PROJECT MANAGER MODULE

The project manager consists of all project related facilities. The creation, load and save of projects. Import and removal facilities also reside in this module. It is separated from its user interface which is another module. This is a result of the MVC (Model-View-Presenter) pattern that we have adopted.

The Project Manager Module consists of:

ProjectManager class: The user interface employs ProjectManager class. It provides wrapper functionality over other classes.

ProjectPresenter class: This class presents the data available in Project class to observers. This provides the protection of data in the Project class and a flexible interface to its contents.

Project class: Project class holds containers for contents which manifest in Content base class. Also meta data about projects is stored in Project class.

ProjectFactory class: ProjectFactory class consists of static methods that serialize/deserialize the project data in xml files. ProjectFactory class makes heavy use of Xerces library to parse and create XML files.

Content class: Content class is the base class encapsulating the basic functionality on contents. It encapsulates name, address, comment and content type data. Image type derives from this c lass.

Image class: Image class holds the ImageHandle class which provides the access layer to the physical image and provides wrapper functionality over this class.

Orthophoto, Dem, Mosaick classes: These classes derive from the Image class. Their difference from casual images is the addition of masks that give information about the real pixels of Mosaicks.

dL submodule: dL module provides the access to dynamic libraries. It filters libraries having a given set of functions. It has both Linux and Windows counterparts.

IMAGE PROCESSING MODULE

Image Processing module provides basic image processing capabilities such as convolution, brightness/contrast adjustment. It consists of:

ImageProcessor class: ImageProcessor class makes use of OpenCV and wxWidgets libraries to provide the image preprocessing facilities. All methods are static.

ImageFactory class: This is a helper class that provides conversion between OpenCV's and our image representation.

IMAGE REGISTRATION MODULE

Image registration module registers images successively. It outputs a homograpy matrix that is used by OkanMos class. It consists of:

OkanReg class: A base (reference) image is set and sensed images are added successively to the reference image.

Registration module makes use of OpenCV and the algorithms provided by Milsoft to achieve the desired result.

IMAGE MOSAICKING MODULE

Image Mosaicking module makes use registration module and creates a mosaick of the images. It consists of:

OkanMos class: Images are added successively to the mosaick. Linear illumination model is handled internally by OkanMos class.

OkanWFMos class: It is the second mosaicking module, provides the ability to mosaick images with associated world files.

ORTHOPHOTO GENERATION MODULE

Orthophoto generation module outputs orthophotos and anaglyphes. It takes images with camera parameters as inputs. It consists of:

OkanOrto class: If multiple images are to be supplied, they are given in a vector. After that, an output image can be taken with desired resolutions by the user.

USER INTERFACE MODULE

The user interface module provides access to other modules. User interface of Photokan is built upon wxWidgets library. It consists of:

ProjectManagerWindow class: This class is the backbone of the user interface module and provides access to ProjectManager class and to other interfaces.

ImageProcessorWindow class: This class provides access to ImageProcessor class.

ItemInfoPane class: This panel gives information about the selected item in the Project Tree.

MosaickerWindow, OPGWindow classes: These classes serve as containers for dynamic libraries which load their contents into the panels provided by these layers.

ImagePane class: This class encapsulates image representation facilities.