Introduce a separate camera calibration module

- Author: Vadim Pisarevsky
- Link: [The feature request](#)
- Status: **Draft**
- Platforms: **All**
- Complexity: 1-2 man-months

Introduction and Rationale

The module [calib3d](#) from OpenCV 2.x and 3.x includes various functionality including camera calibration algorithms. At the same time, there is experimental [ccalib module in opencv_contrib](#) that includes some new camera calibration algorithms. There is also [aruco](#) module that implements detection of the fancy camera calibration rig. It makes sense to consolidate all this code in a dedicated module.

Proposed solution

Create a dedicated `calib` or `ccalib` module in the main repository with the refined camera calibration functionality from `calib3d` and `ccalib` (and may be `aruco`). All the functionality should be rewritten without using old C API. The obsolete non-robust code should be excluded. In particular, we now have many camera pose estimation algorithms (wrapped into `cv::solvePnP()`), but many of them do not work well and thus are excluded. The RANSAC variant of `cv::solvePnP()` also does not work well (links needed) and should be refined.

There is also very good [interactive camera calibration app](#). Perhaps, parts of it should be put to the module as well, and the app itself can be made a sample inside this module.

As a stretch goal, much more radical changes and new features can be put. For example, Rodrigues vector seems to be inferior to quaternions, so we should probably use the latter for rotation vectors.

Also, there are some very interesting self-calibration algorithms, such as [BAZAR](#) (is it the correct link?) that may also be put into the library. But that's probably a longer-term goals.

Impact on existing code, compatibility

Overall, the external API will change slightly, if any, but users will have to put some other include directives in their code (or keep using `#include "opencv2/opencv.hpp"`).

Possible alternatives

nothing really; we need to consolidate the camera calibration code.

References

TBD