PIX4Dengine SDK V.1

FEATURE LIST

	Features	Advantages
INTERFACE	Python 3.6 interface for easy integration	Design your pipelines, scenarios, validations using a simple industry standard scripting language
INPUTS	Aerial –nadir & oblique– and terrestrial imagery	Process images taken from any aerial, terrestrial, manned, or unmanned platform
	Aerial and terrestrial images in .jpg format	Use images acquired with variety of cameras - from consumer-grade to highly specialized cameras
	Large frame images	Process images with resolution >100Mpx
	Images without IMU	Process image datasets with no IMU
	Images without geolocation	Process image datasets with no geolocation
	Multi-camera support in the same project	Create a project using images from different cameras and process them together
	RTK + IMU data support	Get faster and more robust calibration when using the Accurate Geolocation Pipeline
	Camera rig support	Process images using known rig relatives from multiple synchronized cameras
	Ground Control Points (GCPs)	Import GCPs to improve the absolute accuracy of a project
	Exposed camera parameter interface	Manage your camera parameters
	Known or custom reference coordinate system	Select EPSG code from known coordinate systems or define your own local system
PROCESSING	Processing templates	Choose between preset templates for the best processing options (nadir, oblique) or make custom option modifications. Currently limited to calibration and densification.
	Rapid quality check	Automatically classify the RGB dense point cloud into four groups: ground road surfaces, high vegetation, buildings and human-made objects
	Camera self-calibration	Optimize internal camera parameters, such as focal length, principal point of autocollimation and lens distortions
	Rolling shutter effect correction	Correct the warp of images taken with rolling shutter cameras (like GoPro, DJI Phantoms, etc.) to maintain accuracy even when flying fast and low
	Automatic point cloud densification	Produce a dense and detailed 3D point cloud, which can be used as a basis for DSM and 3D mesh
	Automatic point cloud filtering & smoothing	Use presets for point cloud filtering and smoothing options
	Machine-learning point cloud classification	Automatically classify the RGB dense point cloud into five groups: ground road surfaces, high vegetation, buildings and human-made objects
	Automatic DTM/DEM extraction	Remove above-ground objects from DSM and create a bare-Earth model
	Automatic brightness and color correction	Compensate automatically for change of brightness, luminosity and color balancing of images
	Processing area definition	Draw area to generate results inside specific boundaries
	Custom number of keypoints	Set the number of keypoints to filter noise or speed up processing
	Multiprocessor CPU	Increase the processing speed by leveraging the power of CPU cores and threads
	Radiometric processing and calibration	Calibrate and correct the image reflectance, taking the illumination and sensor influence into consideration
	AutoGCPs	Let engine find and mark your GCPs on images without any human intervention required
	Standard QA report	Use our predefined report format and assess the accuracy and quality of projects
RADIOMETRY	Radiometric adjustment interface	Make the vegetation indices more reliable and accurate by applying radiometric corrections
	Reflectance map	Generate an accurate Reflectance map and the preferred resolution as a basis of index maps
	NDVI map	Generate singleband and NDVI maps based on pre-defined formulas without user intervention
	Index formula definition	Create and save your own formulas choosing among each input band and generate custom index maps

OUTPUT RESULTS	**2D output results:**	Google tiles export in .kml and .html output formats
		Index maps (Thermal, DVI, NDVI, SAVI, etc.) in GeoTIFF
		Prescription maps in .shp format
		Nadir orthomosaics in GeoTIFF output format
		Orthomosaics from user-defined orthoplane in GeoTIFF output format
	2.5D output results:	Nadir DSMs in GeoTIFF format
		Nadir DTMs in GeoTIFF format
	3D output results:	Full 3D textured mesh in .ply, .dxf, and .fbx format
		Full 3D textured mesh in .obj format
		Tiled Level-of-detail (LoD) mesh in SLPK format
		Tiled Level-of-detail (LoD) mesh in OSGB format
		Point cloud in .las, .laz output format
		Point cloud in .ply .xyz output format
		Contour lines in .shp, .dxf, .pdf format
		Classified point cloud in .las and .csv format
		User-defined vector objects in .dxf, .shp, .dgn, and kml format

HARDWARE SPECS

CPU: Quad-core or hexa-core
Intel i7/ i9/ Xeon, AMD Threadripper

GPU: GeForce GTX 1070 and up
(compatible with OpenGL 3.2)

HD: Solid state drive (SSD)

OS: Windows (64 bits)
Ubuntu 18.04 (64 bits)

RAM: 8GB RAM (or more)