Introduction

Transition to doxygen documentation format has been made in OpenCV 3.0, but many users still feel themselves uncomfortable with the new format. Following is the list of complaints gathered on user forums:

- no quick navigation
- links to detailed technical pages clutter interface (class list, namespace list, ...)
- code reference pages structure and size
- details level inconsistency between tutorials, pages and code reference
- some code examples are not usable (snippet compilation should be added)
- code samples (`samples/cpp`) are not documented or represented other way
- overall visual style (content width, background, fonts, etc.)
- no PDF documentation and cheatsheet
- no Python and Java documentation
- some tutorials are outdated (installation guides etc.)
- some functions are not documented This document proposes a plan of documentation restructuring, which is intended to improve user experience.

Restructuring

Currently the documentation is generated from public headers and tutorial pages. In the new structure there will be four main kinds of documentation pages:

Module page

Should describe module purpose, contain a list of implemented algorithms, contain a list of related tutorials and provide links to the programming interface documentation. Will be generated from markdown page located in `<module>/doc` folder. Page identifier should be in form `root_<module>` .

Algorithm page (article)

Should provide general information about one or several algorithms with links to articles and other external resources. Should provide list of functions and/or classes implementing described algorithm. Will be generated from other markdown pages located in `<module>/doc` folder.

Tutorial and guide page

Should demonstrate how to solve a practical task step-by-step or illustrate typical usage scenarios. Should contain annotated code snippets, references to algorithm pages (if needed), references to programming interface documentation. Can contain code snippets written in several languages (C++, Java and Python) with buttons to switch between them. Guides are similar to tutorials but do not relate to any specific module, for example installation guide, documetation writing guide, etc. Will be generated from markdown pages located in `<module>/tutorials` and `doc/guides` folders. Page identifier should have `tutorial_` , `guide_` or `tutorial_<module>_` prefix.

Code group page (programming interface documentation)

Should contain documentation for classes, functions and other programming interface elements. Will be generated from public module headers located in `<module>/include` folder. Doxygen commands `@defgroup` and `@addtogroup` will be used to define pages hierarhy and add symbols into them.

Other pages that do not fall into any of these categories such as main documentation index, bibliography, introduction and others will be generated from predefined or generated markdown pages located in `<build>` and `doc` folders.

Transformation steps

Initialize module pages

Create root page for each module with the description and links to the code groups. Update cmake scripts generating main index to use new pages instead of code groups to represent each module.

Reorganize tutorial pages

Move each tutorial to the corresponding module, page and images should go to the `tutorials` subfolder, code should go to the `samples` subfolder (TBD). Merge C++ and similar python tutorial into one if appropriate (TBD). Create deeper hierarchy if needed. Add links to the module root page. Modify cmake script which gather module tutorials if needed.

Reorganize guide pages

Move each tutorial which does not fall into any module to the `doc/guides` folder, add cmake support. Flatten hierarchy, add references to the main index page.

Extract algorithm descriptions

Take algorithm descriptions from code reference and tutorials and put them to separate pages. Create deeper hierarchy if needed. Add references to new pages where appropriate. Add links to the module root page.

Link code reference

Verify that each documented entity has links to corresponding algorithm description and/or tutorial page.

Verify completeness and correctness

Guides should be updated to match recent OSes, platforms and development tools, the steps should be generalized to minimize needs of future updates. C++ code snippets from tutorials should be extracted to compilable files. Python code snipets should be extracted to python tests and python samples.

Make cosmetic changes

Navigation tree. Links on the top bar. Stylesheets.

Discuss and implement other features and tasks

- Doxygen aliases for GitHub, Wikipedia and other often used URLs.
- Find pages which do not have poor content and create issues on the bug tracker.
- Generate images and text snippets for some tutorials during documentation build process.
- Annotate code samples (`samples/cpp` folder) or document them some other way.

Out of scope

Following problems need additional investigation:

- Python and Java documentation integration
- PDF documentation support

Pages structure diagrams

Main index page

```
* Introduction
* Main modules:
  * core (tutorials / code)
  * imgproc (tutorials / code)
  * …
* Contrib modules:
  * aruco (tutorials / code)
  * bgsegm (tutorials / code)
  * …
* Guides:
  * Installation for Windows
  * Installation for Linux
  * …
* Bibliography
```

Module page

```
Description
-----------
<...>
<link to code reference>
<...>

Algorithms
----------
* a1
* a2
* ...

Tutorials
---------
* t1
* t2
* ...
```

Algorithm page

```
Description (theory)
--------------------
<...>

Code and tutorials references
-----------------------------
<...>
```

Code reference page

```
<standard doxygen code reference>
```