's the heuristic I have used on xbr mlv4 more or less, because I was getting some weird results at the time too.)

Looking at the SF gallery, are you using the pure or hybrid one there?

To tell you the truth, I prefer your pure ScaleNx than the hybrid one, for the same reason I prefer pure xbr over its hybrid version. The algorithms don't seam very well, and that's why I abandoned the development of it in favor of the new super-xbr ones.

Maybe you should look at the way super-xbr work to begin a development of a Super-ScaleNx? An algorithm that could work well on digitized games and anti-aliased ones.

hunterk 2016-02-19, 18:43

Oh, shit! Those straight lines instead of stairsteps on the SMW logo..! :O

The numbers from the Shantae HUD look very clean, too. Great job, dude.

Sp00kyFox 2016-02-19, 22:51

thanks for the positive feedback you two!

So, looking at both directions and one of them can be shorter than the other. Then you choose the level based on the shorter one? (it's the heuristic I have used on xbr mlv4 more or less, because I was getting some weird results at the time too.)
no, not quite. to have a common base, the basic idea is that edge interpolation at scale factor 3 can be done by counting the pixels of a stair. the result is the number of subpixels we "shave" off. to make the transition a little bit smoother I'll take one extra subpixel off after a staircase ends. here is a demonstration of level 1 to 3. red pixels are removed, blue pixels are added (of course from the pov of the shader this is the same just looked upside down):

https://i.imgur.com/nzVjThb.png

to explain the rules.. in pass2 the algorithm tags corner pixels, it also provides a basic edge rule I then use in pass3 for the level detection. to make it easy let's assume we are looking at a horizontal staircase pattern and the subpixel in focus is the upper left one. so level 1 is trivial, it just checks if the subpixel is marked. for level2 we check if the edge beginning on the upper left goes on to the next pixel on the right and if one pixel below our position begins a new stair. for every additonal level we check if the edge goes on for one more pixel on each stair.
and another demonstration for the first three levels. red marks the part of the upper stair, blue the part of the lower stair which must both be present to qualify the level:

https://i.imgur.com/BTt9UWL.png

To tell you the truth, I prefer your pure ScaleNx than the hybrid one, for the same reason I prefer pure xbr over its hybrid version. The algorithms don't seam very well, and that's why I abandoned the development of it in favor of the new super-xbr ones. Maybe you should look at the way super-xbr work to begin a development of a Super-ScaleNx? An algorithm that could work well on digitized games and anti-aliased ones.
in the SF3 character select screen examples with the digitized artworks it does an inferior job in comparison to the hybrid version. but I can see what you mean. even if used with rAA anti-aliased edges are a little bit rough. I'll definitely gonna have a look at your super-xbr shader. but first I need some time to rank up in SF5!

Hyllian 2016-02-20, 11:09

...I'll definitely gonna have a look at your super-xbr shader. but first I need some time to rank up in SF5!

Oh, great!

To help you understand how you can modify Super-xBR to introduce ScaleNX ideas, read this small text I wrote past year:

https://drive.google.com/file/d/0B_yrhrCRtu8GYkxreElSaktxS3M/view

It explains the main idea of it.

Sp00kyFox 2016-02-20, 13:12

interesting paper. this helps a lot to understand the concept. I think the approach or scalefx would need to undergo some big changes though to make it work. what I see as the main difference between the two shaders is that xbr looks at the grand scheme while scalefx acts very locally. currently my shader doesn't use something like edge strength to determine how to interpolate.

one of the core elments in scalefx is to avoid artefacts (occuring in the original filter) by resolving disambigous setups at pixel junctions. this is done by taking only the subpixels in consideration which are lying directly at the junction in question. so in other words, in this part of the algorithm I only look at 4 pixels to determine if there is a NW-SE or NE-SW edge (or neither) going through a junction. this is why scalefx honors little details so much which is great for true pixelart. which angle is present is then indirectly calculated by the ruleset for edge levels I explained above.
so the only part where strength values are compared is in the dissolve junction formula (pass2) and these are based on 2 pixel long 45° diagonals.

also scalefx is working on scale factor 3 where the original pixel color is kept in the center subpixel and not the upper left one. not saying that it can't work, but it'll probably need a lot of modifications on both sides.

Hyllian 2016-02-20, 23:10

interesting paper. this helps a lot to understand the concept. I think the approach or scalefx would need to undergo some big changes though to make it work. what I see as the main difference between the two shaders is that xbr looks at the grand scheme while scalefx acts very locally. currently my shader doesn't use something like edge strength to determine how to interpolate.

one of the core elments in scalefx is to avoid artefacts (occuring in the original filter) by resolving disambigous setups at pixel junctions. this is done by taking only the subpixels in consideration which are lying directly at the junction in question. so in other words, in this part of the algorithm I only look at 4 pixels to determine if there is a NW-SE or NE-SW edge (or neither) going through a junction. this is why scalefx honors little details so much which is great for true pixelart. which angle is present is then indirectly calculated by the ruleset for edge levels I explained above.
so the only part where strength values are compared is in the dissolve junction formula (pass2) and these are based on 2 pixel long 45° diagonals.

also scalefx is working on scale factor 3 where the original pixel color is kept in the center subpixel and not the upper left one. not saying that it can't work, but it'll probably need a lot of modifications on both sides.

Have you thought about using xbr-noblend as a first pass at 3x and the next passes using ScaleNx?

Sp00kyFox 2016-02-21, 0:20

no, not really. I'll keep it in mind if I'm coming back for this shader.

Sp00kyFox 2016-02-22, 20:26

tl;dr: improved edge transition and performance.

download:
https://mega.nz/#!bU4AUA7C!3VjlM64gKG1GGd2lrfImisdvujST3DR5wmG1TSj ZEQs

small update. after my explanations in the previous posts I rethought the transition of edges into straight lines. the extra subpixel on level 1 edges is fine but not enough for edges with other levels. if we think about it as linear function graphs with a certain slope then an edge with level N has the slope of 1/N and a straight horizontal line has a slope of 0 of course. so the transition should meet in between those two which gives us (1/N+0)/2 =1/(2N) which corresponds to an edge with level 2N. so we "shave off" 2 subpixels at the last stair for a level1 edge, 4 subpixels for level2, 6 subpixels for level3 and so on and so forth.

it's not a huge difference in general but I verified in tests that it does give a better result. on top of that the positive side effect is that the amount of texture lookups in pass3 could be greatly reduced again from 21 to 13. for performance testing I usually see how it runs with the bsnes accuracy core when applied twice for a scale factor of 9. for the first time it runs in fullspeed for me without struggling. would be interesting to know if this shader also works well on consoles or handhelds after the performance improvements of the last updates.

Hyllian 2016-03-01, 16:30

I just updated the repo with this new version. Thanks, Sp00kyFox.

I have taken a look at your shaders and found the color difference code very interesting. I've read the source page and got the original formulas to test on my xbr shaders. And I find the results better than just using the luma values as I was doing. The only problem is that it's very expensive, so I think I'll only insert it in the multipass versions. It basically get rid of the stray red spike artifacts of current xbr, which I knew were caused by an inaccurate color diff function.

Sp00kyFox 2016-03-25, 22:03

a little teaser...

https://i.imgur.com/JOXwtBm.png

https://i.imgur.com/VZfpaWK.png

Hyllian 2016-03-25, 22:33

It's looking great! I can see that you're taking the same path xbr took. I can see some anti-aliasing filtered.

You can improve some glitches by changing the color distance params. Try those values I posted earlier and you'll see some fixes (Yoshi's eyebrows and nose for example).

hunterk 2016-03-25, 23:02

I really like how it handles gradual curves. That dolphin from SMW looks beautiful.

Hyllian 2016-03-25, 23:22

Here's another image test I use when trying new ideas (maybe it's useful for you):

http://abload.de/img/my_test2pios9.png

Copy directly from this link (do not copy from the resized image above): http://abload.de/img/my_test2pios9.png

Sp00kyFox 2016-03-26, 19:00

It's looking great! I can see that you're taking the same path xbr took. I can see some anti-aliasing filtered.
actually the issue I was trying to fix was zig-zag lines, like the mouth from the ghost in SMW which weren't interpolated at all in prior versions. so I made the requirement for corner candidates less strict which also had a positive effect on the handling of anti-aliased edges. I should mention that the filter doesn't mix colors. I don't have the intention to make the filter go in the same direction as xBR and compete with it. I want ScaleFX to be a interpolation filter specialized in pixelart with decision making based on a very local neighborhood. like in the tradition of classic filters but expanded to a level which wasn't possible back then.

You can improve some glitches by changing the color distance params. Try those values I posted earlier and you'll see some fixes (Yoshi's eyebrows and nose for example).
despite the latest changes your triplets are still not a good choice for this filter. the original one seems to be the most balanced for ScaleFX.

I really like how it handles gradual curves. That dolphin from SMW looks beautiful.
thanks and I appreciate it that you presented my filter on your blog!

Here's another image test I use when trying new ideas (maybe it's useful for you):
here it is:

https://i.imgur.com/82WOdmY.png https://i.imgur.com/xfInRgQ.png

I was trying around with some shader combinations and I found out that xsoft is a pretty good complementary filter if you want a smoother picture.

original
https://i.imgur.com/HlK2D6T.png

ScaleFX + xsoft
http://i.imgur.com/blkWr9v.png

and here is one from the game Blackwell Epiphany (btw an awesome adventure series). too bad it's just a screenshot, we seriously need a libretro dx hook!

original
http://i.imgur.com/v1JQXWw.png

ScaleFX-hybrid + xsoft
http://i.imgur.com/vqAKti5.png

Hyllian 2016-03-26, 23:00

It's looking great. That example shows a very good output from the filter.

SkyHighGam3r 2016-03-27, 16:26

I was trying around with some shader combinations and I found out that xsoft is a pretty good complementary filter if you want a smoother picture.

ScaleFX + xsoft
http://i.imgur.com/blkWr9v.png

Dear God... it's beautiful. These are both just in the Shaders_CG folder right. (is glsl better?) This is going to be my new "HDMI" preset.

Sp00kyFox 2016-03-27, 18:04

@SkyHighGam3r
thanks! not yet. it was just a preview for the upcoming version but I'm still working on it. don't wanna release another version which is outdated by a new update after a few days. you can still use the current version though. you can find it under shaders_cg/scalenx/scalefx-3x.cgp. if you wanna combine it with xsoft like in the screenshot you need to load shaders_cg\xsoft\shaders\4xsoft.cg on top of it (nearest, 2x but 1x works as well).

Sp00kyFox 2016-04-02, 19:09

the wait is over. I finished my filter so far and it's available on the shader repository (you can get it by using the "Download Zip" button on the main page). from now on ScaleFX can be found under the main folder scalefx:
https://github.com/libretro/common-shaders/tree/master/scalefx

here are the noteworthy changes:
- reduced the number of passes to 4 (all metric work is now done in the first pass) -> better performance
- improved edge rules -> more anti-aliased edges are interpolated
- fixed: sometimes little bumps would appear on edges if an interpolation was tangent to it
- improved performance in the final pass by removing a lot of unnecessary switch-case structures

and here are some examples. in the case of SF3 and the Wadjet Eye adventure games I used ScaleFX-hybrid:
https://imgur.com/a/3d7BG
http://screenshotcomparison.com/comparison/167831
http://screenshotcomparison.com/comparison/167832
http://screenshotcomparison.com/comparison/167833
http://screenshotcomparison.com/comparison/167840

(ps: hardcoregaming101.net is a great source of unaltered pixelart screenshots with original resolution)

I hope you enjoy it!

http://i.imgur.com/53juJi2.png
http://i.imgur.com/Zsy5GP6.png

http://i.imgur.com/Ygy6Ny2.png
http://i.imgur.com/GN91eFL.png

http://i.imgur.com/cpVebgq.png
http://i.imgur.com/xdF64iQ.png

Hyllian 2016-04-02, 21:31

It looks great! The slopes in level 6 are very good!

Maybe there's some way to blur a bit less than xSoft, to make it a bit more crispy. IMO, I think that xSoft damage the colors too much.

lordmonkus 2016-04-03, 1:09

Wow, this looks freakin awesome. Not my personal tastes in a shader being more of a CRT guy but damn this looks really good for anyone who prefers the whole "smooth look" type of shader.

Silverbreaker 2016-04-03, 13:03

look at my shader preset, it combines new & old 373
its far more authentic:

(set the zoomlevel to 50% and you can imagine what it looks like on your retroarch)
http://www.bilder-upload.eu/thumb/762eae-1459698108.jpg (http://www.bilder-upload.eu/show.php?file=762eae-1459698108.jpg)

http://www.bilder-upload.eu/thumb/0e2363-1459699124.png (http://www.bilder-upload.eu/show.php?file=0e2363-1459699124.png)

http://www.bilder-upload.eu/thumb/7cdc1a-1459698177.jpg (http://www.bilder-upload.eu/show.php?file=7cdc1a-1459698177.jpg)

http://www.bilder-upload.eu/thumb/caec8f-1459699153.png (http://www.bilder-upload.eu/show.php?file=caec8f-1459699153.png)

http://www.bilder-upload.eu/thumb/5395f5-1459698191.jpg (http://www.bilder-upload.eu/show.php?file=5395f5-1459698191.jpg)

http://www.bilder-upload.eu/thumb/81fef9-1459698204.jpg (http://www.bilder-upload.eu/show.php?file=81fef9-1459698204.jpg)

http://www.bilder-upload.eu/thumb/d0edbd-1459698328.jpg (http://www.bilder-upload.eu/show.php?file=d0edbd-1459698328.jpg)

http://www.bilder-upload.eu/thumb/155521-1459698214.jpg (http://www.bilder-upload.eu/show.php?file=155521-1459698214.jpg)

oh yeah, change your retroarch-shader-paths in the attached Favorit.cgp-file (with notepad++ for example)

Sp00kyFox 2016-04-03, 14:19

@Silverbreaker
ok, but I would prefer it if you wouldn't hijack my thread for you shader preset, especially if it has nothing to do with ScaleFX. you already created your own thread. if you want to have more attention, I would suggest delivering some screenshots instead of posting about it in threads about other filters. thanks.

the thread for xBR is over there:
http://libretro.com/forums/showthread.php?t=88

Silverbreaker 2016-04-03, 14:47

you're right, sorry about that ...
(btw. its not only xbr - its the combination of xbr + blur + ntsc + crt + color/setting adjustments)

BONKERS 2016-04-04, 4:54

I registered just to say, you should totally try this with downsampled rendering.

Either with a custom resolution or using Nvidia DSR/ AMD VSR with a 2x2 (4.0x)factor. With DSR, use a low smoothness value like 15-20%.

You won't be able to capture that result, but if you have FastStone you can take the screenshot put out and downsample it by 0.5 in each direction with a lanczos 2 filter to get an approximation.

I'm just curious to see the results

SkyHighGam3r 2016-04-05, 13:52

the wait is over. I finished my filter so far and it's available on the shader repository (you can get it by using the "Download Zip" button on the main page). from now on ScaleFX can be found under the main folder scalefx:
https://github.com/libretro/common-shaders/tree/master/scalefx

here are the noteworthy changes:
- reduced the number of passes to 4 (all metric work is now done in the first pass) -> better performance
- improved edge rules -> more anti-aliased edges are interpolated
- fixed: sometimes little bumps would appear on edges if an interpolation was tangent to it
- improved performance in the final pass by removing a lot of unnecessary switch-case structures

and here are some examples. in the case of SF3 and the Wadjet Eye adventure games I used ScaleFX-hybrid:
https://imgur.com/a/3d7BG
http://screenshotcomparison.com/comparison/167831
http://screenshotcomparison.com/comparison/167832
http://screenshotcomparison.com/comparison/167833
http://screenshotcomparison.com/comparison/167840

(ps: hardcoregaming101.net is a great source of unaltered pixelart screenshots with original resolution)

I hope you enjoy it!

http://i.imgur.com/53juJi2.png
http://i.imgur.com/Zsy5GP6.png

http://i.imgur.com/Ygy6Ny2.png
http://i.imgur.com/GN91eFL.png

http://i.imgur.com/cpVebgq.png
http://i.imgur.com/xdF64iQ.png

Lol as soon as I get my shaders set to the X-soft method you recommended, you go and release this amazing thing...

Seriously, it looks PHENOMENAL. and I look forward to setting it all up again. (No sarcasm) Thank you for all your hard work. My 2D games look more gorgeous than they ever did through the rose tinted shades of nostalgia.

SkyHighGam3r 2016-04-05, 14:00

https://imgur.com/a/3d7BG

Would it be possible to make a short list of the games you put on display here? I've never seen like half of these and some of them look really cool (Made even cooler by your shader, obviously ;))

Sp00kyFox 2016-04-05, 16:49

Would it be possible to make a short list of the games you put on display here? I've never seen like half of these and some of them look really cool (Made even cooler by your shader, obviously ;))
sure. here is the list in the same order:

1. Street Fighter III: 3rd Strike (CPS3)
2. Akumajou Dracula X: Chi no Rondo (PCE)
3-4. Blackwell Epiphany (Win)
5. Golden Sun (GBA)
6-7. Cybernator (SNES)
8-9. Mega Man 7 (SNES)
10-11. Mega Man X (SNES)
12. Castlevania: Dracula X (SNES)
13. ActRaiser 2 (SNES)
14. Demon's Crest (SNES)
15-17. Pulstar (Neo Geo)
18. Shovel Knight (Multi)
19-20. Last Resort (Neo Geo)
21-22. Shardlight (Win)
23. Mystik Belle (Win)
24-25. Joe & Mac (Arc)
26. Sly Spy (Arc)
27-28. Sam & Max: Hit the Road (DOS)
29. Top Hunter: Roddy & Cathy (Neo Geo)
30. Shantae: Risky's Revenge (NDSi, Win)
31. Technobabylon (Win)
32. Batman Returns (SNES)
33. Mega Man 8 (PSX, Saturn)
34. Dodonpachi (Arc, PSX, Saturn)

SkyHighGam3r 2016-04-05, 19:03

That's amazing, thank you! I'll have to grab some of these.

How are you running the shaders on windows games though?
(Is this a dumb question? I feel like I'm asking a dumb question lol)

Sp00kyFox 2016-04-05, 19:40

I just don't, lol. yeah, I wish I could but I just loaded up a screenshot with the "Imageviewer" core. a libretro directx hook would be awesome, but right now there is no such option to my knowledge.

pralinen 2016-04-05, 20:38

wow, it looks incredible! It looks like this is how good it can get or is there room for improvement still? Great job!

guest.r 2016-05-21, 17:17

It looks great! The slopes in level 6 are very good!

Maybe there's some way to blur a bit less than xSoft, to make it a bit more crispy. IMO, I think that xSoft damage the colors too much.

I think i can help a bit. I added a de-blur code section to the 4xSoft which makes it more crispy, in fact, it's very customizable.

I'll explain how to tweak it.

This code line manages it all: float filterparam = clamp(2.25*dif, 1.0, 2.0);

If you use for example: float filterparam = clamp(2.25*dif, 1.0, 1.0); then there will be no change compared with 4xSoft.

If you use float filterparam = clamp(2.25*dif, 2.0, 2.0); instead, then every "situation" will get deblured with factor 2.0.

Another interesting setup could be: float filterparam = clamp(2.75*dif, 0.25, 2.0); where similar pixels will get additional blur while there will be sharp contrast on the whole.

The algorithm evaluates the situation (surrounding pixels) and calculates how much deblur it applies, but it must be clamped, because filter values below 1.0 blur the image additionally. (can be helpful sometimes).
If we clamp the value between 1.0 and 2.0 then similar looking pixels will stay original while "sharp" situations will get deblured depending from the situation.

It's a subject of testing and personal preferences...

I hope this shader will be helpful for the excellent ScaleFX or any other filter (high regards to the author) and a satisfying setup will be found / used.

Best regards, guest.r

hunterk 2016-05-21, 17:28

Hey guest.r, welcome! Thanks for all the fantastic shaders you've given the community over all these years :)

guest.r 2016-05-21, 18:20

Hey hunterk! Nice to meet you. It's a privilege.

I'm in awe and amazement over what's going on here with emulation and shaders. And i'm glad some of my shaders are still considered useful lol.

My personal gol was to code a PS 2.0 2x or 4x scaler with no color comparisons, no logic used or very crude logic. But this postprocessing thingie is really addictive.

Some real effort was put into it by various contributors and it got past industrial/corporate level which i find outstanding.

Everybody keep up the good work!

Regards, guest.r