QGIS Application - Bug report #14492
raster zoomed-in resampling settings not applied with reprojected rasters

2016-03-14 08:40 PM - Mathieu Pellerin - nIRV

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Rasters		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	end of life
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	22467

Description

Interestingly enough, while zoomed-out resampling settings work fine, *zoomed-in* resampling settings for reprojected rasters (i.e. project CRS != raster CRS) is ignored, always uses nearest neighbour.

Steps to reproduce

1. Open a new project, add a raster layer (any will do, confirmed here using LANDSAT and Sentinel-2 datasets)
2. Zoom into the raster beyond its 1:1 native resolution to see large pixels
3. Open the property window for the raster, and change the zoomed-in resambling to bilinear
4. Apply and leave the property window, notice the smooth resampled zoomed-in raster
5. Open the project's projection settings, active OTF, and change the CRS to one that is not used by the raster
6. Apply the OTF projection, notice the zoomed-in raster is now ignoring the bilinear resampling

History

#1 - 2016-03-15 01:26 AM - Mathieu Pellerin - nIRV

Interesting; looking into the debug messages, the reprojected raster *does* trigger a zoomed-in resampling, but fails to have any impact on the final rendering.

> src/core/raster/qgsrasterresamplefilter.cpp: 205: (block) [1ms] [thread:0xb019790] zoomed in resampling

#2 - 2016-03-15 03:19 AM - Mathieu Pellerin - nIRV

A few more bits.

This is the debug output when the raster is not reprojected:

> src/core/raster/qgsrasterresamplefilter.cpp: 135: (block) [0ms] [thread:0xb2c2390] width = 1539 height = 958 extent = 96.2596964460464477,25.4433501673136320 : 96.2860599454429718,25.4597609746052953
> src/providers/gdal/qgsgdalprovider.cpp: 1088: (capabilities) [0ms] [thread:0xb2c2390] driver short name = GTiff
> src/core/raster/qgsrasterresamplefilter.cpp: 150: (block) [0ms] [thread:0xb2c2390] xRes = 1.71303e-05 providerXRes = 0.000277778 pixelRatio = 0.061669 oversampling = 0.061669
> src/core/raster/qgsrasterresamplefilter.cpp: 161: (block) [0ms] [thread:0xb2c2390] oversampling 0.061669

note: the width and height values properly represent the width and height of my canvas

This is the debug output when the raster *is* reprojected:

> src/core/raster/qgsrasterresamplefilter.cpp: 135: (block) [0ms] [thread:0xb2c2390] width = 113 height = 68 extent = 96.2570833333333411,25.4420833333333363 : 96.2884722222222251,25.4609722222222246

src/providers/gdal/qgsgdalprovider.cpp: 1088: (capabilities) [0ms] [thread:0xb2c2390] driver short name = GTiff
src/core/raster/qgsrasterresamplefilter.cpp: 150: (block) [0ms] [thread:0xb2c2390] xRes = 0.000277778 providerXRes = 0.000277778 pixelRatio = 1 oversampling = 1
src/core/raster/qgsrasterresamplefilter.cpp: 161: (block) [0ms] [thread:0xb2c2390] oversampling 1

*note: the width and height values **do not** properly represent the width and height of my canvas*

#3 - 2016-03-15 06:05 PM - Mathieu Pellerin - nIRV

Tested and confirmed the issue is also present in QGIS 2.6.1, as well as QGIS 2.8.

#4 - 2016-03-16 02:12 AM - Radim Blazek

Oversampling (undersampling in this case) is calculated in QgsRasterResampleFilter::block() as requested resolution divided by raster resolution. Requested resolution comes from the next interface in the pipe, which is renderer if OTF is off or reprojector if OTF is on. The reprojector cleverly does not request higher resolution than the raster's resolution. It results in oversampling 1 and the resampler cannot do anything with that because there is no space to do resampling in requested block.

To fix this, the reprojector must request the resolution higher than the raster's resolution. I am not sure what is the best way to do it. In case of undersampling it should probably request the resolution calculated as input to reprojector regardless of raster resolution. That resolution cannot the reprojector get from the resampler (raster resolution multiplied by oversampling) because oversampling is calculated from requested block resolution.

A clean solution could be to use resampler's capabilities() without Size in reprojector. That would result in not limiting requested resolution but possibly in very high resolutions requested for CRSes which are not similar and maybe other negative side efects.

BTW, the reprojector was placed after the resampler because it is faster to do resampling than reprojection, IIRC.

Currently I don't have time to work on this. Marco Hugentobler, author of resampling, may have clearer ideas, added as watcher.

#5 - 2017-05-01 01:05 AM - Giovanni Manghi

- Regression? set to No
- Easy fix? set to No

#6 - 2018-11-21 07:19 AM - Frank Sokolic

I've also encountered this problem on 3.5.0-Master on Ubuntu. My workaround is to save the raster in the same CRS as the project CRS.

#7 - 2019-03-09 04:10 PM - Giovanni Manghi

- Resolution set to end of life
- Status changed from Open to Closed

End of life notice: QGIS 2.18 LTR

Source:
http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/