Version 3.1 (2018-06)

Python 3 Upgrade

We have upgraded our Python interpreter from Python 2.7 to Python 3.6 in this release. This requires our users to port their Python 2 scripts to Python 3 (if not already done).

To facilitate the porting, we provide the Python futurize library. For more information on porting see Preparing Python Scripts For Python 3.

On Windows, the Python 3 interpreter is now compatible with the official Python 3.6 interpreter. This allows to install binary packages from PyPi (the Python package index). A new `PythonPip` tool module facilitates the installation of Python packages from PyPi in MeVisLab.

Support for 10bit displays

MeVisLab now supports rendering with 10bit precision (using OpenGL). This requires a graphics card and monitor that support 10bit precision. The support has been tested on AMD/Barco graphics cards and on NVidia quadro graphics cards. We try to auto-detect the support for 10bit, but it can be enabled/disabled by the environment variable `MLAB_OPENGL_10BIT` as well. If enabled, MeVisLab automatically uses a 10bit capable OpenGL framebuffer (typically a RGB10A2 or RGBA16F buffer). By default, the `SoView2D` viewer will use a high-precision LUT if 10bit support is enabled. You can see if 10bit support is enabled in the first line of the MeVisLab log (after the graphics card details). It can also be checked in Python using `MLABSystemInfo.glSupports10Bit()`.

Physically-based Path Tracing (CUDA only)

The MeVis Path Tracer offers a state-of-the-art Monte Carlo Path Tracing framework running on NVidia GPUs (CUDA). It supports high-quality physically based rendering of volumes, meshes, thick lines and other geometry. For details, have a look at the MeVis Path Tracer Overview.

OpenVDB integration (sparse grid support)

The OpenVDB library (www.openvdb.org) has been integrated into MeVisLab. OpenVDB supports sparse voxel grids and has a unique level set

representation for meshes. It has been used to implement several new MeVisLab modules:

- `CSOToSurface` creates a WEM surface from a list of CSOs.
- `MarkersToSurface` creates a WEM surface from a marker list by rasterizing the markers as spheres.
- `OpenVDBLoad` loads a sparse grid as an ML image volume.
- `OpenVDBSave` saves an ML image volume as an OpenVDB sparse grid.
- `VoxelizeCSO`, `VoxelizeMarkers`, and `VoxelizeWEM` voxelize the respective structures into a reference image coordinate system.
- `WEMLevelSetBoolean` allows to perform boolean operations on closed WEM meshes using a level set representation.
- `WEMLevelSetFilter` allows to apply a kernel filter on a level set generated from a closed WEM mesh and remesh the result.
- `WEMLevelSetOffset` allows to enlarge or shrink a closed WEM using level sets.
- `WEMLevelSetRemesh` remeshes a closed WEM using a level set representation.

Fixes / Enhancements (MeVisLab)

- IDE
  - Added a context menu that allows editing of user scripts from the user script menu item itself.
  - Write the MeVisLab version to saved network files.
  - Support external links in note items.
  - 3D inspectors can zoom with the mouse wheel.
  - Prepend "./" to the displayed type of local macro modules.
  - The **Run In Separate Process** context menu entry shows a hint why it is disabled when it is disabled.
  - Connecting fields by dragging fields to the automatic panel was not undoable and also didn't mark the network as modified.
  - The MeVisLab Output Inspector now supports specialized inspectors for Python object sub-types.
  - Print an error message if a FieldListener has no field or no command given.
  - Fixed **CTRL**+**Tab** switching between networks in IDE and files in MATE.

    - Fixed inheriting module documentation if the module name is not the class name.
- TestCaseManager
    - Can avoid per-function result directory by setting `perFunctionResultDirectory` to `False` in the .def file of the test.
    - Added option to display test report viewer inside the main `TestCaseManager` panel.
- MATE (integrated editor)
    - Newly opened MATE documents are inserted after the tab from which they were opened.
    - MATE now downloads and installs Pylint version 1.8.4 by default.
    - Added option to run Pylint manually in MATE, and added a separate error check view.
    - One can specify a pylintrc file for the Pylint invocations in the preferences dialog.
    - Pylint check and Jedi auto-completion now regard the module's `importPath`.
    - Only highlight text or numbers on double-click.
    - Added (hidden) option to remove trailing spaces - this can be bound to a key shortcut in the shortcut manager.
    - Improved MATE startup performance by doing delayed package loading.
    - The replace-in-files dialog always writes system-specific line endings now.
- C++ API changes
    - Added new ML list fields: `ml::IntListField`, `ml::DoubleListField`, `ml::Vector2ListField`, `ml::Vector3ListField`, and `ml::Vector4ListField` (similar to the Inventor multi-fields).
    - The `mlError`, `mlWarning`, `mlInfo` macros do not need an error type argument anymore. If omitted, `ML_BAD_PARAMETER` is used.
    - `ml::Fields` don't have a copy constructor anymore.
    - Added new `getInterpolatedValue()` method on `ml::PagedImage` which allows to fetch a tri-linear interpolated value at an arbitrary location.
    - `ml::Rotation` now creates a 180° rotation for exactly opposite vectors (previously it did nothing).
    - The Inventor class `SoEvent` now has a `mouseButtonMask` property that contains the currently pressed mouse buttons.

    - Allow to override existing cursor shapes in Inventor viewers with SoViewerProxy::defineCursor.
    - Added a BLANK_CURSOR value for the cursor shape enum field in Inventor modules.
    - Renamed internal Inventor class SoLightPath to SoLightweightPath because the name was confusing.
    - The destructor of Inventor class SbProjector was made virtual.
    - Added a ML_EXPLICIT_FALLTHROUGH define to all platforms, which expands to `[[fallthrough]]` (C++17) on platforms where an implicit fallthrough would result in a warning at compile time.
    - Added new method ml::OpenGL::isMesaSoftwareRenderer() to check if MeVisLab runs with an OpenGL software renderer.
- Controls
    - Added a PythonTextView control for editing Python code in modules. It supports auto-completion of the underlying module context, context sensitive help and an extended context menu. Many modules that offer Python scripting now make use of this control (RunPythonScript, PythonArithmetic, ...).
    - ItemModelView can show a color and a checkbox in the same column.
    - Implemented an alternatingRowColors attribute on ItemModelView.
    - Added alternateBase (needed for alternatingRowColors), toolTipBase, and toolTipText color roles to MDL style.
    - Added clickedColumnField attribute to ItemModelView control, which writes the column index onto which the user (double-)clicked to the given field.
    - Introduced ComputedAttribute to ItemModelView control. This allows to compute a virtual attribute from other attributes using the same expression parser as, e.g., the dependsOn mechanism.
    - Implemented dependsOn/visibleOn on the Row element of the Table control.
    - Added new listview mode on TabView control. This facilitates the creation of preferences-style panel with a listview instead of a tabbar to select tabs. The listview can be made hierarchical by using the tabHierarchy tag on TabViewItems.
    - Fixed initial enabled state of TabView with only one child.
    - Fixed that dependsOn of ButtonGroupControl items always affected the first item.

    - When a parent checkbox in an ItemModelView control switches several child checkboxes, these are collected into one ItemModel update event.
    - Removed slicerScale attribute on Slider control (it didn't work anyway).
    - The **Space** key is not titled "Any" anymore in the EventFilter control. Also added the numerical "keyCode" info.
    - Call shouldCloseCommand if a modal dialog panel is closed with the **Escape** key.
    - Use system proxy settings by default in WebView control if no proxy is specified in the Preferences dialog of MeVisLab.
    - The Field control now recognizes the comboItems attribute.
    - Fixed wrong auto-alignment groups on the Grid control.
      Previously all Grid children got the same width, regardless of their column.
    - The **More** button on a Field control now shows a non-editable dialog if the control is non-editable.
    - Fixed problem with auto-repeat in step-buttons of number widget if a dialog is opened on value change.
- Scripting
    - New field wrapper classes for the ML list fields and some Inventor multi-fields: MLABIntegerListField, MLABDoubleListField, MLABVector2ListField, MLABVector3ListField, and MLABVector4ListField.
      Since previously these fields were wrapped with MLABStringField you will now get an info message if you use the old value property which has a string type. You should rather use the new listValue property or stringValue()/setStringValue().
    - Added mlab_projects Python meta package for importing Python modules from MeVisLab module projects. (Not supported in auto-completion!)
      Use

      ```
      from mlab_projects.MySuperDuperProject import Utilities
      ```

      in Python code to import the Python module Utilities from file $(MyPackage)/Projects/MySuperDuperProject/Modules/Scripts/python/Utilities.py.

    - Added MLAB.cryptStringUTF8()/MLAB.decryptStringUTF8() scripting methods with UTF-8 support.
    - Added setVr method to MLABMutableDicomTag.
    - Python object __object__ in the module context can now be any Python object to be visible through ctx.object(). Previously the auto-conversion of Python sequences or maps caused a conversion to a different object type.
    - The field associated with a control can now be accessed with the field() method.
    - MLABProcess.runCommandInConsole now has an option to wait until finished.
    - Added new scripting methods on macro modules to determine the sub-type (local, ad-hoc).
    - Added scripting method ctx.shouldAvoidSideEffects(), which returns true if a module is created just for editor auto-completion.
    - MLABModule::addModule does not print package dependency warnings anymore.
    - Print an error message if a call on a module could not be performed because there was no script context.
    - Using the MLABHttpDownload scripting class failed for large downloads in the GB range.
    - Added new scripting method MLABFileManager.setApplicationSupportUmbrellaDirectoryName() so applications can select another name than 'MeVis'.
- Other fixes and improvements
    - Improved CUDA building support, support CUDA 9.
    - Connecting a bool field to float/double fields correctly sets 0 or 1 (only worked for Inventor fields before).
    - Do not ignore legacy values in lazy loading modules.
    - Save persistent flag of fields if required in lazy loading modules.
    - Worker processes (of remote modules) ignore the Logfile variable now.
- Third-party libraries
    - Python
        - MeVisLab's Python distribution was updated to Python 3.6.4.
        - Added pyodbc Python ODBC bridge, version 4.0.21 (Windows only, typically used with sqlalchemy).
        - setuptools was updated to version 38.5.2.

        - NumPy was updated to version 1.14.2.
        - pip was updated to version 9.0.1.
        - coverage was updated to version 4.5.1.
    - Updated Qt to version 5.6.3.
    - Removed ospray library from ThirdParty, but re-added current embree library, version 2.17.0.
    - boost.random can be used in own projects now.
    - Use other method to determine MAC address for UUID generation in dcmtk to avoid certain crashes.
    - The storescu tool supplied by dcmtk library now supports on-the-fly compression.
    - The OpenSSL library was updated to version 1.0.2o.
    - Added OpenMP runtime for MacOS, version 6.0.

Fixes / Enhancements (Modules)

- SoView2D and extensions
    - Added module `SoView2DCine` that allows more control over `SoView2D`'s cine mode.
    - The new `SoView2DCurrentState` module returns the visible portion of an image for use with `SoView2DRectangle`.
    - More control over ruler color in `SoView2DRuler` and `SoView2DAnnotation`.
    - Added a `stepSize` field to `SoView2DSlicer`.
    - `SoView2DOverlayMPR` supports left-handed coordinate systems now.
    - `SoView2DDrawVoxels3D` can draw a brush preview now.
    - The `SoView2DMarkerEditor` module was moved to its own library.
    - `SoView2D` key commands are always available when the mouse is over the viewing area, not just over the image area.
    - `SoView2DExtensions` also show the "Alt" modifier option in their panel.
- Other Open Inventor modules
    - Added new modules for menus in OpenInventor scenes: `SoFixedMenu`, `SoBorderMenu`, and `SoPopupMenu`, with `SoMenuItem`. In contrast to `SoView2DMenu` these support Managed Interaction and are more flexible.
    - Added new modules to query the OpenGL state: `SoGLClearError`, `SoGLColorDepthInfo`, `SoGLGet`, `SoGLStateInfo`.

    - Introduction of the new Managed Interaction Inventor draggers. Have a look at the MIPlaneDragger macro module for how to use them.
    - Added SoMetaInformation1d and SoMetaInformation1i modules.
    - Added a selectOnlyOnClick option to SoSelection and SoSelection2, which doesn't perform the selection if the mouse was moved while holding the mouse button.
    - Added a keepViewportWhileDragging option to SoViewportRegion.
    - Improved compatibility of Managed Interaction modules with classic event handling modules in Inventor scenes.
    - Added support for sRGB interpolation in SoLUTEditor and LUT base classes.
    - Allow to add control points by dragging from the borders in the SoLUTEditor.
    - Allow to choose slab start/center/end for the intersection (and an optional offset) in SoRenderSurfaceIntersection.
    - SoCrosshair can render thin cylinders instead of lines now.
    - Added ADD_BEFORE_INORDER modification type to SoShaderPipelineFunction.
    - Various changes to SoVascularSystem, e.g.:
        - Can work on a copy of the input graph.
        - Enhanced highlighting capabilities.
        - Can choose between integer and float LUT evaluation.
    - Support local transformations in SoDrawInstanced.
- CSO
    - Added functions to CSO scripting wrappers for seed point selection.
    - CSOConvertToImage got an outputTypeMode field to grant more control over the voxel type of the output image.
    - SoCSO3DRenderer, So3DMarkerRenderer: Added functionality to only render CSOs/markers of selected timepoint.
- WEM
    - Fixed an endless loop in WEMSubdivide.
    - SoWEMRenderer can cache WEM patches now (e.g. to quickly switch between time points).
    - Added module WEMRayIntersect and scripting wrapper WEMBoundingVolumeHierarchy for a fast ray intersection with WEM surfaces (based on the embree library).

    - WEMIsoSurface will cope with images with left-handed coordinate system now.
    - The WEMBulgeEditor uses different values (signs) in the PVL for preview and interaction.
    - Added global normal output to MarkerListToWEMPlane module.
- ML modules
    - Added modules QPainterImage and QImageToML for ML image generation through Qt classes.
    - Added the ChromaKey module (for green screen handling of video images).
    - Replaced Type(De)Composer8|16|32|64 with Type(De)ComposerN, which have a dynamic number of visible inputs/outputs.
    - Added fillValue field to ConcatImages.
    - The BaseClear module got inputValid/outputValid fields.
    - Added an option useKeyFrameDistanceForSliceThickness to the MPRPath module.
    - Fixed automatic voxel size and bounding box calculation of the MPR modules.
    - Added Minimum and Replace modes to the AccumulateImage module.
    - The new GraphToVolume module allows to rasterize a ml::Graph to a volume.
    - ImageSave can now save RGB floating-point TIFF images with a linear (gamma=1) sRGB ICC profile.
    - ImageCompare now prints an ML_CALCULATION_ERROR instead of an ML_PROGRAMMING_ERROR if images differ.
    - We did a clean-up of MLVesselGraph library, some old modules were removed.
    - Avoid crashes in GVR modules when the image min/max values are not correct, i.e., there are voxel values outside the stated range.
    - Make CurveToHistogram always correctly update its output.
    - Fixed auto-updating of maximum volume limit in RegionGrowing module.
    - Fixed property extension loading in MLImageFormat.
- Macro modules
    - Created new utility module ExportImagesAsSlices.
    - A new PythonPip tool module allows to install Python packages from PyPi.

    - Lots of fixes and improvements to the AnimationRecorder.
    - Added undo/redo to DrawVoxels3D.
    - GVROrthoOverlay has a blendMode field now.
    - Speed-up loading of file list in ImageLoadMulti.
    - Renamed CommandNotifyer to CommandNotifier and improved panel.
    - Removed the outdated ImageViewer application from the SDK.

Modules and Libraries Provided by Fraunhofer MEVIS

Modules

- Additions
    - New module WalkSiblingDirectories that can be used to compare the contents of two folders file by file.
    - New module SoMIMouseOver allows to detect mouse over and click on child geometry using Managed Interaction.
    - New module SoMIRotateCameraPlaneDragger allows for rotation in the camera plane using Managed Interaction.
    - New module SoMITranslateCameraPlaneDragger allows for 2D translation in the camera plane using Managed Interaction.
    - New module SoView2DInfo that displays SoView2DExtension infos.
    - New module DynamicLayout that creates a layout using an MDL file and embeds viewers found in the connected Inventor graph.
    - New module SoActionFilter that allows to select which actions are applied to the child graph.
    - New module InventorGuard that combines SoIdentifier and SoActionFilter such that the child graph can not be added directly to the scene.
    - New module SoNodeFilter that filters the Inventor scene graph based on node names or types.
    - New module ComposeBoundingBox that calculates a composed bounding box of two aligned images.
    - New module ApplyOrthoOrientation, extending NormalizeOrthoOrientation with an additional "FromReference" mode, which applies the same ortho-orientation (transversal, sagittal, or coronal) as a given reference image.
    - New module OrthoCombination, does a combination of three images in different orthogonal orientations into a single one with a

defined orientation, which puts the images from the different inputs in separate sub images in the c, t, or u dimension. This allows for convenient combination using modules requiring a single input image (e.g. OrthoProjection or kernel filtering).

- New module ImageFromFile that loads a single image from a directory or file of (almost) any image file type supported by MeVisLab (basically an extension to LoadAny, internally making use of it if everything else fails).
- New module ApplyGlobalModalityLUT, allowing for normalizing DICOM data containing either a single frame-unspecific (i.e. "shared" or "global") or multiple frame-specific RescaleIntercept or RescaleSlope values to a single, frame-unspecific, user-defined intercept, slope, and data type. (combining ApplyDicomPixelModifiers and DicomRescale).
- New module BoundingBoxInReferenceSystem that computes the smallest SubImage of input1 that would contain all (reformated) voxels of image0..
- New module LUTColorAtIndex that gets the color at the given LUT index.
- New module LUTColorAtValue that gets the color at the given LUT value.
- New module SoView2DSetPosition allows to set the slice index of connected SoView2Ds such that a given position is matched best.
- New module BoundingBoxListener notifies changes of the Inventor scene boundingbox. Optionally an object selection is used.
- New module RotateAtTarget allows to compute the orientation for a given position, target point and up-vector.
- New module ImageAlignedBoundingBox calculates the image-aligned bounding box of the input image in world coordinates.
- New module WorldAlignedBoundingBox calculates the world-aligned bounding box of the input image in world coordinates.
- New module ApplyTransformationMatrix allows to apply a transformation given as matrix on the input image.
- New module GVRContourOverlay that draws a contour of a mask image using a GVR secondary volume.
- New modules PointSender/PointReceiver that allow to send point lists over a communication channel.

    - New module FilterCurveList to filter a CurveList by curve properties such as title or style.
    - New module SetMarkerCTUCoordinates to set non-spatial coordinates for a whole XMarkerList.
    - New module TranslateXMarkers to translate all positions in an XMarkerList.
    - New module ExtrapolateXMarkersPolynomial to extrapolate an XMarkerList based on a polynomial fit.
- Improvements & extensions
    - Added simpler, bulge-like node movement to WEMExpandToMarkers.
    - The macro module CSOImageStatisticsOverTime now allows more than one input CSO.
    - Renamed NormalizeOrientation to NormalizeOrthoOrientation.
    - ImageCache macro and CacheManager python module now support progress output.

Updated third-party libraries in FMEWork/ThirdParty

- Updated CTK-CLI python package to version 1.4.
- Updated pydicom python package to version 1.0.1.
- Removed mock python package.

Windows Edition

- The Mesa OpenGL fallback driver has been updated to version 17.5.3.
- HiDPI support has been improved, especially for setups with multiple screens of different resolutions.
- Fixed a small color conversion bug in the video generation with OpenCV under Windows - videos were slightly too dark.

macOS Edition

Note

This release requires at least OS X 10.11.5 (El Capitan) and Xcode 8.2 for development.

- Full Support of macOS 10.13 (High Sierra)
- Worker and Background Processes started from the IDE will show up in the Dock and allow network inspection via Context Menu Entry 'Show IDE'

- Support of OpenMP development with Xcode 9.0 and newer (OpenMP runtime is included with MeVisLab)
- Symbol visibility is restricted to explicitly exported symbols
- ADK
    - Support of INSTALLER_MACX_PREINSTALL_COMMAND & INSTALLER_MACX_POSTINSTALL_COMMAND for executing shell commands during the installation of pkg installers (aka guided installers)
    - Support of MACX_SIGN_INSTALLER, MACX_SIGN_CHECK_IDENTITY_BEFORE_USE, MACX_INSTALLER_SIGN_IDENTITY_NAME to sign pkg installers

Application Builder (previously known as ADK)

- Not showing diagnostic info messages for MeVisLab console applications.
- Do not display certain dialogs if MeVisLab is run in batch mode.
- When setting the preferences variable `ShowFieldHelpInApplications` the field help is shown in applications.
- Can specify installer icon under Windows with `INSTALLER_WINDOWS_ICON` (and added a new default installer icon).
- Removed support for putting .py and MDL files into a zip file in standalone installers.
- Overhaul of the multi-language support in MeVisLab:
    - Calling the language tools from the public SDK works now.
    - Can open Qt Linguist from the context menu.
    - Can set the language from the module's context menu.
    - Better detection of translatable strings.
    - Translations will be applied to web panels.
- Make it possible to use the Unicode plugin in the NSIS installer with `INSTALLER_CUSTOM_NSIS_TOP_INCLUDE`.