

Make the new parallel FilterEngine

- Author: Vadim Pisarevsky
- Link: The feature request
- Status: **Draft**
- Platforms: **All**
- Complexity: a few man-weeks

Introduction and Rationale

Since OpenCV 2.x there is **FilterEngine** class in **imgproc** module that is used to implement various separable and non-separable filters in OpenCV. The major reasons to design the FilterEngine many years ago were:

- provide convenient mechanism to handle image borders without using all-image `cv::copyMakeBorder`
- provide convenient mechanism to handle pipelines that include filtering operations. We need to say here that the goal has not been quite accomplished.

But there are a few problems with the class:

- it's quite complex code for what it does; from time to time we fix various memory access and other bugs in it
- it does not support threading
- it knows nothing about GPU/OpenCL
- for some complex non-linear filters it's not flexible enough. For example, bilateral and median filter are implemented from scratch without using FilterEngine.

So, now it's time to simplify the code and at the same time provide more scalable solution.

Proposed solution

- Instead of row-based engine that processes the whole image or the whole ROI sequentially, we need to have tile-based engine (where the tiles are not necessarily square).
- The engine should include `parallel_for` loop, where each thread processes a subset of tiles
- Border handling should be done automatically
- In-place processing should be implemented efficiently where appropriate, otherwise it should be emulated by using something like `cv::copyMakeBorder(src, temp);`
- For simplicity the question of pipelining filtering operations should be put out of scope.

Impact on existing code, compatibility

The `FilterEngine` class is not exposed since OpenCV 3.x, so there should not be any impact on the user code.

Possible alternatives

Very powerful alternative is Halide. Halide has been designed to do image filtering very efficiently. If we have Halide engine inside OpenCV, we probably do not need `FilterEngine`; we can implement most filters directly in Halide and get parallel and pipeline-able code.

References

TBD