
Matrox Imaging Library (MIL) 10.0
MIL 10 Processing Pack 3
Release Notes
(c) Copyright Matrox Electronic Systems Ltd., 1992-2018.

Main table of contents

- Section 1 : Differences between MIL 10 Processing Pack 3 and MIL 10 Processing Pack 2 with Update 63
- Section 2 : Differences between MIL 10 Processing Pack 2 with Update 63 and MIL 10 Processing Pack 2
- Section 3 : Differences between MIL 10 Processing Pack 2 and MIL 10 Processing Pack 1
- Section 4 : Differences between MIL 10 Processing Pack 1 and MIL 10
- Section 5 : Differences between MIL 10 and MIL 9 Processing Pack 2 with Update 56
- Section 6 : Differences between MIL 9 Processing Pack 2 with Update 56 and MIL 9 Processing Pack 2
- Section 7 : Differences between MIL 9 Processing Pack 2 with Update 45 and MIL 9 Processing Pack 2
- Section 8 : Differences between MIL 9 Processing Pack 2 and MIL 9 Processing Pack 1
- Section 9 : Differences between MIL 9 Processing Pack 1 and MIL 9
- Section 10 : Differences between MIL 9 and MIL 8 Processing Pack 4

- Section 1: Differences between MIL 10 Processing Pack 3 and MIL 10 Processing Pack 2 with with Update 63

Table of Contents for Section 1

- 1. Overview
- 2. New functionalities and improvements
 - 2.01 MIL processing specific examples
 - 2.02 Classification module
 - 2.03 Blob module
 - 2.04 Calibration module
 - 2.05 SureDotOCR® module
 - 2.06 Registration module
 - 2.07 Pattern Matching module
 - 2.08 Model Finder module
 - 2.09 Metrology module
 - 2.10 3dMap module
 - 2.11 Code Reader module
 - 2.12 Measurement module
 - 2.13 String Reader module
 - 2.14 Color module
 - 2.15 Primitives
 - 2.16 Graphics
 - 2.17 General
- 3. Deprecated functionalities
 - 3.01 OCR module
 - 3.02 Registration module
 - 3.03 Code Reader module
 - 3.04 Blob module
 - 3.05 Graphics
- 4. Fixed bugs
 - 4.01 Blob module
 - 4.02 Calibration module
 - 4.03 SureDotOCR® module
 - 4.04 Pattern Matching module
 - 4.05 Model Finder module
 - 4.06 Metrology module
 - 4.07 3dMap module
 - 4.08 Code Reader module
 - 4.09 Measurement module
 - 4.10 Edge Finder module
 - 4.11 Registration module
 - 4.12 String Reader module
 - 4.13 Primitives
 - 4.14 Graphics
 - 4.15 Utilities
- 5. Known limitations
 - 5.01 Classification module
 - 5.02 Model Finder module

1. Overview

- MIL 10 Processing Pack 3 includes all the features of MIL 10 Processing Pack 2 with Update 63.
In addition, MIL 10 Processing Pack 3 includes new processing functionalities, performance optimizations, and general improvements such as:
 - . new: Registration photometric stereo context
 - . new: Model Finder rectangle shape finder context
 - . new: Metrology features and tolerances for 3D profile analysis
 - . new: Code Reader 1D code type detection
 - . new: Code Reader training from sample images
 - . new: Classification module

- . new: Primitives such as edge preserving filters,
- . and many more to discover!

- MIL 10 is the minimum requirement for all upcoming Processing Packs until the next major release.
- Note: MIL 10 Processing Pack 3 does not support the MIL GPU system for image processing.
- Windows' automatic 8.3 file name creation needs to be enabled in order for the MIL installer to access the temp folder when the user name contains a space. This option allows Windows to create short file/folder name aliases for ones with long names for programs, such as the MIL installer, that don't support spaces in the file/folder names. Alternatively, the MIL installer needs to run from a user account that belongs to the administrators group and has no spaces in it. Note that the same applies for uninstalling MIL.
- The required Visual C++ 2017 Redistributable needs the presence of KB2919442 and KB2919355. These will need to be obtained and applied before installing this Processing Pack.
- The MIL Processing Pack requires a Windows installation that supports device drivers with SHA-2 digital certificates. Consequently, some Windows 7 installations will require that a Windows Monthly Rollup be applied before the MIL update can be installed.
- Building the MIL examples using Visual Studio 2015 or 2017 also requires the presence of Windows SDK version 8.1, which is installed from the Visual Studio setup.

2. New functionalities and improvements

2.01 MIL processing specific examples

- New MIL 10 Processing Pack 3 examples!
- Note that MIL 10 Update 52 is needed to get the images required for several examples, in particular for the 3D processing examples.

2.02 Classification module

- New: a classification module to predict the category of an input image using a restored pre-trained convolutional neural network (CNN).

2.03 Blob module

- New: MblobTransform() function to apply operations on an existing blob result object.
- New: MblobTransform(M_CONTIGUOUS_LABELS) to perform the relabeling of blobs.
- Improvement: MblobFree() has been optimized in speed when calculation has been performed on large images containing lots of runs.
- Improvement: the convex hull calculation has been accelerated.
- Improvement: MblobGetResult() will no longer return uninitialized values but report an error for non-calculated feret features.
- Improvement: MblobMerge() has been optimized for speed and memory usage.
- Improvement: the M_PRINCIPAL_AXIS_ANGLE numerical stability has been improved for perfect symmetric blobs.
- New: M_MAX_LABEL_VALUE can now be retrieved using MblobGetResult().
- New: The M_BLOB_IDENTIFICATION_MODE can now be retrieved using MblobGetResult.

2.04 Calibration module

- New: McalTransformCoordinate3dList() supports point cloud containers and 3D reconstruction contexts for world-to-world conversions.
- New: McalGrid() now supports regions of interest (ROI).
- Improvement: McalList() with M_DISPLACE_CAMERA_COORD or with M_DISPLACE_RELATIVE_COORD will no longer require setting the principal point. In 3d mode and after a successful call to McalGrid(M_FULL_CALIBRATION), it is now no longer required to specify the principal point before a call to McalList(M_DISPLACE_*).
- Improvement: the detection of chessboard grids has been improved.

2.05 SureDotOCR@ module

- New: DOT_FONT_SEMI_5x9.mdmrf font file has been added following the "SEMI OCR OUTLINES AUX015-1106" specification.
- New: MdmrGetResult() has new results to retrieve the bounding box of a read string.
- New: MdmrDraw(M_DRAW_STRING_BOX) can draw the string bounding box.

2.06 Registration module

- New: Photometric stereo is now supported to compute the albedo and the curvature image from a set of images acquired with light from different directions.

2.07 Pattern Matching module

- Improvement: M_SEARCH_ANGLE_ACCURACY and M_SEARCH_ANGLE_TOLERANCE can now be set to values smaller than 0.1 degree.
- Improvement: MpatDefine(M_REGULAR_MODEL+M_CIRCULAR_OVERSCAN) now reports a better error message when the model definition is too close from an image's border.
- New: M_HAS_DONT_CARE_MASK can now be inquired to retrieve whether a don't care mask is set for a model.

2.08 Model Finder module

- New: a rectangle shape finder is now supported!
- New: a segment shape finder is now supported!
- Improvement: ellipse finder will now return the closest angle to angular

- mid-value when both the angle and its opposite are valid.
- Improvement: MmodFind() with M_SHAPE_CIRCLE and M_SHAPE_ELLIPSE contexts is now optimized for multi-core architectures.
- New: M_HAS_TARGET_EDGES_SAVED can now be inquired to retrieve if the target edges have been saved.
- New: M_RESULT_TYPE can now be inquired to retrieve the type of the result object (i.e. M_GEOMETRIC, M_GEOMETRIC_CONTROLLED, M_SHAPE_CIRCLE, M_SHAPE_ELLIPSE, or M_SHAPE_RECTANGLE).

2.09 Metrology module

- New: two edgel features can be now compared and their differences can be extracted.
- Improvement: Radial fit of a segment within a ring sector region has been improved.
- New: the ordering of the edgels for an edgel feature can now be specified.
- New: the area tolerance between two edgel features can now be calculated.
- New: the edgel features can now be resampled.
- New: copy of edgel feature can now be done for measured and constructed features.
- New: the area under the curve tolerance is now supported.
- Improvement: the straightness tolerance has been optimized for speed.
- Improvement: the ring sector region definition has been slightly improved.
- New: inquires to retrieve the information about an external edgel feature added using MmetPut().
- New: the drawing of labels for both measured and constructed edgels is now supported.
- New: the Max of min distance tolerance between two features is now supported.
- New: the construction of the closest point to a virtual infinite point at a specific direction is now supported.
- New: M_PERIMETER_SIMPLE tolerance is now supported to compute the perimeter of a feature.
- New: M_AREA_SIMPLE tolerance is now supported to compute the area of a feature.
- New: M_PERIMETER_CONVEX_HULL tolerance is now supported to compute the perimeter of the convex envelope of a feature.
- New: M_AREA_CONVEX_HULL tolerance is now supported to compute the area of of the convex envelope of a feature.
- Improvement: M_CLOSEST point construction now supports more combinations of features.
- Improvement: M_MAX_DISTANCE point construction now supports more combinations of features.
- New: the closest point construction now supports an optional reference angle constraint.
- New: the distance tolerances now support directional measure by providing an optional reference angle constraint.
- Improvement: the bisector between two lines can be controlled to construct the feature according to the largest angular sector, the smallest angular sector, or the feature ordering.
- New: measured edgels or edgels during a copy operation can now be post-processed using a median, average, or Gaussian filter.
- Improvement: the furthest point between two arc constructions has been optimized for both speed and accuracy.
- Improvement: closest/furthest constructed point is now supported from multiple features.
- Improvement: new feature, M_WARNING status, is now returned when a feature definition is valid but its construction is not possible. For example a segment constructed from two points, while the two points appear to be identical.
- New: the angularity tolerance can now be set between three points.
- New: the length tolerance can now be set for an edgel feature.
- New: parameterized best fit methods are now proposed for best fitted features.

2.10 3dMap module

- New: M3dmapPut() now supports M_APPEND to append points to an existing point cloud.
- New: M_ALIGN_COMPLETED status can now be inquired using M3dmapGetResult() to retrieve whether the M3dmapAlign() operation successfully completed.
- New: M3dmapExport() function to export a point cloud is now supported.
- New: M3dmapSetGeometry() now supports point cloud containers as input when using M_FIT_POINT_CLOUD.
- New: M3dmapSetGeometry() supports generating and fitting hemispheres.
- New: M3dmapPtCldCalculate() can now calculate the 3d distance between a point cloud and a geometry (plane or sphere center).
- New: M3dmapCopy() to copy or to move a source point cloud to a destination point cloud.
- New: M3dmapCopyCond() to copy or to move a source point cloud to a destination point cloud based on a given condition.
- Improvement: M3dmapArith() and M3dmapStat() now support source depth maps with different calibrations along the Z axis.
- New: M3dmapStat() now supports M_DEVIATION_MIN, M_Z_MIN and M_Z_MAX.

2.11 Code Reader module

- New: McodeDetect() is now supported to automatically detect codes types in an image.
- New: McodeTrain() is now supported to automatically optimize context controls from a set of images.
- New: McodeRead() now supports Character set Extended Channel Interpretations (ECIs) for M_DATAMATRIX, M_AZTEC, M_QRCODE, M_MAXICODE, M_PDF417, M_TRUNCATED_PDF417 and M_MICROPDF417.
- Improvement :McodeGrade() will no longer rarely exhibit slight result differences between grading directly or from a read result for 1d barcode.
- New: McodeGetResult() and McodeGetResultSingle() now supports the result type M_IS_ECI.

- Improvement: McodeGetResult() will now return more accurate and stable bounding box corner results for UPC/EAN code types with add-on.
- New: results M_TOP_LEFT_X, M_TOP_LEFT_Y, M_TOP_RIGHT_X, M_TOP_RIGHT_Y, M_BOTTOM_LEFT_X, M_BOTTOM_LEFT_Y, M_BOTTOM_RIGHT_X, M_BOTTOM_RIGHT_Y, M_POSITION_X and M_POSITION_Y can now be combined with M_2D_COMPONENT to retrieve positions for the 2D component of a composite code.
- New: McodeGetResult() and McodeGetResultSingle() now support the M_TYPE_STRING combination flag.
- Improvement: the threshold mode M_ADAPTIVE is now supported with M_4_STATE codes.
- Improvement: McodeRead() is more robust reading skewed M_4_STATE codes.
- Improvement: McodeRead() is more robust reading aliased M_4_STATE, M_PLANET and M_POSTNET codes.
- New: McodeGetResult()/McodeGetResultSingle() now support retrieving the read results after a call to McodeGrade().
- New: M_GRADE_QUIET_ZONE flag can enable/disable grading the quiet zone for M_QRCODE, M_MICROQR_CODE, and M_DATAMATRIX.
- Improvement: McodeRead() robustness has been improved when M_CHECK_QUIET_ZONE is set to M_DISABLE.
- Improvement: McodeRead() will now prioritize codes with valid quiet zone, even when the M_CHECK_QUIET_ZONE is set to M_DISABLE.
- Improvement: McodeRead() is now more robust regarding the dimension of the spaces for Industrial-25 codes.
- New: M_CELL_NUMBER_X_MIN, M_CELL_NUMBER_X_MAX, M_CELL_NUMBER_Y_MIN, M_CELL_NUMBER_Y_MAX controls are now available for all 2D codes.
- New: M_MAXICODE, M_PDF417, M_TRUNCATED_PDF417, M_MICROPDF417 now support the string format M_RAW_DATA.
- Improvement: McodeRead() and McodeWrite() now support Version 18 of the GS1 General Specifications.
- New: McodeInquire() can now retrieve the list of the supported code types using M_SUPPORTED_CODE_TYPES.
- Improvement: McodeRead() has been optimized for speed when reading multiple linear code models.
- Improvement: McodeGrade() has been optimized for speed for DataMatrix codes.
- Improvement: McodeGrade() robustness has been improved for DataMatrix codes.
- New: McodeControl() now supports M_CODE_SEARCH_MODE (M_FAST, M_BEST) to return either the N first decoded code or the N best codes among all found.
- Improvement: McodeRead() with M_THRESHOLD_MODE set to M_GLOBAL_SEGMENTATION has been improved for linear code.
- Improvement: McodeRead() has been improved for 1d codes in noisy image when M_THRESHOLD_MODE is set to M_ADAPTIVE and the M_MINIMUM_CONTRAST is close or equal to the image contrast.
- Improvement: McodeRead() has been improved for linear codes at angle.
- Improvement: McodeGrade() robustness has been improved for Code 128 codes as well as for 2D codes.
- Improvement: McodeRead() has been improved to reduce the number of false positive solutions when reading M_PHARMACODE codes.
- Improvement: The accuracy of the corners of linear codes has been improved in the presence of distortion.
- Improvement: McodeRead() and McodeGrade() speed have been improved for QR Code, Micro QR Code or Aztec when M_DECODE_ALGORITHM is set to M_CODE_DEFORMED.
- Improvement: McodeRead() robustness has been improved for barcode with poor modulation in blurred images when the M_SCANLINE_HEIGHT is set to a value different than the default value.

2.12 Measurement module

- New: the M_SUB_REGIONS_NUMBER_USED result can now be retrieved using MmeasGetResult().

2.13 String Reader module

- Improvement: MstrRead() is now able to read more than one string along the same line.

2.14 Color module

- New: McolInquire(M_CONTEXT_TYPE) to distinguish between a M_COLOR_MATCHING context and a M_COLOR_CALIBRATION_RELATIVE context.

2.15 Primitives

- New: MimConvolve() with a M_LINEAR_FILTER_IIR_CONTEXT context to perform image filtering using infinite impulse response linear filters such as Deriche or Shen.
- Improvement: MimMorphic(), with M_DILATE/M_ERODE and with structuring element filled with 0s, has been optimized for speed.
- Improvement: MimDilate() and MimErode() have been optimized further for speed.
- Improvement: MimHistogramEqualize() has been optimized further for memory usage with 32-bit sources.
- New: MimArith() now supports M_ABS_SUB, M_AVG, M_MAX_ABS, and M_MIN_ABS operations.
- New: MimFilterAdaptive() function to perform edge preserving image filtering techniques such as the bilateral filter.
- New: MbufBayer() now supports a M_ADAPTIVE_FAST demosaicing operation.
- New: MimRemap() function to perform image intensity remapping.
- Improvement: the accuracy of MimConvert() HSL conversions has been improved.
- New: M_REPLICATE overscan mode for the neighborhood operations is now supported.
- Improvement: MimRotate(), with rotations multiple of 90 degrees, has been optimized further for speed.
- New: MimConvolve() predefined FIR convolution kernel M_LAPLACIAN_ISO_8 is now supported.
- Improvement: MimTranslate() has been improved in accuracy.
- New: MimTranslate() now supports the M_NEAREST_NEIGHBOR and the M_BICUBIC interpolation modes.

- New: MimConvert() now supports HSV conversions.
- New: MimArith() now supports the M_SUB_CONST_ABS operation.
- Improvement: Neighborhood operations better support periodic mirror overscan - that is when the neighborhood operation requires an overscan dimension larger than the image dimension.

2.16 Graphics

- New: it is now possible to hold the Shift key to restrict the interactive translation along the X and Y axes.
- New: interactive translation can now be configured using MgraControlList() with M_MODE_TRANSLATE, M_MODE_TRANSLATE_SHIFT, M_MODE_TRANSLATE_CTRL and M_MODE_TRANSLATE_ALT.
- Improvement: MgraRect() and MgraRectFill() will no longer create a rectangle with negative width or height. Rectangle is now defined by its top left position with positive width and height.
- New: it is now possible to hold the Alt key to interactively rotate graphics by a multiple of 15 degrees.
- New: interactive rotation can now be configured using MgraControlList() with M_MODE_ROTATE, M_MODE_ROTATE_SHIFT, M_MODE_ROTATE_CTRL, M_MODE_ROTATE_ALT and M_ANGLE_SNAPPING_VALUE.
- New: it is now possible to hold the Shift key to keep the same aspect ratio while resizing graphics.
- New: it is now possible to hold the Ctrl key to interactively resize a graphic with respect to its center.
- New: it is now possible to hold the Alt key to force a square aspect ratio while resizing graphics.
- New: interactive rotation can now be configured using MgraControlList() with M_MODE_RESIZE, M_MODE_RESIZE_SHIFT, M_MODE_RESIZE_CTRL and M_MODE_RESIZE_ALT.
- New: when interactively changing the secondary dimension of a rectangle or ring, it is now possible to hold the Ctrl key to change it on both sides.
- New: interactive resize can now be configured to be symmetric using MgraControlList().
- New: interactive secondary dimension control of a graphic can now be configured using MgraControlList() with the M_MODE_RESIZE_SECONDARY_DIMENSION_SHIFT/CTRL/ALT controls.
- New: graphics can now be translated interactively using the keyboard. See MgraControlList() with M_ACTION_... controls for more customization.
- New: graphics can now be rotated interactively using the keyboard. See MgraControlList() with M_ACTION_... controls for more customization.
- New: graphics can now be resized interactively using the keyboard. See MgraControlList() with M_ACTION_... controls for more customization.
- New: the corners can now be inquired for most graphics in a graphics list using MgraInquireList() with M_CORNER_... control flags.
- New: MgraInteractive() now supports M_ROTATE_AROUND_CORNER flag to create a rectangle with M_POSITION_TYPE set to M_CORNER_AND_DIMENSION.
- New: MgraInteractive() now supports the M_ORIENTED_RECT creation mode to create a rectangle at 0 degree using 3 clicks only.
- New: MgraInteractive() now supports the M_CIRCLE creation mode to create a circle using a click-and-drag.
- New: MgraInteractive() now supports the M_AXIS_ALIGNED_RECT creation mode to create a rectangle at 0 degree using a click-and-drag.
- New: MgraInteractive() now supports the M_AXIS_ALIGNED_ELLIPSE creation mode to create an ellipse at 0 degree using a click-and-drag.
- New: MgraInteractive() now supports the M_ARC_THREE_POINTS creation mode to create a circular arc using 3 clicks only.
- New: M_SAME_LOCATION can now be added to M_POSITION_TYPE to change the reference position of a rectangle in a graphics list without changing its location in the image.
- New: M_RESIZE_WIDTH and M_RESIZE_HEIGHT can now be used to change a graphic's dimensions using MgraControlList().
- New: M_MULTIPLE_SELECTION_KEY can now be used to change the combination key for interactive multiple selection using MgraControlList().
- New: M_EASY_SELECTION can now be enabled using MgraControlList() to allow selecting graphics interactively using the entire shape, not only the shape contour.
- New: M_CENTER_X and M_CENTER_Y can now be inquired to retrieve the center of most graphics in a graphics list using MgraInquireList().
- New: MgraControlList() with M_ACTION_MODIFIER_SPEED can now be used to specify the combination key to change the speed of interactive edition.
- New: MgraControlList() with M_ACTION_KEY_DELETE can now be used to specify the key used to delete the selected graphics.
- New: MgraControlList() with M_ACTION_KEY_CANCEL can now be used to specify the key used to cancel the current interactive graphic edition.
- New: MgraControlList() with M_ACTION_KEYS can now be used to activate graphic interactive edition with the keyboard.
- Improvement: when interactively creating dots, the vertices are now updated until the mouse-up event.
- Improvement: when editing a line interactively, the cursor over the end points is now easier to distinguish from the translation cursor.
- Improvement: the default value for M_SELECTION_RADIUS has been increased to 8 to ease graphics list interactivity.
- Improvement: interactively resizing a graphic with M_CONSTRAIN_ASPECT_RATIO set to 1.0 is more consistent with usual resizing behavior.
- New: M_DRAW_DIRECTION specifies the optional arrows to draw for the graphic primitives.
- New: MgraVectors() and MgraVectorsGrid() to draw vectors in an image or in a graphics list. MgraVectors() uses provided coordinates and lengths while MgraVectorsGrid() uses provided x/y vector lengths on pixel grid.

2.17 General

- New: MobjControl()/MobjInquire() with M_OBJECT_USER_DATA_PTR to associate a user data pointer to a MIL object.
- New: MdigInquire() with M_SOURCE_NUMBER_OF_FRAMES to retrieve the number of frames from the grab source (avi, folder,...)

- Improvement: MIL error MessageBox will now display all message boxes from several threads, if any.
- Improvement: Safetypes now support nullptr in C++: new overloads have been added to support nullptr.
- New: MimRotate() is now supported in MIL Lite for 90 degree rotations.
- New: In a process using both the managed and unmanaged MIL APIs, the unmanaged MappInquire(M_ERROR) can now return M_THROW_EXCEPTION if this mode has been set by the managed code.
- New: MsysInquire(M_DISTRIBUTED_MIL_TYPE) to retrieve how the system has been allocated using Distributed MIL.
- New: MsysInquire(M_DISTRIBUTED_MIL_REMOTE_COMPUTER_NAME) to retrieve the name of the computer used with MsysAlloc() for a Distributed MIL allocation.
- New: The dongle serial numbers can now be retrieved using MappInquire(M_KEY_SERIAL_NUMBER) .
- New: MILweb API extension for publishing MIL object data for access from a browser or another standalone application using the HTML5 WebSocket communication protocol. Refer to the MIL Releases Notes for MILWeb for more information.

3. Deprecated functionalities

3.01 OCR module

- MocrControl()/Inquire() M_STRING_SIZE flag is deprecated and replaced by M_STRING_CHAR_NUMBER.

3.02 Registration module

- MregInquire() M_NUMBER_OF_ELEMENTS flag is deprecated and replaced by M_NUMBER_OF_REGISTRATION_ELEMENTS.

3.03 Code Reader module

- McodeControl()/McodeInquire() M_DISTORTION flag is deprecated and replaced by M_DECODE_ALGORITHM.
- Renaming an M_STRING_FORMAT control value from M_GS1_RAW_DATA to M_RAW_DATA.
- M_NUMBER_OF_SCANS_PER_ROW result flag is deprecated and replaced by M_ROW_NUMBER_OF_SCANS.
- The McodeGetResultSingle()/McodeDraw() macro M_GRADE_RESULT_INDEX(...) is deprecated and replaced by M_SCAN_RESULT_INDEX(...) and M_ROW_RESULT_INDEX(...).
- M_GRADE_RESULT_INDEX(ScanReflectanceProfileIndex) macro (McodeGetResultSingle and McodeDraw) is deprecated and replaced by M_SCAN_RESULT_INDEX(ScanReflectanceProfileIndex) and M_ROW_RESULT_INDEX(RowIndex).

3.04 Blob module

- M_BLOB_IDENTIFICATION control flag is deprecated and replaced by M_BLOB_IDENTIFICATION_MODE.

3.05 Graphics

- M_MOVABLE is deprecated and replaced by M_TRANSLATABLE.
- M_FORCE_ASPECT_RATIO_1 is deprecated and replaced by M_SQUARE_ASPECT_RATIO.
- The interactive M_DEFAULT rectangle creation mode is deprecated and replaced by M_AXIS_ALIGNED_RECT.
- The interactive M_DEFAULT ellipse creation mode is deprecated and replaced by M_AXIS_ALIGNED_ELLIPSE.

4. Fixed bugs

4.01 Blob module

- Fix: MblobCalculate() will no longer miss recalculating several Feret features when M_SAVE_RUNS is set to M_DISABLE, M_RETURN_PARTIAL_RESULTS is set to M_ENABLE, and M_MAX_BLOBS is set to M_ENABLE.
- Fix: MblobGetResult() will no longer potentially run into an infinite loop when retrieving calibrated angle results with a calibration containing invalid regions.
- Fix: MblobCalculate(), MblobGetRuns(), MblobDraw(), MblobFill() and MblobLabel() will no longer potentially crash in rare situation after an MblobCalculate() with M_RETURN_PARTIAL_RESULTS and M_MAX_BLOBS and M_TIMEOUT or M_STOP_CALCULATE is enabled.
- Fix: MblobGetResult() will no longer generate an error for general results such as M_NUMBER when M_RESULT_OUTPUT_UNITS is set to M_WORLD while the calculated image was not calibrated.
- Fix: MblobSelect() will no longer crash when merging blobs while using a sorting key.
- Fix: MblobDraw(M_DRAW_HOLES) will no longer crash when holes are not calculated.
- Fix: Fixed bug : MblobCalculate() will no longer potentially crash when calculating the M_FERET_AT_PRINCIPAL/SECONDARY_AXIS_ANGLE + M_GRAYSCALE without having set to calculate the M_FERET_AT_PRINCIPAL/SECONDARY_AXIS_ANGLE + M_BINARY.
- MblobGetResult() will no longer crash when the M_NUMBER_OF_FERETS is set to M_INFINITE and M_FERETS is set to M_ENABLE while retrieving other Feret results not explicitly set to M_ENABLE.

4.02 Calibration module

- Fix: McalGrid() and McalList() with M_LINEAR_INTERPOLATION mode will no longer fail or result in a crash when using points with very different spacing in X and Y.
- Fix: McalFree() will no longer potentially result in a crash when an allocation and a deallocation are performed simultaneously in other threads.

4.03 SureDotOCR@ module

- Fix: MdmrGetResult(M_STRING_NUMBER) will no longer return -1, but instead will return 0, when no read has been performed.
- Fix: MdmrGetResult(M_STRING_ANGLE) will no longer return an angle value

- outside the range [0, 360[.
- Fix: MdmrControlStringModel() will now work properly in C# and VB.net when using the control type M_ADD_PERMITTED_CHARS_ENTRY with the control value M_CHAR_LIST.
- Fix: MdmrGetResult(M_FORMATTED_STRING) will no longer return the wrong result in an ASCII environment.

4.04 Pattern Matching module

- Fix: MpatFind() will no longer potentially crash when the source image is not M_READ_PREFETCH_ENABLED and the pattern matching model has a SizeX less than 6 pixels.
- Fix: MpatFind() will no longer generate a MIL Error when the models' search regions are different and an ROI is associated to the target image. As documented, the ROI takes precedence.
- Fix: M_PAT_MODEL objects are now tagged as internally allocated objects.

4.05 Model Finder module

- Fix: MmodDraw() will no longer draw invalid annotations when a calibration context used by MmodFind() has been freed.
- Fix: Drawing CAD models will no longer potentially break some edges.
- Fix: Circle shape finder will no longer return unexpected circle results when M_FIT_ERROR_WEIGHTING_FACTOR is set to 0.
- Fix: Circle finder will no longer potentially crash, with MP enabled, under rare circumstances.
- Fix: MmodGetResult(M_ANGLE) will no longer return the angle value in World units when the M_RESULT_OUTPUT_UNITS is set to M_PIXEL for a Geometric or a Geometric Controlled context type.
- Fix: MmodFind() with a M_SHAPE_CIRCLE or a M_SHAPE_ELLIPSE context type will now return the best occurrences when, in some rare situations, the M_NUMBER control was set to a fix value and that at least the score of one occurrence was higher than the certainty value.
- Fix: MmodDraw(M_DRAW_POSITION) with an M_SHAPE_CIRCLE or M_SHAPE_ELLIPSE context type will no longer draw the wrong direction when used with a calibration and the specified M_RESULT_OUTPUT_UNITS is set to M_PIXEL.
- Fix: MmodDefineFromFile() will no longer fail to open a CAD DXF file in some rare situation.

4.06 Metrology module

- Fix: The missing MIL.NET overloads for MmetPut() have been added.
- Fix: The region will no longer be displayed when its local frame cannot be calculated.
- Fix: Tolerance M_VALUE_WARNING_MIN/MAX default values are now equal to M_EQUAL_PASS_MIN/MAX to fix a misleading default behavior.
- Fix: Constructing a cloned edgel feature will no longer crash for a very specific derived region while using multi-core acceleration.
- Fix: the intersection between two circles will no longer potentially lead to the wrong constructed points.
- Fix: the extended intersection between two arcs, or between an arc and a circle, will no longer potentially lead to the wrong constructed points.
- Fix: deleting a feature will no longer potentially crash when the feature has dependencies.
- Fix: the perpendicularity tolerance will no longer lead to potentially wrong tolerance status when using different minimum and maximum tolerance limits while the verified feature is not an edge feature.
- Fix: the parallelism tolerance will no longer lead to potentially wrong tolerance status when using different minimum and maximum tolerance limits while the verified feature is not an edge feature.
- Fix: the position tolerance of a single feature relative to the default global frame will no longer be erroneous.
- Fix: the bisector construction will no longer potentially return the wrong result regarding the order of the line features.
- Fix: feature / tolerance label upper limit is now set to 0x01000000L to prevent unexpected behaviors.
- Fix: a circle outer fit could potentially lead to the wrong solution. This is now fixed.
- Fix: the start angle of a constructed arc could potentially be wrongly set to 360 degrees. This will no longer be allowed.
- Fix: in some rare conditions, associating a reference frame to a feature using its label value was not possible. This is no longer the case.
- Fix: the angle constraint of a ring sector will no longer be potentially ignored when calculating a measured feature.
- Fix: the region constraints will no longer be ignored when using M_COPY_FEATURE_EDGELS with a constructed edgel and when setting the M_EDGEL_RELATIVE_ANGLE to M_SAME_OR_REVERSE or when setting the M_EDGEL_ANGLE_RANGE to 360 degrees.
- Fix: Restoring a context will no longer ignore the angularity tolerance M_ANGLE value, if any.
- Fix: the maximum distance between a circle and an arc will no longer be potentially erroneous.
- Fix: calculating the bisector between two identical lines will no longer lead to an invalid construction.
- Fix: Global result status will no longer return pass if there is at least one warning tolerance status. It will now return a fail status when there is at least a fail status, or a warning status if there is at least a warning status and no fail.
- Fix: The ring sector region can now be used to select edgels for the construction of a segment.

4.07 3dMap module

- Fix: M3dmapImport() will no longer fail loading M_PLY files with empty 'face' sections.
- Fix: M3dmapImport() will no longer crash when reading an M_STL file having no triangles.

- Fix: M3dmapImport() will no longer crash when using an existing point cloud label.
- Fix: M3dmapStat() with M_DEVIATION_MAX will no longer potentially returns the wrong result for very large depth map image.

4.08 Code Reader module

- Fix: McodeWrite() will no longer return an error when rendering a model, and another model of the context has an inconsistent input unit.
- Fix: McodeRead() will no longer perform a short read with M_PHARMACODE codes when the M_SEARCH_ANGLE_STEP is enabled.
- Fix: McodeRead() will no longer fail to read or return a wrong string when reading an EAN/UPC code specific type of degradation.
- Fix: McodeInquire() will no longer return M_DEFAULT when the inquire type is M_DOT_SPACING_MIN + M_DEFAULT or M_DOT_SPACING_MAX + M_DEFAULT.
- Fix: McodeRead() will no longer fail decoding 2D code types with a very small cell size (as small as 1 pixel) when setting proper M_CELL_SIZE_MIN/MAX values and setting M_USE_PRESEARCH to M_STAT_BASE.
- Fix: McodeRead() will no longer fail reading 1d codes in low contrast synthetic images when M_THRESHOLD_MODE is set to M_ADAPTIVE.
- Fix: McodeGetResult() and McodeGetResultSingle() will no longer return wrong values for M_NUMBER_OF_ERRORS and M_NUMBER_OF_ERASURES after reading or grading an M_DATAMATRIX in rare situations while multi-core acceleration is used.
- Fix: McodeGetResult() and McodeGetResultSingle() will no longer return M_STATUS_READ_OK or M_STATUS_GRADE_OK when M_NUMBER and M_TOTAL_NUMBER are set to M_ALL and no code is found.
- Fix: McodeRead() will no longer potentially erroneously report an error when decoding a MaxiCode with M_FOREGROUND_VALUE set to M_FOREGROUND_ANY.
- Fix: McodeRead() and McodeGrade() will no longer ignore the control value M_TOTAL_NUMBER for M_DATAMATRIX codes.
- Fix: McodeGrade() will no longer fail grading 2D matrix codes when the dot spacing is not null and M_FOREGROUND_VALUE is set to M_FOREGROUND_ANY.
- Fix: McodeRead() will no longer fail reading a Micro QR code version M1 when M_ERROR_CORRECTION is set to M_ECC_L.
- Fix: a code 128 that does not respect the EAN14 constraints will no longer be read as an EAN14.
- Fix: McodeWrite() will no longer ignore the value of M_CELL_NUMBER_Y when writing a DataMatrix code.
- Fix: McodeRead() and McodeGrade() will no longer return an error when the value of M_CELL_NUMBER_X/Y is set to a fixed valid value and M_CELL_NUMBER_Y/X is set to M_DEFAULT or to M_ANY.
- Fix: McodeRead() and McodeGrade() will now report an error when an invalid value for M_CELL_NUMBER_X or M_CELL_NUMBER_Y is used while reading an M_QRCODE, M_MICROQRCODE or M_AZTEC code.
- Fix: McodeRead() will no longer crash in some rare situations while reading a linear barcode with M_SPEED set to M_VERY_LOW.
- Fix: McodeDraw() will no longer report an error when operations are accumulated to render multiple features.
- Fix: McodeRead() will no longer fail to read M_GS1_DATABAR stacked code when the user provides an inaccurate angle.
- Fix: McodeRead() will no longer fail to decode or return a wrong string for UPC/EAN codes in some rare situation.
- Fix: McodeRead() will no more potentially fail to read 2D matrix codes in specific situations when the M_MIN_CELL_SIZE is set to a value less than 3.
- Fix: McodeGrade() will no longer potentially crash when reading Code 128 codes, in specific rare situations.
- Fix: McodeInquire() with M_ENCODING will no longer return a wrong value for M_MICROQRCODE.
- Fix: McodeRead() will no longer cause a heap corruption in very rare situation while decoding an ANSI X12 encoded DataMatrix.
- Fix: McodeWrite() will no longer potentially encode the wrong string for M_DATAMATRIX with M_ECC_200.
- Fix: McodeRestore()/McodeStream() will no longer crash when restoring a context containing an unknown code type.
- Fix: McodeRead() will no longer potentially crash when reading 4-state codes in images containing no 4-state codes.
- Fix: McodeRead() will no longer potentially fail reading a DataMatrix when M_DATAMATRIX_SHAPE is set to M_SQUARE.
- Fix: McodeRead() will no longer crash when reading a C or N Australian encoded 4-state code while the first decoded character is not valid.
- Fix: McodeRead() will no longer potentially return an invalid string when reading linear codes with M_STRING_SIZE_MAX sets to a non-default value and M_ERROR_CORRECTION sets to M_ECC_NONE.
- Fix: McodeGetResult() and McodeGetResultSingle() will no longer report an error when the result type is M_NUMBER_OF_DATA_CODEWORDS + M_2D_COMPONENT or M_DATA_CODEWORDS + M_2D_COMPONENT.
- Fix: McodeRead() will no longer fail to read 1D codes in some situation while M_FOREGROUND_VALUE is set to M_FOREGROUND_ANY.
- Fix: McodeFree() will no longer potentially crash when an allocation and a deallocation are performed simultaneously in different threads.
- Fix: McodeRead() will now decode entirely the EAN/UPC code, including its add-on, in every situation when M_ENCODING is set to M_ANY.
- Fix: McodeRead() will no longer decode the wrong string for 1D codes in a very rare situation when M_SEARCH_ANGLE_STEP is set to a specific value.
- Fix: McodeRead() will no longer return slightly different results whether the multi-processing is enabled or disabled, in rare situation for a M_DATAMATRIX code (did not affect the decoded string).
- Fix: McodeGrade(), in some situations for M_DATAMATRIX codes, will no longer return wrong grade values for the M_EXTENDED_AREA... grading result types when the M_EXTENDED_AREA_REFLECTANCE_CHECK is set to M_ENABLE.
- Fix: McodeRead() will no longer fail to read a M_QRCODE or a M_MICROQRCODE when there in presence of a perspective distortion.
- Fix: McodeGrade(), for a M_QRCODE or a M_MICROQRCODE, will no longer

- return a wrong M_GRID_NONUNIFORMITY and M_GRID_NONUNIFORMITY_GRADE in some cases with perspective distortion.
- Fix: In rare situations, McodeRead() will no longer detect two occurrences on a same barcode when reading a linear code.
- Fix: McodeRead() will no longer return a wrong angle result when grading a GS1 Databar code type.
- Fix: For 1D barcode, the returned width will now be more consistent with the true size of the barcode when the search angle mode is enabled.
- Fix: In rare situation, McodeGrade() will no longer return an invalid sub-grade of a fixed pattern damage, sub-grade of version and format information, invalid number of codewords, number of data codewords, number of error correction codewords, number of error-corrected and number of erasure when M_GRADING_STANDARD is set to M_AIMDPM_GRADING.
- Fix: The CodeReader module will no longer crash when retrieving the M_STRING result (McodeGetResult / McodeGetResultSingle) in DMIL configuration when no code was previously read, graded, trained or detected.

4.09 Measurement module

- Fix: MmeasGetResult() will no longer ignore SecondResultArrayPtr when retrieving M_DRAW_PROFILE_SCALE_OFFSET.
- Fix: MmeasGetResult() will no longer crash for a M_BOX_EDGEVALUES result type with the SecondResultArrayPtr set to M_NULL.

4.10 Edge Finder module

- Fix: MedgeGetResult() of the information of a buffer not available in the result object will no longer lead to a crash.

4.11 Registration module

- Fix: MregCalculate() will no longer take a long time but fail quickly when a constant color image is provided to a M_STITCHING context type.

4.12 String Reader module

- Fix: MstrSetConstraint() will now validate M_FONT_INDEX(...) properly when using a font-less context.
- Fix: MstrInquire() will no longer crash when inquiring the M_CONSTRAINT at a specific index of a remote context allocated using DMIL.

4.13 Primitives

- Fix: MimResize() with M_MAX or M_MIN will no longer fetch data outside the source image buffer.
- Fix: MimArith() with M_ATAN2 will no longer ignore regions.
- Fix: MimRestore()/MimStream(), applied to an M_MATCH_CONTEXT will no longer crash when the saved model was not initialized.
- Fix: MimMorphic() with M_MATCH will no longer potentially return the wrong result for 8xY structural elements on a 32-bit OS.
- Fix: MimGetResult()/MimGetResultId() will no longer potentially crash when retrieving M_POSITION_X, M_POSITION_Y, or M_VALUE on an uninitialized M_EVENT_LIST.
- Fix: MbufGetLine() will no longer return the wrong result or an error when the line is completely outside of the buffer limits.
- Fix: MbufGetLine() will no longer potentially crash when the line is horizontal and completely outside of the buffer limits.
- Fix: MbufGetArc() will no longer potentially miss a portion of the arc when the angle range was 360.
- Fix: MimResize() will no longer potentially crash when using very large scale values on a 32-bit OS.
- Fix: MimLocatePeakId() will no longer potentially crash in rare situations and after a specific sequence of calls

4.14 Graphics

- Fix: MgraArcFill() will no longer potentially write a full row or column when the radius is less than 0.5.
- Fix: MbufSetRegion() from a graphics list containing high-level module drawings (MxxxDraw) will no longer crash when a 'rasterizing' option is used.
- Fix: the angle of a line graphic will no longer be undefined when its end points are at the same position.
- Fix: rotated rectangles in a graphic list will no longer be incorrectly rendered when the display has different X/Y zoom factors.
- Fix: moving a vertex of a graphic using MgraControlList() with M_POSITION_X/Y or M_TRANSLATE_X/Y will no longer miss updating the global position of the graphic.
- Fix: moving a line end point interactively will no longer forget updating the line position and angle.
- Fix: interactively translating a polygon or polyline of size 0 will no longer change a vertex position instead.
- Fix: interactively translating a circle or arc of size 0 will no longer change its start or end angle instead.
- Fix: interactively resizing a polygon or polyline whose bounding box has a null width or height will no longer inconsistently resize the graphic.
- Fix: a rectangle with its top left coordinate at (-0.5, -0.5) will no longer be rendered outside the buffer.
- Fix: MmeasDraw() with M_DRAW_EDGES will now generate a graphic that can be selected both in pixel units and world units.
- Fix: MgraInteractive() will no longer lead to unexpected behaviors when mouse events occur outside the MIL display.
- Fix: MdispControl() with M_REGION_OUTSIDE_SHOW or M_REGION_INSIDE_SHOW will no longer incorrectly display an inherited region of a child buffer.
- Fix: M_POSITION_X and M_POSITION_Y will no longer return invalid values when interactively defining a zero-sized line using MgraInteractive().
- Fix: interactively editing the start angle or the end angle of an arc will no longer generate empty arcs.

4.15 Utilities

- Fix: Matrox Profiler will no longer potentially crash when the Windows operating system decimal symbol preference is set to "," instead of ".".
- Fix: Matrox Profiler will no longer crash when moving a thread while running under .NET Framework 4.7 runtime.
- Fix: Matrox Profiler will no longer potentially crash when closing it while "Export to Text File" is under execution.

5. Known limitations

5.01 Classification module

- MclassPredict() called with image other than 8+M_UNSIGNED won't support regions (MbufSetRegion) properly.

5.02 Model Finder module

- When a circle is found using a M_SHAPE_ELLIPSE context and a search region is specified, the angle of the returned circle occurrence can be outside the specified angle search range.

Section 2: Differences between MIL 10 Processing Pack 2 with Update 63 and MIL 10 Processing Pack 2

Table of Contents for Section 2

1. Overview
 2. New functionalities and improvements
 - 2.1 SureDotOCR®
 3. Quick reference
 - 3.1 SureDotOCR®: how to set a string angle
 - 3.2 SureDotOCR®: how to set a string italic angle
 - 3.3 SureDotOCR®: how to delete a char entry
-

1. Overview

- MIL 10 Update 63 includes new functionalities, performance optimizations and general improvements for the MIL SureDotOCR® module.
- MIL 10 Update 63 is an update to MIL 10 Processing Pack 2 only.

2. New functionalities and improvements

2.1 SureDotOCR®

- Improvement: the read operation robustness and accuracy has been improved.
- Improvement: the read operation now provides better character localization stability.
- Improvement: the read operation speed has been greatly improved and specifying the string angle and orientation can provide additional speed improvement.
- New: the string italic angle can now be specified using M_ITALIC_ANGLE_MODE and M_ITALIC_ANGLE control flags.
- New: the string angle or orientation can now be specified using M_STRING_ANGLE_MODE, M_STRING_ANGLE and M_STRING_ANGLE_INPUT_UNITS control flags.
- New : MdmrControlStringModel() with M_DELETE_PERMITTED_CHARS_ENTRY is now supported to delete an entry at a specific index.
- New: MdmrDraw() now supports M_DRAW_MIL_FONT_FORMATTED_STRING.

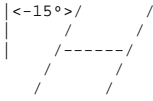
3. Quick reference

3.1 SureDotOCR®: how to set a string angle

- MdmrControl() with M_STRING_ANGLE_MODE control flag can be set to:
 - . M_AUTO (M_DEFAULT): the string angle is automatically determined.
 - . M_ANGLE: a fixed angle specified by the M_STRING_ANGLE control is used to read a string from left to right with the characters facing upward.
 - . M_ORIENTATION: a fixed angle specified by the M_STRING_ANGLE control is used to read a string from left to right with the characters facing upward or from right to left with the characters facing downward.
- MdmrControl() with M_STRING_ANGLE control flag can be set to a value from 0 to +360 degrees, or to M_DEFAULT (0 degrees) interpreted according to the M_STRING_ANGLE_INPUT_UNITS control. The M_STRING_ANGLE control flag can also be set to M_ACCORDING_TO_REGION to use the angle provided by a rectangular region associated to the target image.
- MdmrControl() with M_STRING_ANGLE_INPUT_UNITS control flag can be set to:
 - . M_PIXEL (M_DEFAULT) to use the pixel coordinate system.
 - . M_WORLD to use the coordinate system of a calibrated target image. This control is ignored when M_STRING_ANGLE is set to M_ACCORDING_TO_REGION.

3.2 SureDotOCR®: how to set a string italic angle

- MdmrControl() with M_ITALIC_ANGLE_MODE control flag can be set to:
 - . M_AUTO (M_DEFAULT) : the italic angle is automatically determined.
 - . M_ANGLE : a fixed italic angle specified by the M_ITALIC_ANGLE control is used.
 Note that the M_ITALIC_ANGLE_MODE must be set to M_AUTO (M_DEFAULT) if the M_STRING_ANGLE_MODE is set to M_AUTO (M_DEFAULT).
- MdmrControl() with M_ITALIC_ANGLE control flag can be set to a value between -90 degrees and +90 degrees, or to M_DEFAULT (0 degrees). The italic angle value must be provided counter clockwise.
Example:



3.3 SureDotOCR@: how to delete a char entry

- MdmrControlStringModel() EntryIndex with M_DELETE_PERMITTED_CHARS_ENTRY can be set to:
 - . a value between 0 and M_NUMBER_OF_PERMITTED_CHARS_ENTRIES-1 to delete a specific entry.
 - . M_ALL (M_DEFAULT) to delete all entries.

Example :

```
MdmrControlStringModel(Ctx, M_STRING_LABEL(1), M_POSITION_IN_STRING(2), M_DELETE_PERMITTED_CHARS_ENTRY, 3, M_DEFAULT, M_NULL);
```

Section 3: Differences between MIL 10 Processing Pack 2 and MIL 10 Processing Pack 1

!!! ATTENTION MATROX DESIGN ASSISTANT 4 USERS !!!
The installation of MIL 10, MIL 10 Processing Pack 1, MIL 10 Processing Pack 2 and Matrox Design Assistant 4 may lead to a problematic configuration. The installation of Matrox Design Assistant 4 Service Pack 1 will be required to rectify the configuration issue.

Table of Contents for Section 3

1. Overview
2. New functionalities and improvements
 - 2.01 MIL processing specific examples
 - 2.02 Blob module
 - 2.03 Calibration module
 - 2.04 Code Reader module
 - 2.05 Edge Finder module
 - 2.06 Model Finder module
 - 2.07 Registration module
 - 2.08 Metrology module
 - 2.09 Measurement module
 - 2.10 Pattern Matching module
 - 2.11 3dMap module
 - 2.12 String Reader module
 - 2.13 Ocr module
 - 2.14 Dot Matrix module (SureDotOCR@)
 - 2.15 Primitives
 - 2.16 General
 - 2.17 Utilities
 - 2.18 Distributed MIL
 - 2.19 MIL Help
3. Deprecated functionalities
 - 3.01 Calibration module
 - 3.02 Code Reader module
 - 3.03 Pattern Matching module
 - 3.04 Edge Finder module
 - 3.05 Blob module
 - 3.06 Primitives
 - 3.07 General
4. Fixed bugs
 - 4.01 Blob module
 - 4.02 Calibration module
 - 4.03 Code Reader module
 - 4.04 Edge Finder module
 - 4.05 Model Finder module
 - 4.06 Metrology module
 - 4.07 Measurement module
 - 4.08 Pattern Matching module
 - 4.09 Ocr module
 - 4.10 3dMap module
 - 4.11 Color module
 - 4.12 Primitives
 - 4.13 Interactive utilities
 - 4.14 General
5. Known limitations and bugs
 - 5.01 Interactive utilities
 - 5.02 MIL add-on to Microsoft Visual Studio
 - 5.03 MIL Help
 - 5.04 Pattern Matching
 - 5.05 General

1. Overview

- MIL 10 Processing Pack 2 includes all the features of MIL 10 Processing Pack 1.
In addition, MIL 10 Processing Pack 2 includes new processing functionalities, performance optimizations, and general improvements such as:
 - . new: Calibration support for partial grids,
 - . new: Calibration support for chessboard grids with fiducial,
 - . new: Model Finder dedicated ellipse shape matcher,
 - . new: Registration support for depth from focus,
 - . new: 3D alignment extended to unorganized clouds of points,

- . new: Dot Matrix reader module,
- . new: Code Reader default behaviour modes,
- . new: Primitives such as the integral image and texture statistics,
- . and many more to discover!

- MIL 10 is the minimum requirement for all upcoming Processing Packs until the next major release.

2. New functionalities and improvements

2.01 MIL processing specific examples

- New MIL 10 Processing Pack 2 examples!
- Note that the MIL 10 Update 37 is needed to get the images required for several examples, in particular for 3D processing examples.

2.02 Blob module

- New: the blob module now has a new Context based API! See section 3.05 for more details about the new API.
- New: `M_BLOB_INDEX` can now be used with `MblobGetResult()` to refer to blobs by index instead of label.
- New: `M_INDEX_VALUE` can now be retrieved using `MblobGetResult()` to obtain the index corresponding to a blob label.
- New: `M_INDEX_VALUE` can now be used as a criterion with `MblobSelect()`.
- New: `M_AVAILABLE` can now be used with `MblobGetResult()` to inquire whether a feature was calculated.
- New: `MblobGetResult()` with `M_TOTAL_NUMBER_OF_CHAINED_PIXELS` to retrieve the total number of chained pixels.
- New: `MblobGetResult()` with `M_TOTAL_NUMBER_OF_CONVEX_HULL_POINTS` to retrieve the total number of convex hull points.
- New: `MblobGetResult()` with `M_TOTAL_NUMBER_OF_FERETS` to retrieve the total number of ferets.
- New: `MblobGetResult()` with `M_NUMBER_OF_FERETS` to retrieve the number of ferets.
- New: `MblobGetResult()` with `M_CALCULATION_TYPE` to retrieve whether `MblobCalculate()` was done, and with which type of sources (binary only, or binary and grayscale).
- New: `M_BLOB_INDEX` can be used with `MblobDraw()` to refer to blobs by index instead of label.
- New: with the new Context based API, it is now possible to save, load and restore a blob context using `MblobSave()`, `MblobRestore()` and `MblobStream()`.
- Improvement: `MblobSelect()` with `M_LABEL_VALUE` has been accelerated.
- Improvement: `MblobCalculate()` and `MblobGetResult()` with sorting keys associated to enabled features have been accelerated.

2.03 Calibration module

- New: `M_GRID_PARTIAL` can now be set to `M_ENABLE` to find calibration grids that partially fall outside the image.
- New: `M_GRID_SHAPE` can be set to `M_ANY` when searching for partial non-rectangular grids.
- New: `M_GRID_HINT_ANGLE_X` can now be used to specify the grid orientation when searching for partial grids.
- New: `McalGrid()` `RowNumber` and `ColumnNumber` parameters can be set to `M_UNKNOWN` when searching for partial grids.
- New: `McalDraw()` now supports `M_DRAW_FIDUCIAL_BOX` to draw a box around the grid's fiducial.
- `McalGrid()` now supports chessboard grids with a Datamatrix fiducial. The Datamatrix fiducial encodes information about the grid such as the grid reference point, the orientation as well as the spacing. See `McalControl()` with `M_GRID_FIDUCIAL`.
- New: `McalFixture(M_LEARN_OFFSET, M_MODEL_PAT)` now supports the new Pattern Matching API.
- New: `McalInquire()` `M_GRID_UNITS` and `M_GRID_UNIT_SHORT_NAME` can now be inquired to retrieve the units encoded in the fiducial, after calling `McalGrid()` using a chessboard with fiducial.
- New: a chessboard grid sample with fiducial is now provide in the Matrox Imaging\images directory (`ChessboardCalibrationGrid_Fiducial_Letter.pdf`).
- Improvement: the accuracy of the `M_LINEAR_INTERPOLATION` mode has been improved when the scale difference between the X and the Y coordinates is large.
- Improvement: `McalGrid()` is now more robust when using chessboard cells which contains minor defects.
- Improvement: `McalGrid()` has been accelerated for chessboard grids.
- Improvement: `McalGrid()` is now more robust to very dark or very bright backgrounds.

2.04 Code Reader module

- New: the Code Reader tool now supports the `M_INITIALIZATION_MODE` to change the default behaviour of the module `alloc-time` or `control-time`. The default behaviour can be set to a typical reading rate, or to an improved reading rate. The latter considers by default more degrees of freedom: any polarity, presence of distortion and non uniformity, angular position, flipped geometry, etc...
- New: `McodeControl()` `M_STRING_FORMAT` can now be combined with `M_UTF8` to retrieve a string using UTF-8 encoding.
- New: The `M_FOREGROUND_VALUE` can now be set to `M_FOREGROUND_ANY` for `M_DATAMATRIX`, `M_MAXICODE`, `M_PDF417`, `M_TRUNCATED_PDF417`, `M_GSI_DATABAR`, `M_COMPOSITECODE`, `M_PLANET`, `M_POSTNET`, `M_4_STATE`, and `M_PHARMACODE`.
- New: `McodeGrade()` now supports ISO 15415 and AIM-DPM grading for Aztec Rune.
- New: `McodeGrade()` now supports grading an Aztec code from a read result.
- New: Support the Aztec code detailed segment A and B results for the Fixed Pattern Damage.
 - `M_FIXED_PATTERN_DAMAGE_A_GRADE` to retrieve grade from segment A.
 - `M_FIXED_PATTERN_DAMAGE_B_GRADE` to retrieve grade from all segments B.

- M_NUMBER_OF_FIXED_PATTERN_DAMAGE_B_SEGMENT to retrieve the number of segments B.
- M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_A_GRADE to retrieve grade from segment A in the extended area.
- M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_B_GRADE to retrieve grade from all segments B in the extended area.
- Improvement: McodeRead() robustness to blurred images and to barcodes with poor modulation has been improved.
- Improvement: McodeRead(), to read Datamatrix using the
- Improvement: McodeRead(), to read Datamatrix using the M_PERSPECTIVE_UNEVEN_GRID_STEP mode or the M_UNEVEN_GRID_STEP mode, is now more robust to aliasing effects.
- Improvement: McodeRead() DataMatrix detection in noisy images has been improved when M_DISTORTION is set to M_PERSPECTIVE_UNEVEN_GRID_STEP or M_UNEVEN_GRID_STEP.
- Improvement: McodeRead() DataMatrix detection in noisy images has been improved when M_DISTORTION is set to M_PERSPECTIVE_UNEVEN_GRID_STEP and M_THRESHOLD_MODE is set to M_ADAPTIVE.
- Improvement: Support read/write of empty string with 2D codes (except mode 2 and 3 Maxicodes and Aztec Rune) and cross-row codes. Datamatrix, QrCode, MicroQrCode, MaxiCode, Aztec, MicroPDF417, truncated PDF417, PDF417 accept 0 as a valid value for M_STRING_SIZE_MIN and M_STRING_SIZE_MAX.
- Improvement: McodeRead() and McodeWrite() now supports the application identifiers 7005, 7006, 7007, 7008, 7009 and 7010 as specified in version 15 of the GS1 General Specifications document.
- Improvement: McodeRead() robustness has been improved for M_MICROPDF417 when the M_USE_PRESEARCH is set to M_STAT_BASE.
- Improvement: McodeRead() will now use the angle range to determine the string to return for M_PLANET and M_POSTNET codes in the rare cases it can be decoded in both directions.
- McodeGetResult() now accepts + M_STRING_SIZE as combination value to M_STRING, and M_TEXT to retrieve the size of the corresponding string including the terminating null character ("\0").

2.05 Edge Finder module

- New: M_CHAIN_APPROXIMATION can now be set to M_WORLD_LINE to perform the line approximation in world units.

2.06 Model Finder module

- A new specialized M_SHAPE_ELLIPSE finder is now supported.
- New: MmodFind() can now be used with an Edge result target when using M_SHAPE_CIRCLE context.
- New: MmodInquire() M_HAS_DONT_CARE_MASK, M_HAS_FLAT_REGIONS_MASK, M_HAS_WEIGHT_REGIONS_MASK inquires to verify if one of the mask has been defined.
- New: MmodControl() M_COVERAGE_MAX can now be used to set the expected maximum model coverage for shape (circles and ellipse) finder occurrences.
- New: MmodControl() M_FIT_SCORE_MIN can now be used to set the expected minimum fit score for shape (circle and ellipse) finder occurrences.
- New: MmodGetResult() M_SCORE_FIT can now be retrieved from a shape finder (circle and ellipse) result.
- Improvement: MmodFind() in calibrated targets has been improved for M_SHAPE_CIRCLE context.
- Improvement: the restoration of a context from a file has been accelerated.
- Improvement: MmodFind() has been improved when matching a very large number of occurrences with an M_GEOMETRIC context.
- Improvement: searching a synthetic circle model with an M_GEOMETRIC context has been improved for robustness.
- Improvement: MmodFind() with an M_SHAPE_CIRCLE context has been improved when facing edges shaped in spiral.
- Improvement: greediness of MmodFind() with an M_SHAPE_CIRCLE context has been dramatically decreased, thus improving the robustness.
- Improvement: model size limitation has been increased up to 4K x 4K.

2.07 Registration module

- New: MregCalculate can now process a stack of images to compute the corresponding depth from focus index map.

2.08 Metrology module

- New: MmetCalculate() now supports M_NULL as a source image to calculate contexts which only contains constructed features.
- New: MmetPut() to set external edgels to an M_EXTERNAL_FEATURE constructed edgel feature.
- New: MmetControl() with M_ACTIVE to enable/disable a feature for the calculation. The feature and all its dependencies will fail accordingly.
- New: A point feature can now be constructed from a measured point feature. For example, the construction of a point from the second point of a measured point feature along a segment region.
- New: A local frame can now be constructed from a segment. The local frame is positioned at the centre of the segment and is oriented towards the segment end point.
- New: M_FILL_GAP_ANGLE extraction control flag can now be set for measured features using MmetControl().
- New: M_FILL_GAP_DISTANCE extraction control flag can now be set for measured features using MmetControl().
- New: M_CLONE_FEATURE constructed edgel features now support regions. Only the edgels included in the region are cloned.
- Improvement: edgels are now drawn as crosses when zoomed for better visual experience.

2.09 Measurement module

- Improvement: the MmeasSetMarker(M_DRAW_PROFILE_SCALE_OFFSET) default behaviour is now set to M_AUTO_SCALE_PROFILE and M_AUTO_OFFSET_PROFILE.

2.10 Pattern Matching module

- New: the pattern matching module now has a new Context based API!
See section 3.03 for more details about the new API.
- New: MpatGetResult(M_ANGLE) will no longer be restricted to rotated models, but is available for all model types.
- New: MpatGetResult(M_INDEX) (formerly M_MODEL_INDEX) will no longer be restricted to M_FIND_BEST_MODELS results, but is available for all results.
- New: In MpatGetResult(), it is now possible to add + M_AVAILABLE to a ResultType to determine whether a result is available.
- New: MpatControl(M_NUMBER) with the new API, can now be set to 0 for a specific model in a Context.
- New: MpatStream() with M_MEMORY is supported on remote systems. This was not the case for the model based deprecated API.
- Improvement: MpatGetResult() can now be cast to a specific data type using + M_TYPE_MIL... combination flags.
- Improvement: MpatFind() with multiple models verifies that all models have the same size and use the same search region in the target image. If not, an error is generated.

2.11 3dMap module

- New: M3dmapAlign() no longer has the constraint that the scene point cloud must be of M_LASER_SCAN type. It now supports M_UNORGANIZED scene point cloud.
- New: M3dmapAlign now supports the extraction boxes using the following new control flags M_EXTRACTION_BOX_MODEL and M_EXTRACTION_BOX_SCENE set to M_USE.
- new: M3dmapAlign() result can now return an M_NO_VALID_POINTS status.
- New: M3dmapControl() M_ERROR_MINIMIZATION_METRIC control flag can be set to M_POINT_TO_POINT, M_POINT_TO_PLANE+M_USE_MODEL_NORMALS or M_POINT_TO_PLANE+M_USE_SCENE_NORMALS for an M_PAIRWISE_ALIGNMENT_CONTEXT.
- New: M3dmapImport() now supports M_STL (.stl) 3d CAD file format.
- New: M3dmapGetResult() has new combination constant M_INSIDE_EXTRACTION_BOX when retrieving the M_NUMBER_OF_3D_POINTS.
- New: M3dmapGet() new M_INSIDE_EXTRACTION_BOX combination flag specifies to only return the points that are inside the extraction box defined using M3dmapSetBox().
- New: M3dmapGet() new M_PARTIAL_WRITE combination flag specifies to output into the destination arrays even if they are not large enough to hold all the elements.
- New: M3dmapArith() with M_DIST_NN to compute the euclidean distance between two depth maps.
- New: M3dmapArith() with M_DIST_NN_SIGNED to compute the signed euclidean distance between two depth maps.
- New: M3dmapArith() with M_VALIDITY_MAP to compute the validity map of the pixels from two depth maps.
- New: M3dmapInquire() with M_POINT_CLOUD_INDEX_VALUE and M_POINT_CLOUD_LABEL_VALUE now return M_INVALID when using an invalid point cloud label or index.
- Improvement: M3dmapArith() has been accelerated for multi-core architecture (MP).

2.12 String Reader module

- MstrGetResult() now accepts + M_STRING_SIZE as combination value to M_STRING, M_TEXT, M_FORMATTED_STRING to retrieve the size of the corresponding string including the terminating null character ("\0").
- MstrInquire() now accepts + M_STRING_SIZE as combination value to M_CONSTRAINT and M_DEFAULT_CONSTRAINT to retrieve the size of the corresponding string including the terminating null character ("\0").

2.13 Ocr module

- MocrGetResult() now accepts + M_STRING_SIZE as combination value to M_STRING to retrieve the size of the corresponding string including the terminating null character ("\0").
- MocrInquire() now accepts + M_STRING_SIZE as combination value to M_CONSTRAINT to retrieve the size of the corresponding string including the terminating null character ("\0").

2.14 Dot Matrix module (SureDotOCR®)

- New: A Dot Matrix Reader module to read a dot matrix string of characters such as expiry dates and lot numbers. The module supports font definitions, string model definitions, and string constraints.

2.15 Primitives

- New: MimArith() M_INTEGRAL operation to calculate the integral image of a source buffer.
- New: MbufInquire() with M_NUMBER_OF_ELEMENT_VALUES_VALID to retrieve the number of non M_DONT_CARE elements in a structuring element.
- New: MimConvolve() predefined M_SOBEL_X, M_SOBEL_Y, M_PREWITT_X and M_PREWITT_Y kernels are now supported.
- New: MimProjection() to obtain the sum, minimum, maximum or median projections of a source buffer.
- New: MimProjection() now accepts an image as destination.
- New: MimStatCalculate() to calculate image statistics, cumulative statistics, windowed statistics, and co-occurrence based statistics.
- New: MimStatCalculate() can now returns an M_ORIENTATION_DATA_COHERENCE value associated to the M_ORIENTATION_DATA_MEAN statistics.
- New: MimDraw() with M_DRAW_IMAGE_ORIENTATION now supports a graphic list as a destination.
- New: MimPolarTransform() can now generate the X/Y LUTs that can be used with MimWarp(M_WARP_LUT).
- New: MimTranslate() can now be used to generate the equivalent warp matrix.
- New: MimResize() can now be used to generate the equivalent warp matrix.
- New: MimRotate() can now be used to generate the equivalent warp matrix.
- New: MbufGetLine() can now be used to obtain the coordinates of the pixels.

- New: MimWarpList() to warp X, Y arrays of sources positions to X, Y arrays of destination positions.
- New: MimConvert() can now be used to convert an sRGB image to/from the CIELab colour space.
- New: MimConvert() can now be used to convert an sRGB image to/from the CIElch colour space.
- New: McolDistance() now supports the CIE Delta E distance to measure the distance between colours in the perceptually uniform Lab colour space.
- New: MimBinarize() auto-threshold modes now supports regions.
- New: MbufChildColor now supports M_ALL_BAND to easily create a full sized child.
- New: MimDilate() and MimErode() now support M_BINARY_ULTIMATE_ACCUMULATE.
- New: MbufGetList() and MbufPutList() functions to set/get pixel values at arbitrary positions.
- New: MimProjection() now supports the M_MEAN projection mode.
- New: MbufGetArc() now supports returning the arc positions.
- Improvement: MimResize(M_NEAREST_NEIGHBOR) has been optimized for binary buffers.
- Improvement: MimPolarTransform(M_POLAR_TO_RECTANGULAR) accuracy has been improved for large radii.
- Improvement: subpixel annotations are now available for MimLocatePeakId() results when calling MimDraw() with M_DRAW_PEAKS + M_CROSS.
- Improvement: MgraArcFill() and MgraArc() results and speed have been improved.
- Improvement: MbufGetLine() and MbufPutLine() now support providing the start and end positions outside the buffer limits.

2.16 General

- New: MdigProcess() now supports M_REGION_DELETE_AT_EACH_FRAME to delete any region that could have been associated to the modified buffer in a previous iteration.
- New: MfuncAllocScript(M_INTERPRETER_C_PYTHON35) to allocate a script-based user-defined function for Python 3.5.
- New: MIL for Python now supports the MIL_FONT_NAME() functionality.
- New: MIL for Python now supports the M_PTR_TO_DOUBLE() functionality.
- New: MdigProcess() now supports M_REGION_DELETE_AT_EACH_FRAME to force deleting any ROI that could have been associated to the modified buffer in a previous iteration.

2.17 Utilities

- The DispD3D DLL utility now offers to:
 - set the initial 3d view of the scene,
 - optionally display the point clouds relative coordinate systems,
 - optionally display a point cloud's extraction box.

2.18 Distributed MIL

- Improvement: the default cluster type has been changed from single to multiple clusters, unless Distributed MIL Server Process or Cluster Type settings in MILConfig have been modified prior to this update.

2.19 MIL Help

- New: The Filters button displays a Filter Values by dialog box with a list of different possible filters. The Filter Values dialog box can also be opened using the Filters menu item in the grey bar. Choose the filter you want, and then press Apply. The filters only apply to values within tables. Note that filters are not persistent between different pages, or when you change page display settings, such as the MIL Systems settings or the detailed/summarized view. A dialog prompt will appear when a setting change causes your filters to reset.
- New: Function descriptions and parameters can now be collapsed. Click on the arrow head to the left of the description or parameter title and it will hide (collapse) everything inside that title. This functionality is similar to clicking on the arrow heads to the left of a parameter name (in a parameter association table) or a value (in a table).
- New: Some concepts in the User Guide are now described using animations (for example, fixturing).
- Improvement: Constant values listed in the index now jump to their description on their reference page (instead of jumping to the top of the page).
- Improvement: The data type of the address to pass to void parameters is now always described near the independent parameter value that establishes the required data type.

3. Deprecated functionalities

3.01 Calibration module

- M_GRID_CORNER_HINT_X is deprecated and replaced by M_GRID_HINT_PIXEL_X.
- M_GRID_CORNER_HINT_Y is deprecated and replaced by M_GRID_HINT_PIXEL_Y.
- McalGrid() combination flags M_Y_AXIS_UP and M_Y_AXIS_DOWN are deprecated and replaced by M_Y_AXIS_COUNTER_CLOCKWISE and M_Y_AXIS_CLOCKWISE.
- McalInquire() M_Y_AXIS_UP flag is deprecated and replaced by M_Y_AXIS_DIRECTION.

3.02 Code Reader module

- M_DOT_SPACING is deprecated and replaced by the range of spacing values defined by M_DOT_SPACING_MIN and M_DOT_SPACING_MAX.

3.03 Pattern Matching module

- The "model based" Pattern Matching API has been deprecated and replaced by the following "context based" Pattern Matching API:

- MpatAllocAutoModel() is replaced by MpatAlloc() + MpatDefine(M_AUTO_MODEL).
- MpatAllocModel() is replaced by MpatAlloc() + MpatDefine(M_REGULAR_MODEL).
- MpatAllocRotatedModel() is replaced by MpatAlloc() + MpatDefine(M_REGULAR_MODEL) with the M_SEARCH_ANGLE_MODE set to a specific value, and the M_SEARCH_ANGLE_DELTA_NEG/POS set to 0.
- MpatCopy() is replaced by MpatDraw().
- MpatFindModel() is replaced by MpatFind().
- MpatFindMultipleModel() is replaced by MpatFind() with MpatControl(M_SEARCH_MODE).
- MpatPreprocModel() is replaced by MpatPreprocess().
- MpatGetNumber() is replaced by MpatGetResult(M_NUMBER).
- MpatSetAcceptance() is replaced by MpatControl(M_ACCEPTANCE).
- MpatSetAccuracy() is replaced by MpatControl(M_ACCURACY).
- MpatSetAngle() is replaced by MpatControl().
- MpatSetCenter() is replaced by MpatControl() with M_REFERENCE_X and M_REFERENCE_Y flags.
- MpatSetCertainty() is replaced by MpatControl(M_CERTAINTY).
- MpatSetDontCare() is replaced by MpatMask().
- MpatSetNumber() is replaced by MpatControl(M_NUMBER).
- MpatSetPosition() is replaced by MpatControl() with M_SEARCH_OFFSET_X, M_SEARCH_OFFSET_Y, M_SEARCH_SIZE_X and M_SEARCH_SIZE_Y flags.
- MpatSetSearchParameter() is replaced by MpatControl().
- MpatSetSpeed() is replaced by MpatControl(M_SPEED).

If required, declare the #define M_MIL_WARN_ON_DEPRECATED_MPAT directive set to 0 to discard the compilation warnings associated to the use of the deprecated Mpat... APIs.

- Note that the prototypes of MpatSave() and MpatRestore() have been changed for symmetry with other MIL save/restore APIs. While function overloads exist to preserve the compatibility with former versions, we strongly recommend upgrading to the new API.
- Note that the prototypes of MpatInquire() and MpatGetResult() have been changed to accommodate a new Index parameter. While function overloads exist to preserve the compatibility with former versions, we strongly recommend upgrading to the new Pattern Matching APIs.
- Note to C users: function prototypes for MpatInquire, MpatGetResult, MpatSaveA, MpatRestoreA, MpatSaveW, MpatRestoreW, MpatSave, and MpatRestore have changed. By default, these functions map to the old prototype to preserve backward compatibility. C users must add "#define M_USE_OLD_MPAT_API_IN_C 0" before including "mil.h" to use the new prototypes.

- M_ACCEPTANCE_THRESHOLD is deprecated and replaced by M_ACCEPTANCE in MpatInquire().
- M_CERTAINTY_THRESHOLD is deprecated and replaced by M_CERTAINTY in MpatInquire().
- M_CENTER_X is deprecated and replaced by M_REFERENCE_X in MpatInquire().
- M_CENTER_Y is deprecated and replaced by M_REFERENCE_Y in MpatInquire().
- M_POSITION_ACCURACY is deprecated and replaced by M_ACCURACY in MpatInquire().
- M_NUMBER_OF_OCCURRENCES is deprecated and replaced by M_NUMBER in MpatInquire().
- M_MIN_SPACING_X is deprecated and replaced by M_MIN_SEPARATION_X in MpatInquire() and MpatSetSearchParameter().
- M_MIN_SPACING_Y is deprecated and replaced by M_MIN_SEPARATION_Y in MpatInquire() and MpatSetSearchParameter().
- M_POSITION_START_X is deprecated and replaced by M_SEARCH_OFFSET_X in MpatInquire().
- M_POSITION_START_Y is deprecated and replaced by M_SEARCH_OFFSET_Y in MpatInquire().
- M_POSITION_UNCERTAINTY_X is deprecated and replaced by M_SEARCH_SIZE_X in MpatInquire().
- M_POSITION_UNCERTAINTY_Y is deprecated and replaced by M_SEARCH_SIZE_Y in MpatInquire().
- M_MODEL_INDEX is deprecated and replaced by M_INDEX in MpatGetResult().
- MpatAllocResult() NbEntries parameter has been replaced by ControlFlag and must now be set to M_DEFAULT when using the new API.
- MpatInquire(M_ALLOC_TYPE) is deprecated in the new API and replaced by MpatInquire(M_CONTEXT_TYPE) and MpatInquire(M_MODEL_TYPE).
- Note that an old Pattern Matching model file can be restored into a new Pattern matching context using MpatRestore() with the ControlFlag set to M_PAT_MODEL_TO_CONTEXT. To restore additional model files into a context, use MpatStream(M_LOAD) with the control flag set to M_PAT_MODEL_TO_CONTEXT.

Ex:

```
MIL_ID CtxId = MpatRestore(MIL_TEXT("Old1.mod"), SysId, M_PAT_MODEL_TO_CONTEXT, M_NULL);
MpatStream(MIL_TEXT("Old2.mod"), M_NULL, M_LOAD, M_FILE, M_DEFAULT, M_PAT_MODEL_TO_CONTEXT, &CtxId, M_NULL);
MpatStream(MIL_TEXT("Old3.mod"), M_NULL, M_LOAD, M_FILE, M_DEFAULT, M_PAT_MODEL_TO_CONTEXT, &CtxId, M_NULL);
MpatStream(MIL_TEXT("Old4.mod"), M_NULL, M_LOAD, M_FILE, M_DEFAULT, M_PAT_MODEL_TO_CONTEXT, &CtxId, M_NULL);
```

3.04 Edge Finder module

- M_GENERAL_FERET is now deprecated and replaced by M_FERET_GENERAL.
- M_GENERAL_FERET_ANGLE is now deprecated and replaced by M_FERET_GENERAL_ANGLE.
- M_DRAW_GENERAL_FERET is now deprecated and replaced by M_DRAW_FERET_GENERAL.

3.05 Blob module

- MblobControl() M_LATTICE is deprecated and replaced by M_CONNECTIVITY.
 - M_GENERAL_FERET is deprecated and replaced by M_FERET_GENERAL.
 - M_GENERAL_MOMENT is deprecated and replaced by M_MOMENT_GENERAL.
 - M_ALL_FERETS is deprecated and replaced by M_FERET_DIAMETERS.
- The "feature list based" blob API has been deprecated and replaced by the following "context based" blob API:

- MblobAllocFeatureList() is replaced by MblobAlloc().
- MblobCalculate() parameters have changed when using a blob context: MblobCalculate(IdentImBufId, GrayImBufId, FeatureListId, ResultId) is now replaced by: MblobCalculate(ContextId, IdentImBufId, GrayImBufId, ResultId)
- MblobAllocResult() has new parameters for symmetry with other modules.
- MblobGetResult() has a new parameter LabelOrIndex. LabelOrIndex can be set to M_DEFAULT (= M_GENERAL or M_INCLUDED_BLOBS depending on the specified ResultType).
- MblobSelectFeature() is replaced by MblobControl() to enable/disable the calculation of blob features. In addition:
 - MblobSelectFeature(M_XXX + M_FERET_CONTACT_POINTS_XXXX) is now:
 - MblobControl(M_FERET_CONTACT_POINTS, M_ENABLE)
 - MblobControl(M_GROUP_WITH_M_XXX, M_ENABLE)
 - MblobSelectFeature(M_XXX + M_SORTn_DOWN/M_SORTn_UP/M_NO_SORT) is now:
 - MblobControl(M_SORTn, M_XXX / M_NO_SORT) with
 - MblobControl(M_SORTn_DIRECTION, M_SORT_UP / M_SORT_DOWN) with M_XXX a sortable ResultType.
 - MblobSelectFeature(M_AREA) has no MblobControl() since this feature is always calculated.
 - MblobSelectFeature(M_NO_FEATURES) is replaced by MblobControl(M_ALL_FEATURES, M_DISABLE).
- MblobSelectMoment() is replaced by:
 - MblobControl(M_MOMENT_GENERAL + [M_BINARY / M_GRAYSCALE], M_ENABLE)
 - MblobControl(M_MOMENT_GENERAL_MODE, M_CENTRAL / M_ORDINARY)
 - MblobControl(M_MOMENT_GENERAL_ORDER_X)
 - MblobControl(M_MOMENT_GENERAL_ORDER_Y).
- MblobSelectFeret() is replaced by:
 - MblobControl(M_FERET_GENERAL) and
 - MblobControl(M_FERET_GENERAL_ANGLE)
- MblobGetNumber() is replaced by MblobGetResult(M_NUMBER).
- MblobGetResultSingle() is replaced by MblobGetResult() using the M_BLOB_LABEL() macro to specify the LabelOrIndex parameter.
- MblobGetRuns() is replaced by MblobGetResult() with M_RUN_X, M_RUN_Y and M_RUN_LENGTH. The size of the array can be retrieved using MblobGetResult(M_TOTAL_NUMBER_OF_RUNS).
- MblobFill() is replaced by MblobDraw(M_DRAW_BLOBS).
 - The ContextGraId is used to specify the value with which to fill the blobs.
 - Add "M_DRAW_BLOBS_CONTOUR + M_DRAW_HOLES_CONTOUR" to the operation to specify that the blob's borders should be filled (same result as the deprecated MblobFill(+M_CONTOUR) operation).
 - Note that MblobDraw() does not propagate the calibration.
- The following controls, that previously apply to a blob result, are deprecated and now apply to the new "context based" blob, so it can be saved and restored:
 - M_BLOB_IDENTIFICATION
 - M_FERET_ANGLE_SEARCH_END
 - M_FERET_ANGLE_SEARCH_START
 - M_FOREGROUND_VALUE
 - M_IDENTIFIER_TYPE
 - M_CONNECTIVITY (formerly named M_LATTICE)
 - M_NUMBER_OF_FERETS
 - M_TIMEOUT
 - M_PIXEL_ASPECT_RATIO
 - M_MAX_BLOBS
 - M_RETURN_PARTIAL_RESULTS
 - M_SAVE_RUNS

Note to C users: function prototypes for MblobAllocResult and MblobGetResult have changed. By default, these functions map to the old prototype to preserve backward compatibility. C users must add "#define M_USE_OLD_MBLOB_API_IN_C 0" before including "mil.h" to use the new prototypes.

3.06 Primitives

- MimProject() is deprecated and replaced by MimProjection(). For example, MimProject(SrcId, ProjListOrDestId, M_0_DEGREE) is replaced by MimProjection(SrcId, ProjListOrDestId, M_0_DEGREE, M_SUM, M_NULL)
- M_DCT8x8 control value is deprecated and replaced by M_DCT8X8.
- MimStat() is deprecated and is replaced by MimStatCalculate().

For example:

```
MimAllocResult( SystemId, M_DEFAULT, M_STAT_LIST, &StatResultId);
...
MimStat(ImageId, StatResultId, M_SUM + M_MIN, M_IN_RANGE, 3, 24);
MimGetResult(StatResultId, M_SUM, &SumVar);
...
MimFree(StatResultId);
```

is deprecated and must be replaced by:

```
MimAlloc( SystemId, M_STATISTICS_CONTEXT, M_DEFAULT, &StatContextId);
MimAllocResult( SystemId, M_DEFAULT, M_STATISTICS_RESULT, &StatResultId);
MimControl(StatContextId, M_SUM, M_ENABLE);
```

```

MimControl(StatContextId, M_MIN, M_ENABLE);
MimControl(StatContextId, M_CONDITION, M_IN_RANGE);
MimControl(StatContextId, M_COND_LOW, 3);
MimControl(StatContextId, M_COND_HIGH, 24);
...
MimStatCalculate(StatContextId, ImageId, StatResultId, M_DEFAULT);
MimGetResult(StatResultId, M_SUM, &SumVar);
...
MimFree(StatResultId);
MimFree(StatContextId);

- MimStatMultiple() is deprecated and replaced by MimStatCalculate.
MimAlloc: M_STAT_MULTIPLE_CONTEXT becomes M_STATISTICS_CUMULATIVE_CONTEXT.
MimAllocResult: M_STAT_MULTIPLE_RESULT becomes M_STATISTICS_RESULT.
- MimResize M_FAST and M_REGULAR control values are now deprecated.

```

3.07 General

- MbufControlNeighborhood() is deprecated and replaced by MbufControl().
- Deprecated MIL functions are not available in the MIL Python Wrapper.
- MappControlObject and MappInquireObject are deprecated and replaced by MobjControl and MobjInquire.

4. Fixed bugs

4.01 Blob module

- MblobDraw(M_DRAW_CONVEX_HULL) will no longer draw only a single point when the convex hull contains 2 points.
- MblobCalculate will no longer potentially crash in rare situations when M_LABELED_TOUCHING is set to M_BLOB_IDENTIFICATION.
- MblobMerge() will no longer crash when the SrcResult2 is set to M_NULL.
- MblobMerge() will no longer crash when using a result invalidated by a call to MblobControl().
- MblobSelect() with M_BLOB_TOUCHING_IMAGE_BORDERS will no longer lead to invalid results when using M_MERGE.
- MblobGetResult(M_BLOB_TOUCHING_IMAGE_BORDERS) will no longer lead to invalid results or potentially crash when some blobs are excluded or deleted.
- MblobSelect() with M_BLOB_TOUCHING_IMAGE_BORDERS will no longer result in a crash when M_SAVE_RUNS is set to M_DISABLE.
- MblobDraw() will no longer result in a crash when providing an invalid label.
- MblobSelect() will no longer crash when using M_MERGE along with sorting keys.
- MblobSelect() will no longer miss reporting an error when used on a result which has never been calculated.
- The module will no longer miss reporting an error when M_SAVE_RUNS feature is disabled while required for the calculation of the blob convex hull and for the calculation of the ferets when the number of ferets is set to infinite.
- MblobControl() with M_FERET_ANGLE_SEARCH_START and/or M_FERET_ANGLE_SEARCH_END will no longer discard the result when the value remains unchanged.
- MblobCalculate() will no longer perform a calculation instead of an expected post-calculation when buffer(s) are allocated on a different system than the system of the blob objects (context/result).

4.02 Calibration module

- McalSetCoordinateSystem() will behave correctly when using M_ROTATION_AXIS_ANGLE with a non-unit axis vector.
- McalGrid() will no longer crash when using a circle grid with calibration dots that consist of pure black pixels only.
- McalFixture(M_LEARN_OFFSET, M_MODEL_MOD) will no longer generate an error when using a circle shape (M_SHAPE_CIRCLE) context.
- McalGrid() with a non-null GridOffsetZ will no longer lead to invalid results when transforming coordinates.
- McalGrid() and McalList() will no longer potentially crash due to numerical instabilities when using the M_3D_ROBOTICS mode.
- McalGrid() will no longer potentially fail calibrating from a square chessboard oriented at about 45 degrees when no user hint point is provided.
- McalFixture() with an M_RESULT_PAT will no longer ignore the region angle.
- McalUniform() will no longer forget to reset the relative coordinate system.
- McalControl() no longer interpret M_DEFAULT as a value for the controls M_PRINCIPAL_POINT_X and M_PRINCIPAL_POINT_Y.

4.03 Code Reader module

- McodeRead() will no longer rarely fail to read QrCode Model 2 version 1 and QrCode Model 1 when M_DISTORTION is set to M_UNEVEN_GRID_STEP.
- McodeGrade() will no longer rarely fail to grade QrCode Model 2 version 1 and QrCode Model 1 when M_DISTORTION is set to M_UNEVEN_GRID_STEP.
- McodeRead() will no longer potentially return an M_STATUS_READ_OK and a wrong position results when a time-out occurs.
- McodeGrade() will no longer return several wrong ISO and AIM-DPM grades when grading flipped Datamatrix, Aztec, QrCode or MicroQrCode.
- McodeGrade() will no longer potentially return the wrong grade of version information when grading a QrCode.
- McodeGrade() will no longer return wrong grades for Datamatrix, Aztec, QrCode or MicroQrCode, when the extended area partly falls outside the image.
- McodeRead() will no longer crash when reading a composite code using a small time-out.
- McodeRead() will no longer crash when reading M_MAXICODE with an M_DOT_SPACING sets to a non null value.

- McodeRead() will no longer return a wrong string when reading a PDF417 or truncated PDF417 while most of the code error detection capacity is used.
- McodeGrade() will no longer fail when grading a GS1-128 composite code.
- McodeRead() will no longer potentially perform short read of M_PHARMACODE codes when the M_SEARCH_ANGLE_STEP is enabled.
- The grading of Interleaved 2 of 5 start and end codes will no longer be wrongly computed.
- M_FOREGROUND_VALUE sets to M_FOREGROUND_ANY will now be correctly handled for Aztec, MicroQr and Qr codes when M_DOT_SPACING is set to a non null value.
- McodeGrade() will no longer ignore the M_FLIP control flag value when grading 1D code.
- McodeGrade() will no longer potentially return a wrong grade for the few farthest top and bottom scan lines when grading very long 1D barcode (~450 pixels and more).
- McodeRead() with M_DISTORTION set to M_PERSPECTIVE_UNEVEN_GRID_STEP will no longer potentially crash in images of high complexity.
- McodeRead() will no longer try decoding the same region twice when M_USE_PRESEARCH is set to M_STAT_BASE, in some particular situations.
- McodeRead() will no longer potentially fail when reading in a green or blue child band buffer with an associated region at angle.
- McodeRead() will no longer potentially crash when using a large scan line height with a position accuracy set to high.
- McodeRead() will no longer convert Aztec code FNCL to <GS> when the string format is not set to M_GS1_RAW_DATA.
- McodeRead() will no longer return an invalid string when the number of detected errors is greater than the error capacity for an Aztec code.
- McodeGrade() for Interleaved 2 of 5 will no longer return an erroneous Decode Grade 'F' for a given scan reflectance profile.
- McodeRead() could return a wrong position for M_MICROPDF417. This is no longer the case.
- McodeRead() will no longer return an invalid string for M_4_STATE codes when the M_ENCODING is set to M_ENC_US_MAIL.
- McodeRead() will no longer ignore the validation of the Delivery Point Suffix (DPS) at the end of a UK Mail string.
- McodeRead() will no longer ignore the value of the M_CODE_FLIP control for M_4_STATE, M_PLANET and M_POSTNET codes.
- In very rare situations, McodeRead() could return two results for a single linear code. This is no longer the case.
- McodeRead() will no longer take into account the value of M_CODE_FLIP for M_PHARMACODE as it can always be read in both direction. McodeRead() will only use the search angle range instead.
- McodeRead() and McodeGrade() will no longer terminate with an exception when reading or grading in some rare situations of QR-Code.
- McodeVerify() potential minor memory leak has been fixed.
- McodeGrade() will no longer potentially return a wrong Datamatrix grades for L1 and L2 segments for the fixed pattern damage.

4.04 Edge Finder module

- MedgeCalculate will no longer lead to erroneous edges when using a 16 bit signed source buffer.
- The module will no longer return inconsistent calibrated and uncalibrated results after applying a rare combination of selections, including a relational selection.

4.05 Model Finder module

- M_SHAPE_CIRCLE occurrences will no longer be retrieved outside the search region when using a reference point different from the default centre point.
- M_SHAPE_CIRCLE context will no longer use an erroneous search region when the target is calibrated.
- M_SHAPE_CIRCLE context could potentially crash in rare situations.
- Generally speaking, a few bug fixes are improving the robustness of an M_SHAPE_CIRCLE context.
- MmodFind() will no longer crash when matching an M_GEOMETRIC or an M_GEOMETRIC_CONTROLLED context with an M_SHAPE_CIRCLE result buffer.
- When using an M_SHAPE_CIRCLE context, the M_SCALE result and the forward and reverse coefficients will no longer be wrong when returned in pixel units.
- The M_POLARITY result will no longer be reversed for an M_SHAPE_CIRCLE context when the Y axis coordinate of the calibration is up.
- MmodControl() with a M_SHAPE_CIRCLE result will no longer generated an error when the Index is set to M_DEFAULT.
- MmodFind() with an M_SHAPE_CIRCLE context will no longer potentially crash when the time-out setting is disabled.
- MmodFind() with an M_SHAPE_CIRCLE context will no longer potentially return an occurrence with a model coverage score greater than 100.
- MmodDraw() will no longer sometime let some initialized pixels when drawing a model image.
- MmodPreprocess() will no longer potentially crash, in very rare conditions, for a Geometric Controlled context.
- Two occurrences will no longer potentially share some same edges when M_SHARED_EDGES is set to M_DISABLE for a M_GEOMETRIC Context.

4.06 Metrology module

- Constructed inner fit and outer fit segment will no longer potentially result in erroneous result.
- The extremities of a fitted arc will no longer be potentially wrongly determined when the parent frame is rotated.
- An extremity of a fitted arc will no longer be potentially wrongly found when its sub-pixel location falls outside the ring sector region.
- Constructed fitted features will no longer ignore the region. The feature is now determined using the data included into the region.
- The construction of a point derived from circle will no longer lead to

- invalid results when moving the parent local frame.
- Cloning a multiple measured point feature will no longer clone a single point only.
- In rare situations, the fit of a segment could lead to an invalid result. This is no longer the case.
- MmetStream() will no longer crash when the ObjectIdPtr is set to M_NULL.
- The minimum distance tolerance between a segment and an arc will no longer return the wrong value, in a specific configuration of the features.
- A constructed intersection point between a circle and another feature, such as an arc, will no longer potentially lead to the wrong constructed point.
- The minimum distance between a circle and a segment will no longer return an erroneous result in some configurations of the two geometries.

4.07 Measurement module

- MmeasSetMarker(M_DRAW_PROFILE_SCALE_OFFSET) will no longer report an error when set to M_AUTO_SCALE_PROFILE or M_AUTO_OFFSET_PROFILE.

4.08 Pattern Matching module

- Calls to MpatFind() (formerly MpatFindModel) will no longer potentially modify the M_RESULT_OUTPUT_UNITS state.
- MpatPreprocess() (formerly MpatPreprocModel) will now return an error message when an error occurred during the preprocessing operation and inquiring M_PREPROCESSED will return M_FALSE in this situation.
- MpatFind() (formerly MpatFindModel) will no longer potentially find less occurrences than expected when the angular search and the M_SEARCH_ANGLE_ACCURACY are enabled.
- MpatRestore()/MpatStream() will no longer crash when loading a corrupted model file.
- Improvement: MpatFind() (formerly MpatFindModel) will no longer use a different rounding mode to calculate the score when using the AVX2 instruction set.
- In rare situations, MpatFind() (formerly MpatFindModel) could eliminate good occurrences when using a first level greater than 0. This is no longer the case.
- In rare situations, MpatFind() (formerly MpatFindModel) could return occurrences with a null score. This is no longer the case.
- MpatFind() (formerly MpatFindModel) will no longer fail to find occurrences on the border of the search region.
- MpatFind() (formerly MpatFindModel) will no longer potentially find occurrences outside the limits of the search region.
- MpatGetResult(M_ANGLE) with M_RESULT_OUTPUT_UNITS result setting set to M_WORLD was not transformed in world coordinates system. This is no longer the case.
- The deprecated MpatFindMultipleModel() will no longer return invalid results when using a region with M_SEARCH_ANGLE set to a value other than 0. It will now generate an error.
- The deprecated MpatFindMultipleModel(M_FIND_BEST_MODELS) will no longer wrongly return results relative to the region but relative to the target image.
- The deprecated MpatSetSearchParameter() with a ControlValue of 1409 will no longer be erroneously converted to -32768 for ControlType M_ALLOC_OFFSET_X, M_ALLOC_OFFSET_Y and M_COARSE_SEARCH_ACCEPTANCE.
- The deprecated MpatSetSearchParameter() with a ControlValue of 1409 will no longer erroneously set the control to its default value for ControlType M_LAST_LEVEL, M_FAST_FIND, M_MODEL_STEP, M_MIN_SEPARATION_X, M_MIN_SEPARATION_Y and M_MAX_INITIAL_PEAKS.
- In general, the Pattern Matching parameter checking has been improved with the new API and is now more strict.

4.09 Ocr module

- MocrGetResult() will no longer crash when retrieving the results with the M_SELECT_STRING control set to M_ALL.
- MocrGetResult() will no longer pop an error message when retrieving the result M_STRING + MIL_TEXT_CHAR with the result control M_SELECT_STRING sets to M_ALL.

4.10 3dMap module

- M3dmapExtract() will no longer generate incorrect associated calibrations to destination child buffers.
- M3dmapPut() now correctly handles invalid points (at least one of the coordinate is set to M_INVALID_POINT).
- M3dmapAlign() RMS error M_ALIGN_RMS_ERROR_THRESHOLD and M_ALIGN_RMS_ERROR_RELATIVE_THRESHOLD stop conditions are now tested using '<=' as documented instead of '<'.
- M3dmapAlign() will no longer lead to invalid results when using a user defined pre-alignment while the M_PREALIGNMENT_MODE control is set to M_CENTROID.
- M3dmapGet() with M_INTENSITY now returns 0 as the intensity when the point is invalid.
- M3dmapExtract() with a context allocated on a remote system and image buffers allocated on a local system will now correctly associate a calibration to the image buffers.

4.11 Color module

- McolTransform() with a relative color calibration context using the M_COMPUTE_ITEM_STAT method will no longer produce saturated images when using color triplets as references.

4.12 Primitives

- MimMorphic(), with M_THICK and M_THIN, will no longer ignore values other than 0, 1 and M_DONT_CARE in a structuring element and will consider it as ones.
- The M_ORIENTATION_DATA_MEAN statistics will no longer return an invalid orientation when the whole image is at a multiple of 90 degrees.

- MimResize will now properly generate an error when up-scaling an image using M_FILL_DESTINATION with the M_AVERAGE, M_MIN and M_MAX modes.
- MimDraw() with M_DRAW_DEAD_PIXELS will now fill the image with 0 and buffer maximum value instead of zeros and ones.
- MimLocatePeakId() will no longer miss finding peaks after a peak with its width exceeding the maximum allowed value was detected.
- MimGetResultId output type will now correctly defaults to MIL_INT when an M_FIND_ORIENTATION_LIST was not allocated with the M_FLOAT attribute.
- MimDraw() will no longer erroneously draw an undecimated centered wavelet decomposition result.

4.13 Interactive utilities

- Flipped DIB buffers were not properly handled in the module UIs. The copy-paste of a MIL display into a UI will no longer fail.
- The CodeReader GUI will no longer pop an error message when drawing a read results after reading twice on a disk sequence, using child buffer, and with the "Do Process All" option enabled.

4.14 General

- The MIL display will no longer ignore the mouse events when the displayed buffer has a region of interest, the display has an associated graphics list, and when interactivity is enabled.

5. Known limitations and bugs

5.01 Interactive utilities

- In Inspector, the spacing and spacing variation controls are deprecated and should have been grayed-out.

5.02 MIL add-on to Microsoft Visual Studio

- If both MIL 32-bits and 64-bits are installed on the same system but only one of the installations is updated using MIL 10 Processing Pack 1, the MIL add-on will nevertheless show up in both cases suggesting the most recent MIL flags.

5.03 MIL Help

- CHM based help files does not work properly under the following circumstances:
 - . The operating System is Windows 7 embedded
 - . The latest Windows updates, including IE11, are applied to the system
 - . The user is logged in as a standard user (i.e part of the Users group)
 A work around is to log in with administrator privileges.

5.04 Pattern Matching

- MpatGetResult() with M_SUM_I, M_SUM_II, M_SUM_IM, M_SUM_M, M_SUM_MM and M_NUMBER_OF_PIXELS are not available in M_FIND_BEST_MODELS mode although M_SAVE_SUMS is set to M_ENABLE.

5.05 General

- When using MIL MfuncAllocScript() with Python 2.7.11, you must set the following additional environment variable PYTHONHOME to your Python installation path (eg. set PYTHONHOME=c:\python27).
- When using MIL MfuncAllocScript() with Python 3.4.4, you must set the following additional environment variable PYTHONHOME to your Python installation path (eg. set PYTHONHOME=c:\python34).

Section 4: Differences between MIL 10 Processing Pack 1 and MIL 10

!!! ATTENTION MATROX DESIGN ASSISTANT 4 USERS !!!

The installation of MIL 10, MIL 10 Processing Pack 1 and Matrox Design Assistant 4 may lead to a problematic configuration. The installation of Matrox Design Assistant 4 Service Pack 1 will be required to rectify the configuration issue.

Table of Contents for Section 4

| | |
|------|--------------------------------------|
| 1. | Overview |
| 2. | New functionalities and improvements |
| 2.01 | MIL processing specific examples |
| 2.02 | Bead module |
| 2.03 | Blob module |
| 2.04 | Calibration module |
| 2.05 | Code module |
| 2.06 | Color module |
| 2.07 | Edge Finder module |
| 2.08 | Measurement module |
| 2.09 | Model Finder module |
| 2.10 | Registration module |
| 2.11 | String Reader module |
| 2.12 | 3dmap module |
| 2.13 | Regions |
| 2.14 | Primitives |
| 2.15 | Matrox Profiler |
| 2.16 | General |
| 2.17 | Interactive utilities |
| 3. | Deprecated functionalities |
| 3.01 | Measurement module |
| 3.02 | Code module |
| 3.03 | 3dmap module |
| 3.04 | Primitives |
| 4. | Fixed bugs |

- 4.01 Bead module
- 4.02 Blob module
- 4.03 Calibration module
- 4.04 Code module
- 4.05 Color module
- 4.06 Measurement module
- 4.07 Metrology module
- 4.08 OCR module
- 4.09 Registration module
- 4.10 String Reader module
- 4.11 3dmap module
- 4.12 Pattern matching module
- 4.13 Model Finder module
- 4.14 Graphic functions
- 4.15 Primitives
- 4.16 Interactive utilities
- 5. Known limitations and bugs
 - 5.01 Code module
 - 5.02 Interactive utilities
 - 5.03 MIL add-on to Microsoft Visual Studio
 - 5.04 MIL Help

1. Overview

- MIL 10 Processing Pack 1 includes all the features of MIL 10. In addition, MIL 10 Processing Pack 1 includes new processing functionalities, performance optimizations, and general improvements such as:
 - . new support for Aztec and Industrial 2 of 5 codes,
 - . new color relative calibrations,
 - . new support and calibration for multiple 3D camera-laser systems,
 - . new 3D surface alignment operation,
 - . new extended depth of field operation,
 - . new dedicated circle shape matcher,
 - . support for JIT C#, VB and CPython with MIL,
 - . MIL optimized for AVX2 instructions set,
 - . new versatile example launcher utility,
 - . new extensive wavelet transform operations,
 - . new local segmentation primitive,
 - . new MIL add-on for Microsoft Visual Studio,
 - . and many more to discover!
- MIL 10 is the minimum requirement for all upcoming Processing Packs until the next major release.

2. New functionalities and improvements

2.01 MIL processing specific examples

- New: a new version of the example launcher with categories, previews, keywords, rich information, and a search utility.
- 30+ new MIL examples!
- Note that the MIL 10 Update 24 is needed to get the images required for several examples, in particular for 3D processing examples.

2.02 Bead module

- New: `M_ADD_FROM_GRAPHIC_LIST` to add a bead template defined using a graphics list.

2.03 Blob module

- New: `M_NUMBER_OF_FERETS` can be set to `M_INFINITE` to calculate ferets and feret-based features with a high accuracy.
- New: `M_CONVEX_HULL_PERIMETER` is now supported.
- New: `M_BLOB_CONTRAST` to calculate the difference between the blob min/max pixel grayscale values is now supported.
- New: `M_FERET_ANGLE_SEARCH_START` and `M_FERET_ANGLE_SEARCH_END` to limit the angular range used to determine the blob's ferets is now supported.
- New: `M_FERET_CONTACT_POINTS` (ex. `M_FERET_MAX_DIAMETER + M_FERET_CONTACT_POINTS`) is now supported.
- New: `M_FERET_CONTACT_POINTS_X1,...` to retrieve the coordinates of the corresponding contact points are now supported.
- New: `M_MIN_AREA_BOX` is now supported.
- New: `M_DRAW_MIN_AREA_BOX` to draw the `M_MIN_AREA_BOX` feature is now supported.
- New: `M_MIN_PERIMETER_BOX` is now supported.
- New: `M_DRAW_MIN_PERIMETER_BOX` to draw the `M_MIN_PERIMETER_BOX` feature is now supported.
- New: the minimum area enclosing rectangle feature is now available.

2.04 Calibration module

- New: `M_CALIBRATION_MODE` can now be inquired on any object accepted by `McalInquire`.
- New: `M_CALIBRATION_CATEGORY` to inquire whether the calibration is an `M_2D_CALIBRATION` or an `M_3D_CALIBRATION`.
- New: `M_DEPTH_MAP` to inquire whether the MIL object is a fully corrected depth map or not.
- New: `M_CALIBRATION_INPUT_DATA` to inquire about the type of the data used to perform the calibration (ex. grid).
- New: `M_POINT_CLOUD_CONTAINER_3DMAP` `McalFixture` location type to use the relative coordinate system of a 3D reconstruction result buffer as a fixture provider.
- New: `M_LASER_3DMAP` `McalFixture` location type to use the laser line coordinate system of a 3D reconstruction context buffer as a fixture

- provider.
- New: McalSetCoordinateSystem can now be called on 3D reconstruction result buffers to modify the relative coordinate system, which is used to express points in world coordinates in M3dmapGet and which defines the projection orientation in M3dmapExtract.
- New: M_LASER_LINE_COORDINATE_SYSTEM is now supported. Its X-axis is parallel to the laser line and its Y-axis is parallel to the scanning direction.
- New: McalGetCoordinateSystem can now be called on a 3D reconstruction result buffer to access the relative coordinate system.
- New: McalGetCoordinateSystem can now be called on a 3D reconstruction context to access the laser line coordinate system.
- New: McalGrid GridOffsetZ can now be used to specify the thickness of the grid when performing a 3D calibration.
- Improvement: McalTransformCoordinate/3dList has been optimized for multi-core (MP) architectures.

2.05 Code module

- New: reading and grading multiple Data Matrix codes is now supported.
- New: M_INDUSTRIAL25 to read Industrial 2 of 5 codes.
- New: M_AZTEC to read and to grade Aztec codes.
- New: Add-On for EAN-8 code is supported.
- New: M_STRING_FORMAT now supports M_GS1_RAW_DATA.
- New: QR codes now support GS1 encoding.
- New: It is now possible to verify a code from the result of the reading operation using the new McodeGrade function.
- New: M_DRAW_QUIET_ZONE to draw the quiet zone around the decoded code.
- New: M_QUIET_ZONE_INCLUDED result to determine whether the quiet zone is fully included into the image borders or not.
- New: M_QUIET_ZONE_TOP... to retrieve the position of the quiet zone.
- New: verification of the extended quiet zone including Asterisk notation.
- New: M_EXTENDED_AREA_REFLECTANCE_CHECK to enable the verification using the extended area.
- New: M_DRAW_EXTENDED_AREA to draw the extended area when M_EXTENDED_AREA_REFLECTANCE_CHECK is enabled.
- New: M_EXTENDED_AREA_BOTTOM... and M_EXTENDED_AREA_TOP... to retrieve the position of the extended area.
- New: M_EXTENDED_AREA_QUIET_ZONE_INCLUDED result to determine whether the extended area and quiet zone are fully included into the image borders (ISO/IEC 15415:2011).
- New: M_EXTENDED_AREA_CODEWORD_MODULATION to retrieve the modulations for each codeword in the symbol in the extended area.
- New: M_EXTENDED_AREA_REFLECTANCE_MINIMUM to retrieve the lowest reflectance (Rmin) of any extended area in a 2D symbol, except Maxicode.
- New: M_EXTENDED_AREA_REFLECTANCE_MAXIMUM to retrieve the highest reflectance (Rmax) of any extended area in a 2D symbol, except Maxicode.
- New: M_ASTERISK to retrieve if an Asterisk is appended to the overall symbol grade. Valid for all 2D matrix symbol, except Maxicode.
- New: M_EXTENDED_AREA_SYMBOL_CONTRAST, M_EXTENDED_AREA_MODULATION_GRADE, M_EXTENDED_AREA_CODEWORD_MODULATION_GRADE, M_EXTENDED_AREA_SYMBOL_CONTRAST_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_L1_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_L2_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_QZL1_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_QZL2_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_CLOCKTRACK_SOLID_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_AVERAGE_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_A1_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_A2_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_A3_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_B1_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_B2_GRADE, M_EXTENDED_AREA_FIXED_PATTERN_DAMAGE_C_GRADE,... extended area verification results.
- New: M_FIXED_PATTERN_DAMAGE_L1_GRADE, M_FIXED_PATTERN_DAMAGE_L2_GRADE, M_FIXED_PATTERN_DAMAGE_QZL1_GRADE, M_FIXED_PATTERN_DAMAGE_QZL2_GRADE,... to retrieve the detailed information about the fixed pattern damages per segment, for ISO and AIM-DPM verification.
- New: Additional support for the ISO/IEC 15415:2011 verification: M_REFLECTANCE_MARGIN_GRADE, M_CODEWORD_REFLECTANCE_MARGIN, M_CODEWORD_REFLECTANCE_MARGIN_GRADE, M_CONTRAST_UNIFORMITY, and M_CONTRAST_UNIFORMITY_GRADE are now supported.
- New: M_CODEWORD_REFLECTANCE_MARGIN, M_CODEWORD_REFLECTANCE_MARGIN_GRADE and M_REFLECTANCE_MARGIN_GRADE verification grade are now supported.
- New: M_USE_PRESEARCH now supports M_FINDER_PATTERN_BASE mode for reading Data Matrix.
- New: M_FINDER_PATTERN_MINIMUM_LENGTH to set the shortest acceptable length of either "arm" of the finder pattern of a Data Matrix code is now supported.
- New: M_FINDER_PATTERN_EXHAUSTIVE_SEARCH is now supported for Data Matrix.
- Improvement: improved reading robustness when the M_CELL_NUMBER_X/Y_MIN or M_CELL_NUMBER_X/Y are specified.
- Improvement: improved robustness when reading a large 2D barcode with the presearch enabled.
- Improvement: AIM-DPM verification accuracy.
- Improvement: improved robustness when reading Maxi code and BC412 code.

2.06 Color module

- New: a color relative calibration tool which includes three (3) calibration methods.
- New: McolTransform can apply the color relative calibration transformation to an image.

2.07 Edge Finder module

- Improvement: the module has been optimized for multi-core (MP) architectures.

2.08 Measurement module

- New: M_DRAW_PROFILE_SCALE_OFFSET with M_AUTO_OFFSET_PROFILE control value to specify the offset for the profile drawing operations.
- New: M_DRAW_PROFILE_SCALE_OFFSET with M_AUTO_SCALE_PROFILE control value to specify the scale for the profile drawing operations.
- Improvement: it is no longer required to specify an edge modifier (+M_EDGE_FIRST/SECOND) when calling MmeasGetScore with (M_DISTANCE_FROM_BOX_ORIGIN_SCORE/M_SPACING_SCORE/M_EDGE_INSIDE_SCORE/M_STRIPE_WIDTH_SCORE + M_RESULT) on a stripe marker.

2.09 Model Finder module

- A new specialized circle finder is now supported.
- Improvement: robustness of the Geometric controlled context has been improved.
- Improvement: the module has been optimized for multi-core (MP) architectures.

2.10 Registration module

- New: the module now supports the Extended Depth of Field (EDF) fusion operation from stacked images.

2.11 String Reader module

- Improvement: the robustness of the fontless context has been improved.
- Improvement: the MstrRead memory exception safety has been improved.
- Improvement: MstrRead baseline deviation calculation has been improved when reading a string at angle.

2.12 3dmap module

- New: 3D surface-based alignment between a model's point clouds and an organized target's point cloud is now supported.
- New: M3dmapImport to import (ASCII and binary) PLY 3D CAD files is now supported.
- New: M3dmapClear to clear the contents of a 3D reconstruction result buffer is now supported.
- New: M3dmapPut to store world points and their features inside an unorganized point cloud.
- New: M3dmapGet to retrieve world point coordinates and their features from one or more point clouds is now supported.
- New: adding controls to filter out the 3D points according to the Z-axis during laser scanning process.
- New: M_EXTRACTION_RANGE_Z control, used by M3dmapAddScan, to specify the range of valid values along the Z-axis.
- New: M3dmapCalibrateMultiple to calibrate multiple 3D reconstruction contexts simultaneously.
- New: M3dmapCalibrate accuracy has been improved.
- New: M_CALIBRATION_STATUS can now return M_GLOBAL_OPTIMIZATION_ERROR when the convergence of the M3dmapCalibrateMultiple operation fails.
- New: M_CALIBRATED_CAMERA_LINEAR_MOTION+M_CAMERA_LABEL/M_LASER_LABEL(value) to specify the camera/laser label for a context.
- New: M_CAMERA_LABEL_VALUE and M_LASER_LABEL_VALUE inquires are now supported.
- New: the extraction of the depth map (M3dmapExtract) now takes into account the relative coordinate system of the 3D reconstruction result buffer. This allows merging depth maps from two different result buffers, adjusting the depth map according to a fitted plane, or creating slices of an object.
- New: M_EXTRACTION_SCALE_MODE can now be set to M_USE_DESTINATION_CALIBRATION to use the correction information and the relative coordinate system associated of the destination image buffer.
- New: M_EXTRACTION_OVERLAP to control the behaviour of the depth map generation when two or more 3D points projects onto the same depth map pixel.
- New: M_EXTRATE_Z to set whether or not to saturate Z coordinates that fall outside the region when generating the depth map.
- New: M_RESULTS_DISPLACEMENT_MODE and M_RESULTS_DISPLACEMENT_Y to set whether the point coordinates and the extracted depth maps are returned according to the object's initial position (before scanning) or according to any other position, including its current position.
- New: M_TOTAL_DISPLACEMENT_Y returns the total conveyor displacement.
- New: M_ASSUMED_PERPENDICULAR_TO_MOTION to inquire whether the "vertical" laser plane assumption was assumed during calibration.
- New: M3dmapArith() between two depth maps now allows having different relative coordinate systems.
- Improvement: improved numerical stability when accumulating scans in continuous mode.
- New: M_LOCATE_PEAK_ID_CONTEXT_ID can now be inquired for a 3D reconstruction context to retrieve its internal locate peak id context.
- New: M_SCAN_LANE_DIRECTION can now be retrieved for a 3D reconstruction result to obtain the scan lane direction using M3dmapGetResult.
- New: M3dmapSetBox function to easily define the extraction box to use to generate the depth map using M3dmapExtract.

2.13 Regions

- New: the following primitives now support regions:
 - . MimArithMultiple
 - . MimArith
 - . MimHistogramEqualize
 - . MimLocateEvent
 - . MimCountDifference
 - . MimStat

- . MimProject
- . MimFindExtreme
- . MbufClear
- . MbufClearCond
- . MimLutMap

2.14 Primitives

- New: MbufBayer now supports an M_NULL destination to determine the white balancing coefficients only.
- New: MimBinarizeAdaptive to perform a local segmentation operation is now supported.
- New: MimArith now supports M_ATAN2.
- New: MimHistogramEqualizeAdaptive (CLAHE) is now supported.
- New: MimFindOrientation to determine the dominant orientations in an image is now supported.
- New: MimStat now supports M_ANGULAR.
- New: MimStat now supports M_ORIENTATION_DATA_MEAN.
- New: MimConvert now supports conversion from RGB to Normalized RGB.
- New: MimHistogram now supports user-defined and automatic binning modes.
- New: MbufClone to easily clone a MIL buffer is now supported.
- New: M_HORIZONTAL_EDGE_SOBEL and M_VERTICAL_EDGE_SOBEL predefined FIR convolution kernels are now supported.
- New: M_HORIZONTAL_EDGE_PREWITT and M_VERTICAL_EDGE_PREWITT predefined FIR convolution kernels are now supported.
- New: MimResize now supports M_MAX and M_MIN interpolation modes.
- New: MimWaveletDenoise is now supported.
- New: MimWaveletTransform to perform wavelet decomposition and reconstruction is now supported.
- New: MimSetWaveletFilter to specify a user-defined wavelet is now supported.
- New: the M_FLAT_FIELD_CONTEXT gain constant can be now set to M_AUTOMATIC.
- New: M_EFFECTIVE_GAIN_CONST result to retrieve the value of the M_FLAT_FIELD_CONTEXT gain constant when automatically determined by the operation.
- New: MgenLutFunction now supports M_COLORMAP_JET, M_COLORMAP_HOT, M_COLORMAP_HUE and M_COLORMAP_SPECTRUM to generate pseudo-color palettes.
- Improvement: MgraLine has been specifically optimized for the case of straight vertical and horizontal lines.
- Improvement: MimLocatePeak accuracy, robustness, and flexibility have been improved.
- Improvement: MimPolarTransform now supports an angle range from -360 to 720 degrees.
- Improvement: MimPolarTransform precision has been improved.

2.15 Matrox Profiler

- New: traces into circular memory buffer are now supported.
- New: extensive global statistics information is now supported.
- New: the profiler sessions can now be saved and restored.
- New: the MIL constant are now translated into their equivalent names.
- New: many new interactivity features and improvements.
- More information can be found in the What's New Profiler read-me file!

2.16 General

- Improvement: loading .TIFF/.MIM large images files (>4GB) is now supported.
- New: MbufChildColor2dClip to allocate a clipped child buffer from one or more specified bands of a previously allocated parent buffer.
- New: MbufChildMove M_CLIP control flag to automatically adjust the parameters, if necessary, to make sure the child stays within its parent's buffer limits.
- New: M_OWNER_SYSTEM can now be inquired for the result buffers.
- New: most primitives and modules have been optimized for SSE4.1 and AVX2 instruction sets.
- New: MfuncAllocScript to run MIL code in JIT CPython, C# and VB .NET environments.
- New: MIL traces can now be performed into circular memory buffers.
- New: MappControl M_TRACE_SAVE_TO_FILE to dump a trace stored in memory into a file.
- New: MappTrace(M_TRACE_SET_TAG_INFORMATION) to customize the name and the color associated with a TraceTag is now supported.
- New: MappInquire(M_TRACE_ACTIVE) to retrieve whether traces are enabled.
- New: MthrAlloc M_TRACE_LOG_DISABLE control flag for M_THREAD object type to disable the log of traces for and at the creation of a thread.
- New: New Distributed MIL protocol "dmilshm" which uses shared memory for interprocess communication on a same machine (localhost) and which is intended to replace the use of dmiltcp://localhost.
- Improvement: M...GetResult and M...Inquire combination flag M_TYPE_SHORT has been renamed M_TYPE_MIL_INT16.
- New: MbufImport(M_MIL+M_WITH_CALIBRATION) can now be used to restore an image associated to M_DEFAULT_UNIFORM_CALIBRATION if the image has a constant pixel size and if the image was previously saved using MbufExport(M_MIL+M_WITH_CALIBRATION). If not, the function will restore an uncalibrated image.

2.17 Interactive utilities

- MIL add-on to Microsoft Visual Studio 2012 and up to:
 - . context sensitive help for C++, C# and VB
 - . statement completion for C++, C#, and VB
 - . MIL menu and toolbar
- Note that other Microsoft Visual Studio plugins which provide statement completion may interfere with the MIL extension for Visual Studio.

3. Deprecated functionalities

3.01 Measurement module

- M_POSITION_IN_MARKER is deprecated and replaced by M_DISTANCE_FROM_BOX_ORIGIN.
- ResultType M_FOUND_BOX_ANGLE is deprecated and replaced by M_BOX_ANGLE_FOUND.
- ResultType M_TOTAL_SCORE is deprecated and replaced by M_SCORE_TOTAL.
- ResultType M_BOX_EDGE_VALUES_NUMBER is deprecated and replaced by M_BOX_EDGEVALUES_NUMBER.
- ResultType M_BOX_EDGE_VALUES is deprecated and replaced by M_BOX_EDGEVALUES.
- M_TOTAL_SCORE is deprecated and replaced by M_SCORE_TOTAL.
- M_FOUND_BOX_ANGLE is deprecated and replaced by M_BOX_ANGLE_FOUND.
- M_POSITION_INSIDE_STRIPE is deprecated and replaced by M_INCLUSION_POINT_INSIDE_STRIPE.

3.02 Code module

- CodeVerify has been deprecated and replaced by the more flexible McodeGrade function.
- M_USE_PRESEARCH M_ENABLE control value is deprecated and replaced by M_STAT_BASE.
- M_MINIMUM_LSIDE_LENGTH control flag is deprecated and replaced by M_FINDER_PATTERN_MINIMUM_LENGTH.
- M_MAX_GAP control flag is deprecated and replaced by M_FINDER_PATTERN_MAX_GAP.

3.03 3dmap module

- M_EXTRACTION_SCALE_MODE M_USE_DESTINATION_SCALES control value is deprecated and replaced by M_USE_DESTINATION_CALIBRATION.
- The M_LASER_DATA result object is deprecated, use M_LASER_CALIBRATION_DATA, M_DEPTH_CORRECTED_DATA, or M_POINT_CLOUD_CONTAINER instead. See M3dmapAllocResult documentation.
- M3dmapAddScan with M_RESET, M_CLEAR_DATA, or M_REMOVE_LAST_SCAN is deprecated and replaced by M3dmapClear() with M_DELETE, M_CLEAR, and M_REMOVE_LAST_SCAN.
- M3dmapGetResult with M_3D_POINTS_X, M_3D_POINTS_Y, M_3D_POINTS_Z, M_3D_POINTS_I is deprecated and replaced by the M3dmapGet function.
- M3dmapGetResult with M_DEPTH_CORRECTION_STATE (M_UNCORRECTED, M_PARTIALLY_CORRECTED or M_FULLY_CORRECTED) is deprecated. 3D result objects now have specialized types. Use MappInquireObject() with M_OBJECT_TYPE to inquire the type of object.
- M_PEAK_WIDTH is deprecated. Use M_LOCATE_PEAK_ID_CONTEXT_ID to retrieve the internally allocated locate peak id context and configure it using MimControl with M_PEAK_WIDTH_NOMINAL and M_PEAK_WIDTH_DELTA.
- M_MIN_INTENSITY is deprecated. Use M_LOCATE_PEAK_ID_CONTEXT_ID to retrieve the internally allocated locate peak id context and configure it using MimControl with M_MINIMUM_CONTRAST.
- M_ORIENTATION is deprecated. Use M_LOCATE_PEAK_ID_CONTEXT_ID to retrieve the internally allocated locate peak id context and configure it using MimControl with M_SCAN_LANE_DIRECTION.

3.04 Primitives

- MimLocatePeakId now supports a new improved peak detection method. Using MimLocatePeakId without a context object is now deprecated.
- The default overscan for the morphological operations using a custom structural element, in binary mode, is now M_TRANSPARENT. MbufControlNeighborhood(StructElement, M_OVERSCAN, M_DISABLE) must be used to obtain the previous default behaviour.
- M_LAPLACIAN_EDGE predefined FIR convolution kernel is deprecated and replaced by M_LAPLACIAN_4.
- M_LAPLACIAN_EDGE2 predefined FIR convolution kernel is deprecated and replaced by M_LAPLACIAN_8.
- M_SHARPEN predefined FIR convolution kernel is deprecated and replaced by M_SHARPEN_8.
- M_SHARPEN2 predefined FIR convolution kernel is deprecated and replaced by M_SHARPEN_4.
- M_EDGE_DETECT predefined FIR convolution kernel is deprecated and replaced by M_EDGE_DETECT_SOBEL_FAST.
- M_EDGE_DETECT2 predefined FIR convolution kernel is deprecated and replaced by M_EDGE_DETECT_PREWITT_FAST.
- M_HORIZ_EDGE predefined FIR convolution kernel is deprecated.
- M_VERT_EDGE predefined FIR convolution kernel is deprecated.

4. Fixed bugs

4.01 Bead module

- The training operation will no longer crash when providing invalid world positions.

4.02 Blob module

- After a call to MblobMerge(M_MOVE), the MblobDraw(M_DRAW_BLOBS_CONTOUR) will no longer draw the contour of the blobs in the first top frame only.
- The MblobMerge operation will no longer result in the blob's convex hulls being wrongly offset.
- M_CONVEX_HULL_AREA will no longer return an invalid result for a convex hull made of 2 points.
- Successive calls to MblobCalculate will no longer result in wrong feret calculations when using multi-core acceleration together with M_WHOLE_IMAGE or M_LABELED_TOUCHING modes.
- M_CONVEX_HULL_AREA, M_CONVEX_HULL_COG_X and M_CONVEX_HULL_COG_Y could be wrongly computed under specific circumstances. This is no longer the case.
- BlobInquire will no longer crash when inquiring another flag than M_OWNER_SYSTEM.
- MblobReconstruct will no longer crash when providing a NULL seed buffer. It will now report an error.

4.03 Calibration module

- McalDraw(...+ M_DRAW..._COORDINATE_SYSTEM) will no longer lead to an infinite loop in the presence of degenerated calibrations.
- McalFixture with M_RESULT_PAT will no longer lead to the wrong result when the M_RESULT_OUTPUT_UNITS of the pattern matching result object is manually set to M_WORLD.
- McalRestore() will no longer restore a calibration object with a wrong relative coordinate system when restoring from a file containing an image with a constant pixel size and a relative coordinate system that is not coplanar to the absolute coordinate system. It will report an error. Use MbufImport(M_MIL+M_WITH_CALIBRATION).
- M_DISPLACE_CAMERA_COORD will no longer ignore the M_CALIBRATION_PLANE.

4.04 Code module

- McodeRead will no longer crash when reading in an image smaller than 3 pixels with M_USE_PRESEARCH set to M_ENABLE.
- McodeDraw index for M_DRAW_REFLECTANCE_PROFILE will no longer be wrongly validated.
- When reading a 1D code, the M_STATUS will no longer be M_STATUS_READ_OK but M_STATUS_NO_RESULT_AVAILABLE when no code has been found.
- McodeRead will no longer crash when reading Postnet/Planet code at angle in a very large buffer.
- Reading a Data Matrix code will no longer fail when the top right corner of the code falls outside the image, for uneven grid and perspective uneven grid modes.
- Data Matrix, QrCode, and MicroQrCode decode algorithm has been fixed when number of error correction was odd. Small 10x10 and 12x12 Data Matrix will no longer be erroneously found within larger codes or textured areas.
- McodeRead will no longer fail reading multiple identical overlapped 1D codes.

4.05 Color module

- McolDistance will no longer miss propagating the calibration of the source to the destination.
- McolProject will no longer miss propagating the calibration of the source to the destination.
- Restoring a color context with McolStream(... M_RESTORE ...) in DMIL will no longer lead to a new allocated MIL_ID with an invalid object type.

4.06 Measurement module

- MmeasGetResult/Single(M_EDGEVALUE_PEAK_WIDTH / M_EDGE_WIDTH) will now return both edge widths of a stripe marker.
- MeasGetResult/Single will no longer crash when trying to retrieve some result types not supported by a measurement result.
- MmeasDraw(M_DRAW_LINE) on a measurement result will no longer crash if no result was calculated.
- MmeasSaveMarker was not returning immediately after parameter checking detect an error. This is no longer the case.
- MmeasSetMarker(M_BOX_ANGLE_REFERENCE) was not validating the value provided by the user. This is no longer the case.
- MmeasSetMarker(M_BOX_ANGLE_MODE) was not validating the value provided by the user. This is no longer the case.
- MmeasSetMarker(M_BOX_ANGLE_TOLERANCE) was not validating the value provided by the user. This is no longer the case.
- MmeasSetMarker(M_SEARCH_REGION_ANGLE_INTERPOLATION_MODE) was not validating the value provided by the user. This is no longer the case.
- Under a specific set of conditions, M_EDGE_INSIDE_SCORE will no longer prevent finding a marker.
- When a score-offset is used for a given feature, the feature's score was not dropping to zero when the feature was outside the range. It now does.
- MmeasGetResult(M_BOX_EDGEVALUES) will no longer crash when the number of submarkers change from "more than one" to "a single one" between successive calls to MmeasFindMarkers.
- MmeasSetMarker(M_BOX_SIZE) will no longer accept a null value. The control must be set to a value greater than 0.
- The mean angle (for either edges or stripes) returned the M_MEAN combination flag is used will no longer be erroneous.
- MIL_TYPE_MIL_ID was not supported for inquiring about MIL_ID's information such as the M_OWNER_SYSTEM. It is now supported.
- Calculation of the edge limits has been improved for very small smoothed edges.
- MmeasGetResult(M_STRIPE, M_SPACING) will no longer require edge modifiers to get the spacing from its first/second edges while it did not when not there is no calibration involved.
- MmeasSetMarker will no longer update the first value when the first value is valid but not the second.

4.07 Metrology module

- Drawing an edgel feature will no longer draw all the edgels, but the active ones.
- Metrology will no longer wrongly support the orientation of 3D calibration systems.
- The calibrated drawing of arc regions will no longer lead to erroneous annotations when using low accuracy annotations.
- The parametric angle of a constructed local frame will no longer wrongly follow the global frame, but its local reference frame.
- The coverage measure of the fitted features will no longer be wrongly calculated in the presence of image aspect ratio.

4.08 OCR module

- A very rare situation of MocrReadString crash has been fixed.
- MocrCalibrateFont will no longer fail when changing the ROI or the step value.

4.09 Registration module

- MregSetLocation called with parameter type M_COPY_REG_RESULT after a call with M_WARP_4_CORNER, M_WARP_4_CORNER_REVERSE, or M_WARP_POLYNOMIAL, will no longer result in a memory leak.

4.10 String Reader module

- String Reader will no longer change the global SPI_SETFONTSMOOTHING system parameter when defining the system font.
- M_STRING_ANGLE_DELTA_NEG will no longer control the positive delta value and M_STRING_ANGLE_DELTA_POS will no longer control the negative delta value.
- The preprocess step will no longer crash in the presence of large characters in the font images.
- A potential out of memory error has been fixed when reading a string using a fontless context.
- MstrRead will no longer potentially dilate the user image buffer when using regions at 0 degrees and when the control M_THICKEN_CHAR is used.
- MstrInquire will no longer crash when a NULL destination is provided.

4.11 3dmap module

- Filling gaps with M_MIN or M_MAX will no longer result in different decisions in the depth map versus the intensity map.
- M3dmapGetResult() with M_NUMBER_OF_3D_POINTS on an empty result buffer will no longer return an invalid index error, but 0.

4.12 Pattern matching module

- MpatAllocModel call will no longer perform unnecessary computations when M_AUTO_CONTENT_BASED is not used.
- MpatAllocModel will no longer crash when one of the provided offset is negative.
- In rare situations, MpatFindModel could lead to unstable results when using a search region and the fast find mode.

4.13 Model Finder module

- MmodDraw will now give priority to the destination buffer calibration when drawing the edges of a context.

4.14 Graphic functions

- MgraRectAngle will no longer ignore the rotation of the relative coordinate system when drawing the shape at 0 degrees in world units with the conversion mode set to M_PRESERVE_SHAPE_AVERAGE.
- When calling MbufSetRegion(M_RASTERIZE) with a graphics list that has its interactivity enabled, the interactive handles will no longer wrongly appear in the rasterized region.

4.15 Primitives

- MimMorphic(M_DILATE, M_ERODE, M_THICK and M_THIN), in binary mode and with the overscan set to replace min or max, will no longer use the wrong overscan value. This affected only board systems that may perform the operation.
- MimResize(M_BILINEAR) accuracy has been fixed for very large buffers.
- MimDraw will no longer modify the unchanged pixels of the destination when drawing a stat-multiple result.
- MimDraw will no longer crash when drawing the model image or the mask image in a large destination buffer.
- MimLocateEvent will no longer return an invalid MIL_INT result when the result ID is M_NULL.
- MimCountDifference will no longer return an invalid MIL_INT result when the result ID is M_NULL.
- MimEdgeDetect will no longer potentially crash when the source image is a child band of a YUV buffer.
- MimPolarTransform(M_POLAR_TO_RECTANGULAR) accuracy of the returned destination sizes have been fixed.
- MimRank, in binary mode and with signed destination buffer, will no longer return incorrect results when the rank values of the structuring element are greater than 128 for an 8-bits destination buffer, or greater than 32768 when the destination buffer is 16-bits.
- MimResize will no longer access data outside the source buffer for rare specific conditions and specific scale values.

4.16 Interactive utilities

- A GDI memory leak has been fixed for the Model Finder, Edge Finder, and String Reader GUIs.
- A stability issue has been fixed when using the column ordering in the result table or the module's UIs.
- The new Model Finder circle matcher has been added to the Model Finder UI.

5. Known limitations and bugs

5.01 Code module

- Grading on read result is not yet supported for newly added Aztec codes.
- Grading is not supported for the special 11x11 Aztec code "rune".

5.02 Interactive utilities

- In Inspector, the spacing and spacing variation controls are deprecated and should have been grayed-out.

5.03 MIL add-on to Microsoft Visual Studio

- If both MIL 32-bits and 64-bits are installed on the same system but only one of the installation is updated using MIL 10 Processing Pack 1, the MIL add-on will nevertheless show up in both cases suggesting the most recent MIL flags.

5.04 MIL Help

- CHM based help files does not work properly under the following circumstances:
 - . The operating System is Windows 7 embedded
 - . The latest Windows updates, including IE11, are applied to the system
 - . The user is logged in as a standard user (i.e part of the Users group)
- A work around is to log in with administrator privileges.

 Section 5: Differences between MIL 10 and MIL 9 Processing Pack 2 with
 Update 56

Table of Contents for Section 5

- 1. Overview
- 2. New functionalities and improvements
 - 2.01 MIL processing specific examples
 - 2.02 Bead module
 - 2.03 Blob module
 - 2.04 Calibration module
 - 2.05 Code module
 - 2.06 Color module
 - 2.07 Edge Finder module
 - 2.08 Measurement module
 - 2.09 Metrology module
 - 2.10 Model Finder module
 - 2.11 OCR module
 - 2.12 Pattern matching module
 - 2.13 Registration module
 - 2.14 String Reader module
 - 2.15 3dmap module
 - 2.16 Graphic functions
 - 2.17 Regions
 - 2.18 Primitives
 - 2.19 Matrox Profiler
- 3. Deprecated functionalities
 - 3.01 General
 - 3.02 Bead module
 - 3.03 Calibration module
 - 3.04 Color module
 - 3.05 Measurement module
 - 3.06 Metrology module
 - 3.07 Model Finder
 - 3.08 Edge Finder
 - 3.09 Interactive utilities
- 4. Fixed bugs
 - 4.01 General
 - 4.02 Bead module
 - 4.03 Blob module
 - 4.04 Calibration module
 - 4.05 Code module
 - 4.06 Color module
 - 4.07 Edge Finder module
 - 4.08 Measurement module
 - 4.09 Metrology module
 - 4.10 Model Finder module
 - 4.11 OCR module
 - 4.12 Pattern matching module
 - 4.13 Registration module
 - 4.14 String Reader module
 - 4.15 3dmap module
 - 4.16 Graphic functions
 - 4.17 Regions
 - 4.18 Primitives
 - 4.19 Interactive utilities
- 5. Known limitations
 - 5.01 Graphic functions
 - 5.02 Interactive utilities
 - 5.03 Regions
 - 5.04 Metrology
 - 5.05 Matrox Profiler
 - 5.06 Streaming functions

 1. Overview

- MIL 10 includes all the features of MIL 9 Processing Pack 2.
 In addition, MIL 10 includes new processing functionalities, performance optimizations and general improvements.
- MIL 10 is the minimum requirement for all upcoming Processing Packs until the next major release.

2. New functionalities and improvements

- 2.01 MIL processing specific examples
 - The /Processing/ directory includes new MIL processing specific examples (see MIL Control Center -> Directories -> MIL examples): a multiprocessing framework, various product inspections, such as bottle cap and label verification, color analysis for food inspection of product identification, 3D object localization, etc., as well as additional 3D camera interfacing examples.
 - The MIL Examples have been updated and use the graphics list object to display the result with zoomable subpixel annotations.

- A new MIL example has been added which illustrates the benchmarking of processing operations.

2.02 Bead module

- New: Support to train and to validate stripe-type bead templates with non-uniform widths.
- New: M_BEAD_EDGE bead template type, to train and to validate the presence and position of an edge-type bead along a path.
- New: M_NUMBER_FOUND result type to retrieve the number of bead sections found for a given bead template.
- New: M_TRAINING_PATH with M_CIRCLE and associated M_TEMPLATE_CIRCLE... control types to parametrically define a circular bead template.
- New: M_TRAINING_PATH with M_SEGMENT and associated M_TEMPLATE_SEGMENT... control types to parametrically define a line segment bead template.
- New: M_INDIVIDUAL_WIDTH_NOMINAL to retrieve the user defined bead section width for a given bead template.
- New: M_TRAINED_SIZE, M_TRAINED_POSITION_X, M_TRAINED_POSITION_Y, and M_TRAINED_INDIVIDUAL_WIDTH_NOMINAL to retrieve the trained information for a given bead template.
- New: M_INTENSITY_NOMINAL_MODE control type to select whether to train and to validate the color of a stripe-type bead.
- New: M_INTENSITY_NOMINAL control type to define the expected nominal color of a given bead template.
- New: M_INTENSITY_DELTA_POS and M_INTENSITY_DELTA_NEG control types to set color tolerance of a given bead template.
- New: MbeadDraw M_FAIL_INTENSITY_COLOR_MIN and M_FAIL_INTENSITY_COLOR_MAX to display the failed bead sections regarding the color range criterion.
- New: M_ANGLE_ACCURACY_MAX_DEVIATION control to help improve the accuracy of the width measure when the bead section deviates from the expected path.
- New: MbeadTemplate M_SET_WIDTH_NOMINAL to set a specific width value for each user point in order to define bead templates with non-uniform width.
- New: MbeadControl M_WIDTH_NOMINAL_MODE can be set to M_AUTO_UNIFORM or M_AUTO_CONTINUOUS to either train a bead template with a uniform or with a non uniform width.
- New: MbeadGetResult M_CLOSURE to retrieve whether a template is closed.
- New: MbeadGetResult M_OFFSET_MAX to retrieve the offset value of the maximum offsetted bead section of a template.
- New: MbeadGetResult M_OFFSET_MAX_INDEX to retrieve the index of the maximum offsetted bead section of the template.
- New: MbeadGetResult M_STATUS_FOUND to retrieve the status to determine if at least one bead section has not been found for a template.
- New: MbeadGetResult M_STATUS_OFFSET to retrieve the status to determine if at least one bead section offset is out of the tolerance for a template.
- New: MbeadGetResult M_STATUS_WIDTH_MIN to retrieve the status to determine if at least one bead section width is out of the min tolerance for a template.
- New: MbeadGetResult M_STATUS_WIDTH_MAX to retrieve the status to determine if at least one bead section width is out of the max tolerance for a template.
- New: MbeadGetResult M_STATUS_INTENSITY_COLOR_MIN to retrieve the status to determine if at least one bead section color is out of the min tolerance for a template.
- New: MbeadGetResult M_STATUS_INTENSITY_COLOR_MAX to retrieve the status to determine if at least one bead section color is out of the max tolerance for a template.
- New: MbeadGetResult M_INTENSITY_MIN to retrieve the minimum bead section color value of a template.
- New: MbeadGetResult M_INTENSITY_MAX to retrieve the maximum bead section color value of a template.
- New: MbeadGetResult M_INTENSITY_MIN_INDEX to retrieve the index of the bead section with the lowest color value of a template.
- New: MbeadGetResult M_INTENSITY_MAX_INDEX to retrieve the index of the bead section with the highest color value of a template.
- New: MbeadGetResult M_INTENSITY to retrieve the bead section color values for a bead template.
- New: MbeadGetResult M_ANGLE to retrieve the bead section angle values.
- New: MbeadGetResult M_TRAINED_INDEX to retrieve the corresponding trained point index in the trained set of points.
- New: M_FAIL_WARNING_OFFSET control to increase the search box dimension to find offsetted bead sections beyond the valid offset range.

2.03 Blob module

- New: M_CONVEX_HULL_COG_X and M_CONVEX_HULL_COG_Y blob features.
- New: M_FERET_PERPENDICULAR_TO_MIN_DIAMETER and M_FERET_PERPENDICULAR_TO_MAX_DIAMETER blob features.
- New: M_FERET_MIN_DIAMETER_ELONGATION and M_FERET_MAX_DIAMETER_ELONGATION blob features.
- New: M_FERET_AT_PRINCIPAL_AXIS_ANGLE and M_FERET_AT_SECONDARY_AXIS_ANGLE blob features.
- New: M_ALL_FERETS to retrieve all the calculated ferets.
- New: M_FERET_PRINCIPAL_AXIS_ELONGATION blob feature.
- New: M_RECTANGULARITY blob feature.
- New: M_BLOB_TOUCHING_IMAGE_BORDERS blob feature.
- New: M_BLOB_INCLUSION_STATE result to retrieve whether a specific blob is included or excluded.
- New: M_TOTAL_NUMBER_OF_RUNS to retrieve the total number of runs in an image.
- Improvement: improved calculation responsiveness and accuracy when limiting the processing time using the M_TIMEOUT control.
- Improvement: better use of multiprocessing capabilities.

2.04 Calibration module

- New: McalFixture M_SAME_AS_SOURCE location type to move the relative

- coordinate system to the position of the relative coordinate system of the source.
- New: McalWarp function to define the mapping of a calibration context based on another calibration context and an optional warping.
- New: McalTransformCoordinate3dList, McalTransformCoordinate, and McalTransformResultAtPosition M_NO_EXTRAPOLATED_POINTS mode to return M_INVALID_POINT for pixels transformed outside a piecewise linear calibrated area.
- New: McalTransformCoordinate3dList now supports 2d calibration contexts and depth-map buffers through the M_DEPTH_MAP mode.
- New: McalGrid and McalList now work using M_UNIFORM_TRANSFORMATION calibration contexts.
- Improvement: McalGrid is now more robust using M_CHESSBOARD_GRID grid type.
- Improvement: M_DISPLACE_RELATIVE_COORD now supports a minimum of 4 points for the computation to succeed.
- New: McalFixture supports M_RESULT_MET as a location type to fixture from from the global or a local frame result.

2.05 Code module

- New: M_SIMPLIFIED_CHINESE, M_JAPANESE, M_LATIN, M_KOREAN for M_STRING_FORMAT.
- Improvement: reading PDF417, TruncatedPDF417 codes at low and very low speed is more robust in the presence of local deformation or damaged pattern corners.
- Improvement: reading M_ANY foreground MicroPDF417 is more robust.
- Improvement: reading 4-State codes in the presence of extra bars and/or uneven spacing is more robust.
- New: M_SEARCH_ANGLE_STEP control used for reading 1D barcodes.
- Improvement: M_TYPE_MIL_ID combination flag is now supported by McodeInquire.

2.06 Color module

- New: M_HISTOGRAM_VOTE matching technique based on the histogram of the image.
- New: McolDefine(+M_ADD_COLOR_TO_SAMPLE) to facilitate the definition of a color sample from multiple color patches.
- New: M_DRAW_SAMPLE_MOSAIC and M_DRAW_SAMPLE_MOSAIC_DONT_CARE drawing operations for samples defined from multiple color patches.

2.07 Edge Finder module

- Improvement: building the edge chains is more efficient in speed and memory usage.

2.08 Measurement module

- New: MmeasFindMarker now accepts buffers with regions (MbufSetRegion) to define the search region.
- New: M_EDGEVALUE_VAR_MIN control to set the minimum expected prominence of the edge peaks.
- New: M_EDGE_CONTRAST result defined by the difference between the intensities measured at the edge's maximum of curvature positions.
- New: M_EDGE_CONTRAST_SCORE based on the M_EDGE_CONTRAST result value.
- New: MmeasSetMarker M_CIRCLE_INSIDE_SEARCH_REGION control allows to find a circle lying partially outside the search region.
- New: M_FIT_ERROR_MAX result is now available for circle markers.
- New: MmeasSetMarker M_CIRCLE_ACCURACY control allows the use of a low accuracy but typically faster find algorithm.
- New: MmeasSetMarker M_SUBPIXEL_MODE control offers two modes (M_LOCAL and M_GLOBAL) to determine the subpixel location.
- New: MmeasSetMarker M_SEARCH_REGION_CLIPPING... controls to select the behavior of the clipping of the search region.
- New: M_EDGE_WIDTH result based on maximal curvature locations, contrary to M_EDGEVALUE_PEAK_WIDTH, which is based on the threshold. (see also new M_EDGE_START, M_EDGE_END results).
- New: M_DRAW_EDGEVALUE_PEAK_WIDTH_IN_PROFILE to draw the width of the measured marker along the edge profile annotation (edge and stripe markers).
- New: M_DRAW_EDGEVALUE_MIN_IN_PROFILE to draw the threshold levels along the edge profile annotation (edge and stripe markers).
- New: MmeasDraw with M_RESULT_PER_SUBREGION() macro allows various drawing operations for annotating the subregions (edge and stripe markers).
- New: MmeasDraw with M_DRAW_POSITION_IN_PROFILE to draw the position of the edge along the edge profile annotation (edge and stripe markers).
- Improvement: MIL_TYPE_MIL_ID combination flag is now supported by MmeasInquire.
- New: M_DRAW_INCLUSION_POINT to draw the location of M_INCLUSION_POINT.
- Improvement: improved accuracy of the returned angle when measuring in a calibrated source image.

2.09 Metrology module

- New: M_DERIVED_GEOMETRY_REGION object and its associated M_REGION... controls to define regions from other features (e.g. a ring defined from three points).
- New: MmetSetRegion(...M_FROM_DERIVED_GEOMETRY_REGION...) to define the region of a feature from an M_DERIVED_GEOMETRY_REGION object.
- New: MmetSetRegion M_FROM_GRAPHIC_LIST to define a region from a graphic list object.
- New: MmetDraw M_DRAW_FITTED_EDGELS to draw the edgels participating in the final fit of a feature.
- New: MmetControl M_EDGEL_TYPE to construct an M_EDGEL feature either the active edgels of a feature, or from the fitted edgels of a feature.
- New: MmetControl M_GAIN and M_OFFSET to correct for tolerance bias before validating the result.
- New: MmetInquire M_INDEX_FROM_LABEL to retrieve the index of a feature

- or tolerance.
- New: MmetGetResult M_LABEL_VALUE to retrieve the labels of features and tolerances.
- New: M_COVERAGE values of fitted features can be retrieved.
- New: MmetControl M_REGION_ACCURACY_HIGH can be disabled to allow using faster but lower accuracy region mapping.
- Improvement: M_EDGEL_SELECTION_RANK now accepts M_DISABLE, which is the default behavior.
- Improvement: MmetCalculate has additional optimizations when calculating measured features from rectangular, ring, ring sector, or 1D regions.
- Improvement: has been optimized for multi-core (MP) architectures.

2.10 Model Finder module

- New: MmodControl M_ACTIVE_EDGELS to control the percentage of the most relevant edgels from the target image that will be used during the find process, for M_GEOMETRIC contexts.
- New: MmodDraw M_DRAW_ACTIVE_EDGELS to visualize selected edgels from the target image used during the find process.
- New: M_SEARCH_POSITION_FROM_GRAPHIC_LIST control allows defining the search region from a graphics list object.

2.11 OCR module

- New: MocrReadString, MocrVerifyString and MocrCalibrationFont functions now accept buffers with an associated rotated rectangular region (see MbufSetRegion and MocrControl(M_STRING_ANGLE, M_ACCORDING_TO_REGION))

2.12 Pattern matching module

- New: MpatFindModel and MpatFindModelMultiple now accept buffers with an associated rotated rectangular region (see MbufSetRegion).

2.13 Registration module

- Improvement: MregTransformCoordinate error message has been clarified when the result is not calculated.

2.14 String Reader module

- New: The '-' character is now correctly considered as a punctuation character when added to a font definition.
- New: MstrControl now supports M_RESULT_OUTPUT_UNITS to retrieve result values in world units. Note that the default setting for the result output unit is M_ACCORDING_TO_CALIBRATION. Existing calibrated applications may need to explicitly change this setting.
- Improvement: MstrRead is more robust with strings and fonts containing punctuation characters.

2.15 3dmap module

- New: "3D Setup Helper.xls" Microsoft Excel sheet, located in the "Matrox Imaging/Tools/" directory, to help determine the adequate hardware configuration for a camera-laser setup.
- New: M_Y_THEN_X control to select an alternative gap filling direction.
- New: M_FILL_THRESHOLD_X and M_FILL_THRESHOLD_Y controls to adjust the gap filling decision of a depth map.
- New: M3dmapDraw M_DRAW_CALIBRATION_PEAKS to draw all the extracted laser lines during calibration operation.

2.16 Graphic functions

- New: modules can now draw (MmodDraw, MmetDraw...) into a graphics list object to display the result with zoomable subpixel annotations.
- New: it is now possible to interactively edit a graphics list (see MdispControl with M_GRAPHIC_LIST_INTERACTIVE and MgraInteractive).
- New: MgraArcAngle M_CONTOUR and M_SECTOR control flag values to specify an arc style.
- New: MgraControlList M_ARC_STYLE to control the style of an arc in a graphics list.
- New: MgraControlList M_GRAPHIC_SELECTED control flag to set whether a given graphic is selected or not.
- New: MgraControl and MgraControlList with M_GRAPHIC_SOURCE_CALIBRATION and M_FIXTURE to allow drawing annotations under different fixtures.
- New: MgraControl, MgraInquire, MgraControlList, and MgraInquireList, new control flags for managing interactivity: M_SPECIFIC_FEATURES_EDITABLE, M_RESIZABLE, M_ROTATABLE, M_MOVABLE, M_EDITABLE, M_SELECTABLE, M_VISIBLE, M_INTERACTIVE_ANNOTATIONS_COLOR, M_SELECTED_COLOR, M_SELECTION_RADIUS.
- New: MgraControl new control flags to draw with a zoom and/or an offset: M_DRAW_OFFSET_X, M_DRAW_OFFSET_Y, M_DRAW_ZOOM_X, M_DRAW_ZOOM_Y.
- New: MgraCopy function to copy one or more graphics from a graphics list to another.
- New: MgraHookFunction to attach or detach a user function to a graphics list event such as a modification to the graphics list.
- New: MgraGetHookInfo function to retrieve the information about a graphics list event.
- New: MgraControlList M_GRAPHIC_CONVERSION_MODE control to determine how to perform the conversion between world and pixel units of a graphic. See M_RESHAPE_FROM_POINTS, M_RESHAPE_FOLLOWING_DISTORTION and M_PRESERVE_SHAPE_AVERAGE.
- New: MgraLines M_INFINITE_LINES control flag to draw infinite lines from an array of pairs of points.
- New: MgraDraw M_NO_INTERACTIVE_ANNOTATION control flag to disable drawing interactive annotations.
- New: MgraInquireList M_INTERACTIVE_GRAPHIC_STATE to retrieve the current state of the interactivity of a graphics list (e.g. IDLE, hover handle)
- New: MgraControlList M_CONSTRAIN_ASPECT_RATIO control flag to set whether to force the width of objects equal to the height, or not.
- New: MgraControlList M_MULTIPLE_SELECTION control flag to set whether to enable interactive selection of multiple graphics using the Ctrl key,

or not.

2.17 Regions

- New: MbufSetRegion M_FILL_REGION combination flag to interpret all graphics list elements as filled graphics.

2.18 Primitives

- New: MbufClearCond function to clear the pixels satisfying a specified condition.
- New: MimDilate and MimErode M_BINARY_ULTIMATE mode to perform ultimate morphological dilate and erosion.
- New: MimArith operations: M_EXP, M_CONST_EXP, M_LOG, M_LOG_CONST and M_CONST_LOG.
- New: MimArith allows many existing operations to be performed interpreting pixels as logical boolean values (see M_LOGICAL).
- New: MimArith can now perform operations with constant operand in floating-point accuracy using the M_FLOAT_PROC combination flag.
- New: M_POWER function in MgenLutFunction ($A \cdot X^B + C$).
- New: the following context types can be streamed with MimSave, MimRestore, and MimStream: M_DEAD_PIXEL_CONTEXT, M_FLAT_FIELD_CONTEXT, M_MATCH_CONTEXT, M_REARRANGE_CONTEXT and M_STAT_MULTIPLE_CONTEXT.
- New: buffers allocated with the M_LUT attribute now support calls to MbufControlNeighborhood so that overscan parameters can be set.
- New: MimClip now has the M_SATURATION condition to allow clipping to be done according to the minimum and maximum of the destination buffer properties.
- New: MimLocateEvent can now receive M_ALL as a condition to consider all pixels.
- New: MimLocateEvent has new combination constants: M_LOCAL_MAX_NOT_STRICT, M_LOCAL_MAX_STRICT_MEDIUM, M_LOCAL_MIN_NOT_STRICT and M_LOCAL_MIN_STRICT_MEDIUM.
- New: MimConnectMap now supports white overscan. See MbufControlNeighborhood M_OVERSCAN_REPLACE_VALUE control set to M_REPLACE_MAX on the received LUT.
- Improvement: the accuracy of the optimized version of MbufBayer with M_AVERAGE_2X2 interpolation has been improved.
- Improvement: MbufBayer with M_WHITE_BALANCE_CALCULATE now normalizes coefficients with the maximal average value from the r, g, b bands.
- Improvement: MbufBayer white balancing coefficient calculation is improved.
- Improvement: the robustness of MimBinarize(...M_TRIANGLE_BISECTION...) to small changes in the histogram of the image has been improved.
- Improvement: MimCountDifference is now optimized for SSE2 CPU and for multiprocessing as well.

2.19 Matrox Profiler

- New: Matrox Profiler tool allows to graphically and interactively analyze the MIL traces including the MIL functions, the MIL events, the MIL errors, etc...

3. Deprecated functionalities

3.01 General

- Several MIL constant values (e.g. M_TYPE_CHAR, M_TYPE_SHORT, M_TYPE_MIL_INT, M_TYPE_MIL_INT32, M_TYPE_MIL_INT64, M_TYPE_MIL_FLOAT, M_TYPE_DOUBLE, M_TYPE_MIL_ID, M_TYPE_MIL_TEXT, M_AVAILABLE, and M_SUPPORTED) now require a MIL_INT64 variable to store it, if needed.
- Queuing results by calling several M_GetResult(), M_GetResultID(),... with an M_NULL result pointer is now deprecated.
- M_DRAW_RELATIVE_ORIGIN_X, M_DRAW_RELATIVE_ORIGIN_Y, M_DRAW_SCALE_X and M_DRAW_SCALE_Y controls are now deprecated. They are replaced by MgraControl on the graphic context with M_DRAW_OFFSET_X, M_DRAW_OFFSET_Y, M_DRAW_ZOOM_X and M_DRAW_ZOOM_Y.
- MbufControlRegion is deprecated and replaced by MbufControlArea to remove the ambiguity with the region (MbufSetRegion) functionalities.

3.02 Bead module

- M_POSITION_MODE control is deprecated and replaced by M_TRAINING_PATH.
- M_USER_DEFINED position mode is deprecated and replaced by M_POLYLINE training path.
- M_AUTO position mode is deprecated and replaced by M_POLYLINE_SEED training path.

3.03 Calibration module

- McalTransformResult M_ALLOW_INVALID_POINT_OUTPUT combination flag is deprecated. M_INVALID_POINT is returned when the value cannot be transformed.
- McalTransformCoordinate, McalTransformCoordinateList, McalTransformCoordinate3dList, and McalTransformResultAtPosition ; M_ALLOW_INVALID_POINT_OUTPUT combination flag is deprecated and replaced by M_NO_POINTS_BEHIND_CAMERA. Note that even if not using M_NO_POINTS_BEHIND_CAMERA, the function may return M_INVALID_POINT when transforming pixels on the horizon line.
- McalTransformCoordinate3dList M_PLANE_INTERSECTION mode flag is deprecated ; use M_DEFAULT instead.

3.04 Color module

- M_CIEDE94_GRAPHIC_ARTS is deprecated and replaced by M_CIE94_GRAPHIC_ARTS.
- M_CIEDE94_TEXTILE is deprecated and replaced by M_CIE94_TEXTILE.

3.05 Measurement module

- M_POSITION_MIN edge result type is deprecated and replaced by M_EDGEVALUE_PEAK_POS_MIN.
- M_POSITION_MAX edge result type is deprecated and replaced

- by M_EDGEVALUE_PEAK_POS_MAX.
- M_CONTRAST edge result type is deprecated and replaced by M_EDGEVALUE_PEAK_CONTRAST.
- M_CONTRAST_SCORE edge score property is deprecated and replaced by M_EDGEVALUE_PEAK_CONTRAST_SCORE.
- M_WIDTH edge result type is deprecated and replaced by M_EDGEVALUE_PEAK_WIDTH.
- M_WIDTH stripe result type is deprecated and replaced by M_STRIPE_WIDTH.
- M_WIDTH_SCORE stripe score characteristic is deprecated and replaced by M_STRIPE_WIDTH_SCORE.
- The value M_ENABLE for M_SEARCH_REGION_CLIPPING in MmeasSetMarker is deprecated and replaced by M_MAXIMIZE_AREA.
- M_EDGE_THRESHOLD control is deprecated and replaced by M_EDGEVALUE_MIN.
- The M_ALL parameter index in MmeasDraw is deprecated and replaced by M_RESULT_ALL_OCCURRENCES.
- The deprecated M_WEIGHT_FACTOR is now obsolete and will lead to a MIL error. MmeasSetScore must be used instead.
- M_DRAW_ARROW draw type is deprecated and replaced by M_DRAW_SEARCH_DIRECTION.

3.06 Metrology module

- MmetInquire M_METHOD control flag is deprecated and replaced by M_OPERATION control flag.
- M_CONSTRUCTION_FEATURE_LABEL is deprecated and replaced by M_BASE_FEATURE_LABEL.
- M_CONSTRUCTION_FEATURE_INDEX is deprecated and replaced by M_BASE_SUBFEATURE_INDEX.
- M_NUMBER_OF_CONSTRUCTION_FEATURE_LABEL is deprecated and replaced by M_BASE_SUBFEATURES_ARRAY_SIZE.
- M_NUMBER_OF_CONSTRUCTION_FEATURE_INDEX is deprecated and replaced by M_BASE_FEATURES_ARRAY_SIZE.
- M_FILTER_MODE, M_FILTER_WIDTH and M_KERNEL_DEPTH are deprecated.
- M_VISIBLE is deprecated and replaced by M_DRAWABLE.

3.07 Model Finder

- M_FILTER_MODE, M_FILTER_WIDTH and M_KERNEL_DEPTH are deprecated.

3.08 Edge Finder

- M_FILTER_MODE, M_FILTER_WIDTH and M_KERNEL_DEPTH are deprecated.

3.09 Interactive utilities

- All M_INTERACTIVE dialogs are deprecated, except for Save, Restore and Stream operations.

4. Fixed bugs

4.01 General

- MdigProcess did not propagate its associated calibration to the grabbed images.
- The M_PROC buffer attribute was not required by all processing functions including: MedgeMask, MedgeDraw, MblobDraw, MblobFill, MblobLabel, MmeasDraw, MmeasFindMarker, MmetDraw, MocrDraw, MregDraw, MregControl, MstrDraw, MstrExpert, McodeDraw, MpatDraw, MpatAllocModel, MpatCopy, MpatSetDontCare, McolDefine, McolDistance, McolDraw, McolMask, McolMatch, McolProject, M3dmapAddScan, M3dmapArith, M3dmapDraw, M3dmapExtract, M3dmapSetGeometry, M3dmapStat, MbeadDraw, MbeadTrain, MbeadVerify, MmodControl, MmodDefine, MmodDraw, MmodMask, and MmodFind.

4.02 Bead module

- MbeadTrain, MbeadVerify and MbeadTemplate functions now validates that only unsigned 8-bit mono buffers are supported.
- MbeadGetResult did not work properly on remote systems when using DMIL.
- MbeadVerify will no longer lead, under some circumstances, to an exception with an image having no bead.
- MbeadDraw will no longer draw the M_DRAW_SEARCH_BOX centered on the found position instead of the trained position.

4.03 Blob module

- MblobDraw(...M_DRAW_CONVEX_HULL...) will no longer crash when the necessary features were not previously selected with MblobSelectFeature().
- MblobGetRuns will no longer crash when using the M_LABELED identification mode and calculating the convex hull.
- MblobCalculate wrongly accepted binary buffers as the grayscale image, leading to erroneous results.
- M_RETURN_PARTIAL_RESULTS will no longer return invalid results for some features such as the general centered moments.
- Inquiring the M_RETURN_PARTIAL_RESULTS will no longer return M_YES/M_NO but M_ENBALE/M_DISABL as expected.

4.04 Calibration module

- McalTransformCoordinate was not thread-safe.
- McalGrid will no longer fail, in some special cases, when the grid had an orientation near 45-degrees combined with very high perspective deformation.
- McalGrid will no longer find wrong positions with M_CHESSBOARD_GRID when the image is highly distorted.
- McalList did an incorrect parameter check for the minimum number of 3D points needed; it always required 6 points.
- McalFixture was not taking into account the relative coordinate system when using a calibration context as a reference (instead of an image).

4.05 Code module

- Mix of parity patterns were not properly validated for EAN-13.

- McodeDraw will no longer alter the destination buffer associated calibration, if any.
- PDF417 and TruncatedPDF417 error correction will no longer fail in rare situations.
- PDF417 and TruncatedPDF417 accepted impossible combinations of the numbers of cells in X and Y with $X*Y > 928$, which is greater than the maximum number of codewords allowed.
- PDF417 and TruncatedPDF417 will no longer detect the wrong number of columns or rows for some specific codes.
- Size of the 2D part of a composite code will no longer be incorrectly estimated.
- Under rare circumstances the wrong string will no longer be read with MicroPDF417: codewords 916 and 917 will no longer be incorrectly handled.
- Stability of the AIM DPM verification has been fixed for Data Matrix.
- Reading a code in a region outside the target image did not refresh the M_STATUS of the results.
- McodeRead will no longer fail decoding a degraded QRCode containing the maximum error correction capacity.
- McodeRead will no longer potentially crash when reading 2-State and 4-State codes.

4.06 Color module

- McolDistance normalization was not working properly for M_BGR32 buffers.
- McolMatch with M_HISTOGRAM_MATCHING directly on a sample image will no longer lead to an incorrect M_SCORE_RELEVANCE value over 100%.
- McolInquireSafeType (M_MIL_USE_SAFE_TYPE) will no longer lead to false errors in rare circumstances.
- McolSetMethod wrongly generated an error when using M_HISTOGRAM_MATCHING with the M_MANHATTAN distance and M_CIELAB color space.

4.07 Edge Finder module

- MedgeGetNeighbors with M_GET_EDGELS will no longer lead to an exception under some specific circumstances.

4.08 Measurement module

- MmeasSetMarker(...M_RING_RADII..) will no longer lead to an inner radius greater than the outer radius when the outer radius was set to M_DEFAULT.
- MmeasDraw(...M_DRAW_SEARCH_REGION..) did not draw anything when the M_BOX_ANGLE control was set to M_ANY. It now draws the box at 0 degree.
- MmeasDraw(...M_DRAW_SEARCH_REGION...) did not draw anything when M_BOX_SIZE was set to M_DEFAULT.
- MmeasFindMarker stability with circle and stripe markers has been fixed.
- The default search region, defining the whole image, will longer results in potential unstable calculations.
- The default search region for circle markers was not perfectly at the image subpixel center.
- MmeasInquire used with the M_SUPPORTED combination flag will no longer lead to a MIL error.
- Calling MmeasGetResult before MmeasFindMarker will no longer cause an exception.
- M_CONTRAST result will no longer give negative values when using an M_SHEN or M_PREWITT filter types.
- MmeasFindMarker will no longer find circles whose centers are not inside the search region.
- MmeasFindMarker will no longer falsely report a "region out of image" error message.
- MmeasFindMarker returned inaccurate subedges results if using an aspect ratio.
- The subpixel location of an edge will no longer be biased in the presence of very close consecutive edge transitions.
- When using multiple sub boxes for edge and stripe markers, the data of the last sub box will no longer be partially projected.
- M_SUB_EDGES_POSITION and M_SUB_EDGES_WEIGHT results will now be ordered similarly to M_SUB_EDGES_MARKER_INDEX results.
- A specific sequence of MeasSetScore/MmeasFindMarker/MmeasSetScore/MmeasGetScore calls will no longer result in a crash.
- MmeasDraw with M_DRAW_BOX no longer leads to an incorrect drawing when some box settings are set to M_DEFAULT.
- All edge and stripe marker annotations (crosses) are all consistently drawn at the angle of the measurement box.
- MmeasStream will no longer generate a small memory leak.
- The M_INCLUSION_POINT will no longer ignore the input unit setting.

4.09 Metrology module

- When drawing in a calibrated destination, the Y-axis of the local frames was not aligned with the world Y-axis, but orthogonal to the world aligned X axis.
- MmetCalculate will no longer skip calculations if only the calibration information of an image changed and not its content.
- MmetDraw with M_ACTIVE_EDGELS will no longer draw the fitted edgels instead of the selected ones.
- MmetDraw, horizontal and vertical constructed parametric lines could not be drawn.
- The arrow of an arc region was not drawn properly when the start angle was greater than the end angle.
- MmetDraw of edgels did not properly consider the calibration of the image used by MmetCalculate.
- MmetDraw will no longer draw unsuccessful calculated features at (0, 0). It now draws nothing.
- MmetDraw will no longer ignore the calibration information of the destination image when drawing tolerances.
- The farthest point (M_MAX_DISTANCE_POINT) between a circle and an arc will no longer give the wrong point.

- MmetGetResult will no longer return the wrong pixel unit coordinates for measured points when calculating on a calibrated image.
- M_RESULT_OUTPUT_FRAME set to M_GLOBAL_FRAME did not return the correct results.
- Perpendicularity and parallelism tolerances were not always returning the smallest angular values.
- The construction of a perpendicular line, a parallel line, or a line at angle was not properly validating the type of the reference feature. An error message will now pop when the construction is invalid.
- MmetCalculate will no longer find edgels outside the measured region when using a calibration with a very distorted pixel to world mapping.

4.10 Model Finder module

- MmodFind will no longer generate a run-time exception in very rare configuration with very short extracted edge chains.
- Under some circumstances, MmodFind could not find a synthetic rectangular model if M_ACCURACY was set to M_LOW.
- MmodRestore(... M_WITH_CALIBRATION ...) will no longer lead to an error in MmodPreprocess.
- Robustness fixes with M_GEOMETRIC_CONTROLLED contexts when using the M_LAST_LEVEL control.
- MmodDraw M_DRAW_MODEL_IMAGE and M_DRAW_EDGES for synthetic and CAD models did not take into account the graphic context color setting, but forced the color to 255.

4.11 OCR module

- M_CHAR_SPACING result returned incorrect values when the read string was at an angle.
- MocrDraw ignored the color of the received graphic context.
- MocrGetResult with M_CHAR_SPACING, M_CHAR_SIZE_X/Y, M_STRING_ANGLE was not returning results in world units even if the read was performed with a calibrated image.
- MocrPreprocess will no longer lead to an exception if M_TARGET_CHAR_SPACING is set lower than 2.
- MocrReadString will no longer give unstable results when there are fewer characters in the image than the number of characters to read.
- MocrReadString was not working properly when reading characters at angle with a different scale than the font.
- MocrReadString will no longer lead to an exception when M_STRING_ANGLE_DELTA_NEG or M_STRING_ANGLE_DELTA_POS are set to 2.
- Predefined SEMI font contexts will no longer read with a bad checksum if the number of strings is greater than one.

4.12 Pattern matching module

- MpatFindModel was not working perfectly with very large buffers (one pixel size over 65535).
- MpatFindModel with circular overscan model will no longer find multiple occurrences at the same position with multiprocessing enabled.
- MpatFindModel will no longer have a memory leak with multiprocessing enabled.
- MpatFindModel is now able to find a model if the occurrence position is too close to the search region borders.
- M_MIN_SPACING_X, M_MIN_SPACING_Y were not working properly when searching for a model at an angle.
- M_SAVE_SUMS set to M_ENABLE will no longer lead to incorrect results when the center of the model is also changed.
- MpatFindModel will no longer crash when setting an M_MODEL_STEP to an unexpected value.
- MpatFindModel is now able to find large models with don't care mask and model step of 2.
- In rare cases, MpatFindModel will no longer crash when using very large models.

4.13 Registration module

- MregTransformImage erroneously accepted destination buffer formats other than 8+M_UNSIGNED.

4.14 String Reader module

- For MstrRead and MstrExpert, a lack of memory will no longer lead to a memory leak in very special circumstances.
- MstrRead will no longer be unable to read in an image with an associated region and with the M_THICKEN_CHAR control set to a non-null value.
- MstrRead will no longer potentially crash or lead to incorrect results when disabling some characters from a fontless context.

4.15 3dmap module

- M3dmapExtract M_FILL_MISSING_DATA_ONLY was not allowed on empty results.
- M3dmapExtract with M_USE_DESTINATION_SCALES now works on physically corrected images with a pixel rotation (fixtured buffers).
- M3dmapTriangulate now works with DMIL remote systems.
- M3dmapArith did not work properly on two source images with different calibration child offsets. It now reports an error.
- M3dmapExtract with M_USE_DESTINATION_SCALES did not take into account the calibration child offsets when extracting from a child of the 3d-corrected buffer.
- M3dmapExtract with M_USE_DESTINATION_SCALES did not work properly with a physically corrected image destination having a rotated relative coordinate system (fixture).
- M3dmapCalibrate in M_CALIBRATED_CAMERA_LINEAR_MOTION mode will no longer ignore the M_Y_AXIS_UP setting of the camera calibration.
- M3dmapExtract will no longer ignore to set the relative coordinate system. The relative coordinate system is now correctly set to the 3d reconstruction coordinate system.
- M3dmapArith will no longer ignore to set the relative coordinate system.

The relative coordinate system is now correctly set to the 3d reconstruction coordinate system.

4.16 Graphic functions

- MgraLines did not always render the exact same line when executed on child buffers versus when executed on its ancestor buffer.
- MgraArcFill will no longer crash for very large arcs.
- MgraInquireList(...M_LAST_LABEL...) will no longer erroneously return a zero label value.
- MgraClear will no longer ignore the colors from the received graphic context with on-board allocated buffers.
- Mgra functions execution will no longer be abnormally slow on GPU systems.
- MgraControlList on M_ALL or M_ALL_SELECTED objects will no longer report an error when the control does not apply to some graphic objects.
- MgraControlList did not correctly handle the M_AVAILABLE, M_SUPPORTED, and M_DEFAULT combination flags.
- MgraText will now correctly interpret the M_FONT_SIZE setting.
- After deleting the last graphic of a graphic list, inquire to M_LAST_LABEL will no longer return an invalid label value but M_NO_LABEL instead.

4.17 Regions

- MbufSetRegion function will no longer lead to an exception if function calls to the calibration module are not previously done.

4.18 Primitives

- Loading or streaming an image with no associated region into a buffer was deleting its region, if any. The region now remains intact. MbufSetRegion(...M_DELETE...) must be used to remove a region.
- MbufBayer applied the gamma correction after the demosaicing operation.
- MbufBayer with M_AVERAGE_2X2 will no longer lead to an exception with very small buffers.
- MimMorphic with M_BINARY processing mode used uninitialized overscan data when the destination was an 8-bits buffer.
- MimMorphic with 0 iterations had inconsistent behaviors. It now copies the source to the destination.
- MimMorphic performed all the iterations except the first one in the data type of destination. All iterations are now done and saturated according to the data type of the source buffer, then the result is put to the destination buffer.
- MimMorphic was truncating intermediate results when either the source, the destination or the structuring element was a floating-point buffer.
- MimThin and MimThick with 0 iterations had inconsistent behaviors regarding MimMorphic. It now copies the source to the destination.
- MbufCopyColor2d was allowed for subsampled bands (e.g. U and V bands for subsampled YUV formats). It is now protected.
- MimTransform with M_FFT will no longer give inexact results with some images when doing a reverse transform using only the 'real' part of the results.
- MimDraw is now working properly on 3-band color buffers.
- MimDraw now correctly considers the color set by the M_DEFAULT graphic context.
- MimDraw with M_DRAW_PEAKS will no longer give annotations off by half a pixel.
- MimDilate, MimErode, MimThin and MimThick functions always perform a buffer copy if the number of iterations is set to 0.
- MimPolarTransform will no longer lead to an exception when using a small angle.
- MimConvert will no longer give erroneous results with M_BGR32+M_PACKED image buffers when using matrix multiplication with a 4x3 matrix.
- MimLocateEvent will no longer lead to an exception when using multiprocessing optimization.
- MimMatch will no longer lead to an exception when using very large models.
- MimPolarTransform will no longer have inconsistent results when doing a transform and its reverse.
- Copying a YUV source image to an RGB destination with a type equal to or greater than 16 bits no longer leads to erroneous values.
- MimPolarTransform will no longer leads to erroneous values when the buffer is a child band of a non-regular buffer.
- MimThin with M_BINARY2 and a white overscan no longer occasionally leads to a crash.
- Under Windows XP, MappInquireMP(M_MEMORY_BANK_AFFINITY_MASK) and MbufAllocColor will no longer be inconsistent.
- MbufExport(...M_WITH_CALIBRATION...) will no longer malfunction with compressed buffers.

4.19 Interactive utilities

- Code Reader interactive utility will no longer prevent you from properly editing the number of cells in X with PDF417 and MicroPDF418 codes.
- Code Reader interactive utility will no longer lead to an exception when reading many code symbols.
- OCR Reader interactive utility did not report read, verify of preprocessing errors.
- OCR Reader interactive utility did give the focus to an inaccurate window when selecting a result.
- Color Analysis interactive utility will no longer display some artifacts when zooming the display of a color sample.
- Model Finder interactive utility did not correctly update the preprocess status when modifying the pixel scale of a synthetic model.
- Metrology interactive utility will no longer lead to an exception when opening the 'modify' dialog on a tolerance defined with a single feature.
- Metrology was not correctly displaying a reference template images when using 32+M_FLOAT format buffer.
- The calibration dialog in the interactive utilities erroneously allowed associating an image document to an empty calibration, leading

- to MIL errors.
- Using an interactive utility with a remote Iris system (DMIL) and with no local license prevented the deinterlace and the calibration dialogs from working properly.
- Pasting an image in an interactive utility gave an inaccurate error message when no license was present.
- Stability fixes in the continuous grab functionality.
- Stability fixes in the calibration dialog when using the 'Associate to digitizer' option.
- Opening the MIL help will no longer lead to an incorrect section if MIL help was already opened.

5. Known limitations

5.01 Graphic functions

- Interactive edition only tracks the region distortion, but not the drawing. The region drawing can track the region distortion when the M_CONVERSION_MODE is set to M_RESHAPE_FOLLOWING_DISTORSION.

5.02 Interactive utilities

- The MIL Interactive utilities (e.g. Matrox Inspector) ignore regions of images.
- M_DRAW_OFFSET_X/Y and M_DRAW_ZOOM_X/Y controls are currently not used by the interactive edition nor by the processing modules when using M_VECTOR type regions.
- Calibration dialog of the specific MIL Interactive utilities (e.g. Model Finder GUI): if a child buffer is selected, the calibration is performed using the child, but associated to the parent. This leads to incoherent results. Selecting a child should be avoided when performing a calibration using a specific MIL Interactive utilities.
- Inspector Measurement dialog has been updated to prevent from using the old deprecated scores.

5.03 Regions

- The raster version of a MIL 10 image's region is discarded when the image is loaded in an older version of MIL (i.e. MIL 9 PP2 and older). Forcing MbufSetRegion(...M_RASTERIZE...) after loading the image will restore the raster version.

5.04 Metrology

- Contexts having features or tolerances with an associated name (MmetSetName) and saved under Windows, cannot be restored on Linux; and vice-versa. Removing the names solves the issue.

5.05 Matrox Profiler

- Combining application and thread level activation or deactivation of the trace log can lead to incoherent results when visualized with the Matrox Profiler.

5.06 Streaming functions

- Loading a stream (file or memory) containing a different object type when using MxxxStream with an M_LOAD operation will not generate a MIL error as it should. The loaded object could be in an unstable state, particularly on a remote DMIL system.

Section 6: Differences between MIL 9 Processing Pack 2 with Update 56 and MIL 9 Processing Pack 2

Table of Contents for Section 6

1. Overview
2. Fixed bugs
 - 2.1 Pattern Matching

1. Overview

- MIL 9 Update 56 includes important bug fixes to the MIL Pattern Matching module.
- MIL 9 Update 56 is an update to MIL 9 with Processing Pack 2 only.

2. Fixed bugs

2.1 Pattern Matching

- MpatFindModel: a memory leak, when using circular overscan with MIL MP (multi-core processing) enabled, has been fixed.
- MpatFindModel: could return multiple occurrences at the same location when using circular overscan with MIL MP (multi-core processing) enabled.
- After setting the advanced M_MAX_INITIAL_PEAKS parameters, the number of occurrences returned by the module was always 1.

Section 7: Differences between MIL 9 Processing Pack 2 with Update 45 and MIL 9 Processing Pack 2

Table of Contents for Section 7

1. Overview
2. New functionalities and improvements

- 2.1 Code reading
- 2.2 Code verification
- 3. Fixed bugs
 - 3.1 Code reading
 - 3.2 Code writing
 - 3.3 Code verification
- 4. Known issues

1. Overview

- MIL 9 Update 45 includes new functionalities, performance optimizations and general improvements for the MIL Code Reader module.
The key features are:
 - * MicroQR code support;
 - * More distortion support for Data Matrix, QR and 4-State codes.
 - * Verification of Data Matrix, QR and MicroQR codes using ISO 15415 and AIM DPM-1-2006 standards;
- MIL 9 Update 45 is an update to MIL 9 Processing Pack 2 only.
- MIL 9 Update 45 updates the MIL Help files accordingly.

2. New functionalities and improvements

2.1 Code reading

- New support for reading Micro-QR codes
- QR codes: M_DISTORTION control type supports M_UNEVEN_GRID_STEP to read symbols with a moderate amount of distortion between alignment patterns.
- QR codes: M_FOREGROUND_VALUE control type supports M_FOREGROUND_ANY.
- QR codes: supports character sets as defined in ISO/IEC 18004:2006.
- Data Matrix: M_DISTORTION control type supports the new flag M_PERSPECTIVE_UNEVEN_GRID_STEP to read symbols distorted by a combination of uneven grid steps and an affine transformation e.g. (perspective, shearing, etc.)
- MicroQR codes: supports character sets as defined in ISO/IEC 18004:2006.
- Data Matrix: improved localization performance especially for rotated codes.
- 4-State codes: improved robustness for severely wavy distorted codes.
- PDF417: improved robustness when facing tight ROIs or small quiet zones.
- Truncated PDF417: improved robustness when facing tight ROIs or small quiet zones.
- Pharmacodes: increased robustness for out-of-specification bar size variations.
- New M_SEARCH_ANGLE_STEP control type for 1D codes.
- New M_CHECK_QUIET_ZONE control type to enable/disable assuming the existence of a valid quiet zone around the codes.
- New M_STRING_FORMAT control type, supported on Windows platforms except CE.
- New M_STRING_FORMAT control type: M_SIMPLIFIED_CHINESE for outputting decoded strings in Simplified Chinese (Windows 936) code page, for M_PDF417, M_MICROPDF417, M_TRUNCATED_PDF417, M_QRCODE, M_MICROQRCODE, and M_DATAMATRIX.
- New M_STRING_FORMAT control type: M_JAPANESE for outputting decoded strings in Japanese (Windows 932) code page, for M_PDF417, M_MICROPDF417, M_TRUNCATED_PDF417, M_QRCODE, M_MICROQRCODE, and M_DATAMATRIX.
- New M_STRING_FORMAT control type: M_LATIN for outputting strings in Latin (Windows 1252) code page, for M_PDF417, M_MICROPDF417, M_TRUNCATED_PDF417, M_QRCODE, M_MICROQRCODE, and M_DATAMATRIX.
- New M_STRING_FORMAT control type: M_KOREAN for outputting strings in Korean (Windows 949) code page, for M_PDF417, M_MICROPDF417, M_TRUNCATED_PDF417, M_QRCODE, M_MICROQRCODE, and M_DATAMATRIX.

2.2 Code verification

- New AIM DPM-1-2006 standard support for Data Matrix, QR and MicroQR codes.
- Complete ISO 15415 standard support for Data Matrix, QR and MicroQR codes.
- New M_VERIFICATION_STANDARD control type to select the desired specification to use for Data Matrix codes, QR codes, and Micro-QR codes. The possible values are: M_ISO_VERIFICATION, and M_AIMDPM_VERIFICATION.
- Deprecation and potential change of behavior: the old ANSI verification is deprecated and replaced, by default, by the ISO verification standard.
- New M_NUMBER_OF_SCANS_PER_ROW control type to set the number of scans per row for the verification of linear codes and cross-row codes.
- New M_INSPECTION_BAND_RATIO control type to set the inspection band ratio for the verification of linear codes and cross-row codes.

3. Fixed bugs

3.1 Code reading

- PDF417 code: start/stop pattern detection could not work properly.
- PDF417 code: could fail to correctly determine the number of columns and rows.
- Planet code: could fail to correctly determine the module size.
- Postnet code: could fail to correctly determine the module size.
- 4-State code: could fail to correctly determine the module size.
- PDF417 code: the error correction could fail when the number of codewords was large.
- 2D code drawings: a regular grid was drawn instead of the fetched positions.

3.2 Code writing

- Australian 4-State code: fixed crash when encoding raw data.
- EAN14 code: fixed potential heap corruption.
- PDF417 code: code not generated correctly: the symbol length descriptor was not computed correctly.

3.3 Barcode verification

- MicroPDF417 code: fixed potential Row Address Pattern (RAP) grade; it cannot be larger than 52.

4. Known issues

- 4.1 It is not recommended reinstalling MIL 9 Processing Pack 2 after MIL 9 Update 45: this will replace some files required for the proper functioning of this update.
- 4.2 When upgrading processing specific example projects from Visual Studio 2005 to Visual Studio 2008 or Visual Studio 2010 with UAC enabled, there will be a warning in the conversion log indicating that current user privileges have been added to the project. This will not cause any problems.

Section 8: Differences between MIL 9 Processing Pack 2 and MIL 9 Processing Pack 1

Table of Contents for Section 8

1. Overview
2. New processing functionality and improvements
 - 2.1. Processing specific examples (New!)
 - 2.2. Bead module (New!)
 - 2.3. Calibration module (New fixturing!)
 - 2.4. Color module
 - 2.5. 3dmap module
 - 2.6. Metrology module
 - 2.7. Code Reader module
 - 2.8. Blob module
 - 2.9. Measurement module
 - 2.10. Pattern Matching module
 - 2.11. Model Finder module
 - 2.12. Edge Finder module
 - 2.13. String Reader module
 - 2.14. OCR module
 - 2.15. Registration module
 - 2.16. Graphic functions
 - 2.17. Mim and general processing functions
 - 2.18. Interactive utilities
3. Fixed bugs
 - 3.1. Code reader module
 - 3.2. Blob module
 - 3.3. Calibration module
 - 3.4. String Reader module
 - 3.5. Edge Finder module
 - 3.6. Model Finder module
 - 3.7. Color module
 - 3.8. Metrology module
 - 3.9. 3dmap module
 - 3.10. Pattern matching module
 - 3.11. Measurement module
 - 3.12. Registration module
 - 3.13. OCR module
 - 3.14. Graphic functions
 - 3.15. Mim and general functions
 - 3.16. Interactive utilities
4. Mim functions and performance optimizations
5. Board specifics
6. Miscellaneous

1. Overview

- MIL 9 Processing Pack 2 includes new processing functionalities, performance optimizations and general improvements, such as: a new bead inspection module, a GUI for the color module, new fixturing capability, world input unit support, enhanced 3dmap capabilities,... and much more!
- MIL 9 Processing Pack 2 also installs ActiveMIL controls. For more information, see the ActiveMIL readme file.
- MIL 9 Processing Pack 2 incorporates and updates the display functionality beyond that introduced by MIL 9 Update 16.
- MIL 9 is the minimum requirement for this Processing Pack 2.

2. New processing functionality and improvements

1. Processing specific examples (New!)

The new %MIL_PATH%\Examples\Processing-Specific directory includes MIL processing specific examples (frameworks, utilities, applications...)

- Preprocessing: SuperResolution, Morphology

- Pattern Matching: PCBModelMatching
- Measurement: CircleMeasurement, AdvancedMeasurement
- Color Inspection: FoodInspection_Cereal
- Code: VariousCodeReading, FixturedCodeRead, DotSpacing, CodeGrading
- CameraLaser3D: PhotonFocus, SickRangerC, MatroxGatorEye
- 3dRobotic: CameraOnRobotArmCalibration
- 3dReconstruction: CameraLaserCalibration, StereoCalibration

2- Bead module (New!)

- New: module for inspecting beads such as sealants or adhesives. It calculates the bead's width and the bead's positions, and identifies out-of-range bead's properties and gaps.

3- Calibration module

- New: Fixturing capability in MIL.
- New: McalFixture API to move or to learn a relative position using either user-defined values or result values, such as Model Finder results or Pattern Matching results; this allows you to further fixture subsequent operations.
- New: M_..._INPUT_UNITS controls to set in world/pixel input units most of the module setting values (blob, code, measurement, bead,...)
- New: McalTransformImage supports M_OVERSCAN_FAST interpolation mode.
- New: McalRelativeOrigin and McalSetCoordinateSystem can now be set to a calibrated image; this allows you to further fixture subsequent operations.
- New: McalTransformResultAtPosition allows you to accurately transform from/to world/pixel angles values at a given location.
- New: M_CONSTANT_PIXEL_SIZE flag to inquire whether a calibrated image has a constant pixel size.
- Improvement: M_Y_AXIS_UP can now be inquired on a calibrated image.
- New: McalUniform API to associate a default uniform calibration to an image and to set its scale, rotation, and translation properties; equivalent inquiries added to retrieve information about translation, rotation, and scale values.
- New: M_WITH_CALIBRATION flag allows an associated calibration to be saved with its owner such as a buffer, a model finder context or a metrology context.
- New: McalDraw supports new operations to draw the coordinate systems, with optional axis, frames, tick marks, and legends.
- New: M_TRANSFORM_IMAGE_FILL_MODE is now deprecated and replaced by the use of combination flags in McalTransformImage.
- New: M_USE_DESTINATION_CALIBRATION McalTransformImage combination flag; this allows you to further fixture subsequent operations using corrected images.
- New: M_OUTPUT_UNITS has been deprecated in McalControl and replaced by the M_RESULT_OUTPUT_UNITS controls in modules.
- Improvement: better M_MATHEMATICAL_EXCEPTION protection when unstable configuration of calibration points are submitted.
- Improvement: it is now possible to pass an index to McalDraw when using M_DRAW_WORLD_POINTS or M_DRAW_IMAGE_POINTS on a calibrated image.
- New: McalRelativeOrigin(M_COMPOSE_WITH_CURRENT) receives coordinates in the relative coordinate system.
- New: McalDraw(M_DRAW_FIXTURING_OFFSET) to draw the fixture offset.
- Improvement: McalInquire(... M_Y_AXIS_UP...) is now available on image buffer.

4- Color module

- New: McolDistance and McolProject have been optimized using SIMD instructions.
- New: McolDistance and McolProject have been optimized for multi-core (MP) architectures.
- New: McolMatch with the vote method has been optimized using SIMD instructions and for multi-core (MP) architectures.
- New: McolMatch histogram based method useful for matching multi-colored areas.
- New: advanced inquires about color sample information and statistics.
- New: support for additional CIE distance types (M_CMC_PERCEPTIBILITY, M_CMC_ACCEPTABILITY, M_CIEDE94_GRAPHIC_ARTS, M_CIEDE94_TEXTILE, and M_CIEDE2000).
- New: McolDraw, M_INVERTED_COLORS combination flag for drawing LUT and result images using inverted colors.

5- 3dmap module

- New: uniform calibration automatically associated for generating calibrated depth-map.
- New: automatic scaling for corrected depth map.
- New: it is now possible to call multiple M3dmapExtract() on a same destination buffer, and to use this destination buffer as "3D point accumulator" by setting the M_EXTRACTION_CUMULATIVE control of the result buffer.
- New: M_CLEAR_DATA control to clear the result buffer data while leaving the current position unchanged.
- New: M3dmapSetGeometry function to fit a surface on a depth map or to define a geometry using analytic parameters in world units.
- New: M3dmapStat function to compute a variety of statistics on the depth map image, such as volume or deviations. Some statistics can also be measured relative to a reference geometry (see M3dmapSetGeometry).
- New: M3dmapArith function to perform point-to-point arithmetic between depth maps and/or reference geometries (see M3dmapSetGeometry).
- New: M3dmapDraw to draw various 3dmap context or result information or

- reference geometries.
- New: M3dmapSave, M3dmapRestore, M3dmapStream to save, restore, and stream 3dmap contexts.
- New: M_LASER_CONTEXT_TYPE flag to inquire about the context type (M_DEPTH_CORRECTION or M_CALIBRATED_CAMERA_LINEAR_MOTION).
- New: inquires and drawings to help the diagnostic of the result of the calibration process, such as the number of columns with missing data, the plane parameters, the calibration plane fit error,...
- New: M_REMOVE_LAST_SCAN control to undo last call to M3dmapAddScan.

6- Metrology module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: M_EDGESEL_SELECTION_RANK control to specify which edgel position to select relative to the orientation of rectangular search regions.
- New: segment M_CONSTRUCTION from arc/circle centers and/or local frames.
- New: local frame M_CONSTRUCTION from arc/circle centers and/or local frame
- New: circle M_CONSTRUCTION from arc/circle centers and/or local frames.
- New: arc M_CONSTRUCTION from arc/circle centers and/or local frames.
- Improvement: of the inner circle fit calculation.
- New: MmetName API to set/get names to features and tolerances.
- New: M_DRAW_NAME combination flag to draw feature and tolerance names.
- Improvement: the M_DRAW_REFERENCE_FEATURES flag is deprecated and is replaced by the M_DRAW_TOLERANCE_FEATURES flag.
- New: M_NUMBER_OF_CONSTRUCTION_FEATURE_LABEL to retrieve the number of features used to define a constructed feature or a tolerance.
- New: M_NUMBER_OF_CONSTRUCTION_FEATURE_INDEX to retrieve the number of features indices used to define a constructed feature or a tolerance.
- New: M_CONSTRUCTION_FEATURE_LABEL(i) to retrieve the label of the i-th feature to define a constructed feature or a tolerance.
- New: M_CONSTRUCTION_FEATURE_INDEX(i) to retrieve the label of the i-th feature index to define a constructed feature or a tolerance.
- New: M_METHOD to retrieve the method used for a constructed feature.
- New: M_DEPENDENCIES to inquire if a feature has some other features or tolerances derived or based on it.
- New: M_MODIFY MmetAdd... flag to modify the construction properties of an already existed feature or tolerance.
- New: MmetSave/Restore/Stream support M_WITH_CALIBRATION.

7- Code Reader module

- New: M_DISTORTION control to enable reading distorted codes. Currently supported is the M_UNEVEN_GRID_STEP distortion mode for Data Matrix codes.
- New: supports for 4-States (also known as IMB) barcodes with US, UK, Australian C and N encodings.
- New: the flag name M_RSSCODE is deprecated and replaced by the flag name M_GS1_DATABAR.
- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: M_..._INPUT_UNITS control to set input settings such as the cell size, the dot spacing, the angle, the aperture size, ... in world or pixel units.
- New: McodeRead supports input regions (see MbufSetRegion).
- New: supports for EAN14 and GS1-128 encodings.
- New: M_STRING_FORMAT control to set the output string format when applicable. Currently supported is the M_UPCE_COMPRESSED mode for UPC-E codes; M_GS1_HUMAN_READABLE for output in human readable (applied only on M_CODE128, M_GS1_128, M_EAN14, M_GS1_DATABAR, M_COMPOSITECODE and M_DATAMATRIX); and M_AUTO_FORMAT for default recommended output format.
- New: M_DATA_CODEWORDS result flag to retrieve the data code for most of the composite and 2D code types.
- Improvement: of the localization for 1D barcodes.
- Improvement: of the position accuracy when reading multiple codes.
- Improvement: MicroPDF417 start/stop grade symbol detection.
- Improvement: the reading of multiple codes with speed = M_VERY_LOW.
- New: McodeRead and McodeVerify take advantage of the new MbufSetRegion function used to set a region that defines the pixels of interest for an image.
- New: support for GS1 Data Matrix.

8- Blob module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: M_BOX_AREA feature to determine the blob's bounding box area.
- New: M_BOX_ASPECT_RATIO feature to determine the ratio between X and Y dimensions of the blob's bounding box.
- New: M_BOX_FILL_RATIO feature to determine the ratio between the blob's area and its bounding box area.
- New: M_CONVEX_HULL_FILL_RATIO feature to determine the ratio between the blob's area and its convex hull area.
- New: M_CONVEX_HULL_X and M_CONVEX_HULL_Y to determine the X and Y coordinates of the convex hull perimeter.
- New: M_NUMBER_OF_CONVEX_HULL_POINTS feature to determine the number of point of the convex hull.
- New: M_CONVEX_HULL_AREA feature to determine the blob's convex hull area.
- New: MblobCalculate additional optimizations using SIMD instructions for calculating the blob's RLE.
- New: M_RETURN_PARTIAL_RESULTS control flag to enable retrieving the partially calculated blobs when a stop criterion condition is met (i.e. when the M_MAX_BLOBS or M_TIMEOUT condition is met).
- New: MblobCalculate takes advantage of the new MbufSetRegion function used

to set a region that defines the pixels of interest for an image.
- New: M_WORLD_BOX, M_WORLD_FERET_X and M_WORLD_FERET_Y features and corresponding drawings to determine the features in the world system when using calibration.

9- Measurement module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: circle markers (M_CIRCLE) are now supported.
- New: MmeasSetScore API to define advanced edge selection rules.
- New: M_..._INPUT_UNITS flags to set input settings in world or pixel units
- New: M_SEARCH_REGION_CLIPPING control. When enabled, an optimal box will be computed when the search box falls partially outside the image limits.
- New: M_SEARCH_REGION_WAS_CLIPPED result to retrieve whether a measurement was performed using a clipped measurement box.
- New: M_FOUND_BOX_ANGLE status result can now be retrieved.
- New: M_POSITION_IN_MARKER result value to retrieve the distance relative to the origin of the measurement box of an edge or a stripe occurrence.
- New: M_FIT_ERROR_MAX result to retrieve the maximum distance between the fitted geometry and the transition positions found in measurement sub-regions for an edge, stripe, or circle marker.
- New: M_MAX_ASSOCIATION_DISTANCE control to reject outliers when fitting the marker on sub-box positions.
- New: M_NUMBER_OF_OUTLIERS result to retrieve the number of transition positions found in measurement sub-regions discarded by the fit process for an edge, stripe, or circle marker.
- New: M_LINE_END_POINT_FIRST and M_LINE_END_POINT_SECOND results.
- Improvement: some additional optimization have been performed when using sub-boxes.
- New: world unit line equation (A,B,C) parameters.
- Deprecation: M_POSITION_INSIDE_STRIPE flag has been renamed to M_INCLUSION_POINT_INSIDE_STRIPE and is used in conjunction with the M_INCLUSION_POINT flag.

10- Pattern Matching module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: the limitation of the max number of models allowed (i.e. 15) for MpatAllocAutoModel has been removed.
- New: M_SEARCH_ANGLE_TOLERANCE control can now be set to M_AUTO. If set to M_AUTO, the tolerance is optimally determined regarding the angular model auto-correlation and the new M_ROTATED_MODEL_MINIMUM_SCORE control value.
- New: M_FIRST_LEVEL control can now be set to either M_AUTO_SIZE_BASED or to the new M_AUTO_CONTENT_BASED mode, useful for but not limited to large models composed of small features.
- New: M_MAX_INITIAL_PEAKS control flag which can be set to M_ALL to force an exhaustive analysis of the correlation peaks when a very large set of potential occurrences (>100) is present in the target image.
- New: maximum first and last level limitation (i.e. 7) has been removed, and the new M_MODEL_MAX_LEVEL flag allows inquiring about the max possible level for a given model.
- New: MpatFindModel now supports 1-D target images.
- New: MpatFindModel has been optimized for multi-core (MP) architectures.
- New: MpatFindModel has been optimized using SIMD instructions for large models.

11- Model Finder module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: low accuracy control enables using a less accurate but faster method to compute the edgels' angle.
- New: MmodFind can now be done with a mix of uncalibrated models/target and uniform calibrated scale=1 models/target.
- New: saving/restoring model finder context with M_WITH_CALIBRATION.

12- Edge Finder module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: M_ANGLE_ACCURACY accuracy control. Setting M_LOW enables a less accurate but faster method to compute the edgels' angle.

13- String Reader module

- New: MstrRead supports input regions (see MbufSetRegion)
- New: MstrRead takes advantage of the new MbufSetRegion function used to set a region that defines the pixels of interest for an image.

14- OCR module

- New: M_RESULT_OUTPUT_UNITS to retrieve results in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).
- New: M_EXTRA_CHARACTERS control, which is useful when the target image string contains more characters than the expected characters to be read.

15- Registration module

- Improvement: the registration module is now robust to not-enough-memory exceptions.

16- Graphic functions

- New: MgraAllocList, MgraControllist, MgraInquireList, and MgraDraw, which enables vectorial capabilities in MIL.
- New: MgraLines capability to draw Polygons and Polyline geometries.
- New: M_TEXT_ALIGN_HORIZONTAL/VERTICAL control to set the text alignment.
- New: M_INPUT_UNITS control flag to support annotation capabilities in world / pixel units.
- New: M_ASSOCIATED_GRAPHIC_LIST_ID MdispControl flag to associate a graphic list to a MIL display object.
- New: M_UPDATE_GRAPHIC_LIST (M_ENABLE/M_DISABLE, MdispControl) to set whether or not modifications applied to the graphic-list associated to the display must be updated to the display.
- New: M_REGION_INSIDE_COLOR, M_REGION_INSIDE_SHOW, M_REGION_OUTSIDE_COLOR, and M_REGION_OUTSIDE_SHOW display controls to set the buffer region display properties.
- New: M_MOUSE_LEFT_BUTTON_DOWN, M_MOUSE_RIGHT_BUTTON_DOWN, M_MOUSE_LEFT_BUTTON_UP, M_MOUSE_RIGHT_BUTTON_UP, M_MOUSE_MOVE display hook types to hook on mouse events.
- New: M_MOUSE_POSITION_X, M_MOUSE_POSITION_Y, M_MOUSE_POSITION_BUFFER_X, and M_MOUSE_POSITION_BUFFER_Y flag to retrieve the information about the mouse position.
- New: display overlay now automatically follows the calibration of the selected buffer, when calibrated.
- New: Exclusive(M_EXCLUSIVE) displays can be set to the current desktop resolution using M_CURRENT_RESOLUTION DisplayFormat in MdispAlloc.

17- Mim and general processing functions

- New: MbufSetRegion function to set a region that defines the pixels of interest for an image. The region can be stored either in a vector format with the help of a graphics list, or in a raster format. Some processing operations such as McodeRead, McodeVerify, MstrRead, McalFixture, MblobCalculate... can use and take advantage of a region when set.
- New: MimFlatField function and context to perform an image flat field correction operation.
- New: MimRearrange function and context to copy an entire list of ROIs from a source buffer to a destination buffer at arbitrary locations.
- New: MimDeadPixelCorrection function and context to perform a dead pixel correction operation.
- New: MimMatch function and context to compute the matching score between a model image and a destination image using either the correlation or a faster sum-of-differences based calculation.
- New: MimThin, M_BINARY3 processing mode which usually performs faster than other thinning modes.
- New: MimPut() and MimGet() APIs to set / get values to Mim contexts.
- New: M_CUMULATIVE_VALUE histogram result (i.e. M_HIST_LIST result type) to retrieve the cumulative histogram of an image.
- New: M_PERCENTAGE histogram combination flag result (i.e. M_HIST_LIST result type) to be combined with M_VALUE or M_CUMULATIVE_VALUE to retrieve the histogram information as percent values.
- New: M_PERCENTILE_VALUE histogram result (i.e. M_HIST_LIST result type) to retrieve the percentile histogram values of an image.
- New: MimBinarize M_BIMODAL, M_PERCENTILE_VALUE, M_TRIANGLE_BISECTION_DARK, and M_TRIANGLE_BISECTION_BRIGHT modes. Note that invoking the automatic thresholding by setting the ConditionLow to M_DEFAULT is deprecated and replaced by the M_BIMODAL flag.
- New: MimArith operations Square, Cube, Square root, Ln, Log10, Log2 and Exp_const are supported.
- New: MimMorphic M_LEVEL operation to perform a morphological window leveling correction.
- New: MimMorphic M_OPEN and M_CLOSE morphological operations.
- New: MimMorphic M_TOP_HAT and M_BOTTOM_HAT morphological operations.
- New: MimStatMultiple to calculate pixel statistics on a sequence of images.
- New: MimDraw to draw Mim context and results informations.
- New: MimGetResult2d to retrieve 2D results from a Mim result object.
- New: support overscan color for MimThin and MimThick in binary mode using M_BINARY+M_OVERSCAN_REPLACE_MAX/M_OVERSCAN_REPLACE_MIN operation flag.
- New: M_RESULT_OUTPUT_UNITS to retrieve results such as M_POSITION_X and M_POSITION_Y in world or pixel units (replaces M_OUTPUT_UNITS in McalControl).

18- Interactive utilities

- New: GUI for the color module (matching context only).
- New: McolSave / Restore / and Stream interactive tools.
- New: Processing GUIs support RGB-to-HSL and HSL-to-RGB conversions in the image menu.
- New: button in the Code Reader GUI to save a txt report containing most of the results from a verify operation.
- Improvement: the metrology GUI pops a warning message when deleting a feature with dependencies.

3. Fixed bugs

1- Code Reader module

- Fix: MicroPDF417 could fail to read with scan reverse enabled and the target code rotated at 180 degrees.
- Fix: in the code verification related to a wrong interpretation of the validation of the results in section J.1. Phase 1 of the ISO 24728 spec.
- Fix: in rare situations, a pair (CW + RAP) can "lose" against the score of a pair (RAP + CW) in a MicroPDF417 code resulting in a bad reading result.

- Fix: string size min/max controls had no effect for Pharmacodes.
- Fix: large Maxicode symbols were often not read when using the adaptive threshold (M_ADAPTIVE).
- Fix: M_SCAN_EDGE_DETERMINATION_GRADE was wrong for cross-row codes.
- Fix: in some rare cases, an GS1 Databar could have an erroneous decode grade of F.
- Fix: McodeWrite did not scale codes optimally in the destination.
- Fix: M_SCAN_MODULATION result could be erroneous for several types of code
- Fix: the checksum calculation for code93 encoding 15 characters and more.
- Fix: McodeWrite/Read could crash when writing and reading "big" QR Code.
- Fix: McodeWrite for QR code was not using accurately the M_CELL_SIZE_MIN.
- Fix: ECMin verification fixed for 1D codes: now takes into account the full quiet zone as specified in ISO/IEC 15416.
- Fix: MicroPDF417 position in composite codes for the top left and bottom right position values.
- Fix: possible floating point exception fixed only when changing the FPU exception mode using MappAlloc.

2- Blob module

- Fix: MblobDraw was not multi-thread safe or exception safe with M_DRAW_BLOBS_CONTOUR and M_DRAW_HOLES_CONTOUR options when chains (M_CHAINS) were not previously calculated.
- Fix: when M_MAX_BLOBS condition was reached and when using the multi-core capability, MblobCalculate could fail to properly merge the results.
- Fix: API protection: chain codes cannot be computed when SAVE_RUNS is set to M_DISABLE.
- Fix: potential floating point exception when explicitly casting a large GetResult value using +M_TYPE_CHAR,...
- Fix: some timeout scenarios could cause a race condition leading to a GPF.

3- Calibration

- Fix: in particular cases, McalDraw could generate wrong output with calibrated destination buffer.
- Fix: the Tsai calibration process could lead to a floating-point exception in extreme cases.
- Fix: McalInquire(M_RELATIVE_ORIGIN_X/Y/Z/ANGLE) on an image could return incorrect values when the calibration was previously modified.
- Fix: McalTransformImage could produce wrong results for a specific McalAssociate/McalRelativeOrigin\McalTransformImage sequence of calls.
- Fix: McalTransformImage on a child followed by McalTransformCoordinate call could produce incorrect result values.
- Fix: the Tsai calibration could fail when using very large buffers.
- Fix: McalTransformImage with M_CORRECT_LENS_DISTORTION_ONLY could perform a full correction, regarding the order of operations.

4- String Reader module

- Fix: a call to MstrRead or MstrDraw changed the color of the default graphic context.
- Fix: a memory leak when restoring a fontless context.
- Fix: a potential floating point exception in x64 when reading with a fontless context.

5- Edge Finder module

- Fix: MedgeDraw, M_DRAW_VALUE didn't work properly on calibrated images.
- Fix: MedgeStream on a result could temporarily change the M_OUTPUT_UNITS of the calibration, when having calibrated results.
- Fix: MedgeGetResult(M_ELLIPSE_FIT_ANGLE) in pixels was not correct if calculated on a calibrated image.
- Fix: streaming of M_NEIGHBOR_ANGLE and M_NEIGHBOR_ANGLE_TOLERANCE edge results was not done properly.

6- Model Finder module

- Fix: MmodDefine was not working properly with a calibrated corrected image
- Fix: a potential overflow could lead to unstable results in rare cases.
- Fix: for synthetic models, the edges absent from the target image could wrongly contribute to the model score.
- Fix: a memory leak in the preprocessing of a geometric controlled context when the angle search was enabled.
- Fix: fractional M_EXTRACTION_SCALE was not properly handled for models defined from an Edge result buffer.
- Fix: MmodGetResult(M_CHAIN_xxx + M_MODEL) crashed on remote systems.
- Fix: MmodPreprocess fix a case where the First level could be negative.
- Fix: possible wrong first/last level values when using calibrated synthetic or CAD models.
- Fix: potential exception fixed in geometric controlled context with the search angle range enabled and when the reference point happens to be exactly on top of an edgel.
- Fix: MmodDraw M_DRAW_POSITION did not draw the reference point with proper angle (model or result) when using calibrated model/target.
- Fix: MmodFind with M_GEOMETRIC_CONTROLLED on synthetic model was not working properly when M_PIXEL_SCALE != 1.
- Fix: MmodFind "from seed" on calibrated image was not working properly when M_FIRST_LEVEL != M_LAST_LEVEL.
- Fix: CAD or edge result model could have an inaccurate calculated score.

7- Color module

- Fix: the sequence McolPreprocess, McolSetMethod, McolInquire could lead to

- an abnormal behavior of the module for a few inquired controls.
- Fix: M_LABEL_AREA_IMAGE_SIZE_BIT, when no image generation is enabled, must not be available in GetResult.
- Fix: McolAlloc with M_DEFAULT as object type was generating an exception.
- Fix: the average of the H band was not handling properly the angle wrap around leading to wrong H values around 0 / 360 degrees.

8- Metrology module

- Fix: the angle of the output frame was not taken into account for the angular measure of measured arcs and segments.
- Fix: retrieving result of a null edge data feature should not pop an error message.
- Fix: MmetDraw(M_DRAW_TEMPLATE_REFERENCE) was not propagating the calibration, if any, to destination image.
- Fix: MmetDraw(M_DRAW_TEMPLATE_REFERENCE) with LabelOrIndex M_GENERAL reports an error.
- Fix: some MmetDraw did not consider the destination calibration of the target image if any.
- Fix: in rare situations, a constructed intersection could lead to a crash when no edgels were found.

9- 3dmap module

- Fix: M3dmapCalibrate could have an unpredictable behavior when the X axis of the camera is perpendicular to the laser plane.
- Fix: M3dmapTriangulate documentation of parameters XPixelArrayPtr and YPixelArrayPtr was incorrect.

10- Pattern Matching module

- Fix: MpatInquire with M_PROC_FIRST_LEVEL always returned 0 for a circular overscan model instead of first level used.
- Fix: a model could not be found when the occurrence is at 180deg +-epsilon and delta angle was set to 180deg.
- Fix: MinSpacing with normalized rotated models did not take into account the effect of the rotation.
- Fix: occurrences at rotation partially occluded by the image border could lead to unstable match results.
- Fix: occurrences reference point could fall outside a specified search region when searching with rotation.

11- Measurement module

- Fix: MmeasStream + M_LOAD with two types of markers did not work.
- Fix: MmeasSetMarker + M_POSITION was not handled properly under specific conditions and sequence of Meas calls.
- Fix: M_POSITION_Y result values were wrong when using a calibrated source buffer.
- Fix: MmeasGetResult(...M_BOX_CORNER_BOTTOM_LEFT, M_BOX_CORNER...) results were not converted into world coordinates.
- Fix: measuring with boxes with no rotation in a flipped host buffer generated a crash.
- Fix: Angle result could have a +-180 degrees instability.
- Fix: the contrast of a marker could be negative when using M_SHEN/M_PREWITT filters.
- Fix: under some conditions, the angle of the box is returned instead of the real angle of the edge/stripe when angle mode is enabled.

12- Registration module

- Fix: MregCalculate could fail due to the accumulation of small numerical errors.

13- OCR module

- Fix: a potential exception in OCR General when reading in small regions.

14- Graphic functions

- Fix: a potential exception in MgraArc when one of the radii was set to 0.
- Fix: MgraControl(M_COLOR) and MgraControl(M_BACKCOLOR) were not working properly.

15- Mim and general processing functions

- Fix: MimBinarize 32F->U8 was not a thread-safe operation.
- Fix: the default combination flag M_DRAW_PEAKS of the MimDraw operation was not handled correctly.
- Fix: MthrControl(M_MP_XXX) with a specific MIL thread ID did not work.
- Fix: MimStat(...M_MIN, M_MAX...) did not return the correct results in 64 bits.
- Fix: MimBinarize did not work as expected with a float source containing values between 0 and 1 and a binary destination buffer.
- Fix: MbufCopyCond did not work correctly under x64 when the condition buffer was an offsetted child of a binary buffer.
- Fix: MimPolarTransform could access memory outside the source image in very rare situations.
- Fix: GPU buffers are not unlocked after some calls to Mim functions.
- Fix: MimBinarizeautomatic threshold on floating-point buffers did not work as expected.
- Fix: potential fetch outside the source buffer for MimMorphic binary operations.

16- Interactive utilities

- Fix: when opening a jpg image with the wrong extension in a GUI, and without the appropriate license lead to a GPF.
- Fix: when changing the current frame in a sequence, annotations were not redrawn, even if the current selection would indicate otherwise.
- Fix: switching documents (images) in the GUIs with Ctrl+Tab or Ctrl+Shift+Tab cleared the annotations.
- Fix: Model Finder GUI crashed when performing a search in a small image (less than 16x16) using a geometric controlled context.
- Fix: feature IDs in the Metrology GUI were inconsistent between the main feature's tree and features in sub dialogs.
- Fix: modifying the model finder model reference point angle should require to re-preprocess.

4. Mim functions and performance optimizations

- New: McolDistance and McolProject have been optimized using SIMD instructions.
- New: McolDistance and McolProject have been optimized for multi-core (MP) architectures.
- New: McolMatch with the vote method has been optimized using SIMD instructions and for multi-core (MP) architectures.
- New: MpatFindModel has been optimized for multi-core (MP) architectures.
- New: MpatFindModel has been optimized using SIMD instructions for large models.
- New: MblobCalculate has been optimized for multi-core (MP) architectures.
- New: MblobCalculate additional optimizations using SIMD instructions for calculating the blob's RLE.
- New: MimMorphic, in binary mode, has been optimized using SIMD instructions.
- New: MimResize M_AVERAGE with integer scale factors has been optimized using SIMD instructions.
- New: MimResize M_AVERAGE has been optimized for multi-core (MP) architectures.
- New: MimConvolve special moving average optimization for uniform user-defined kernels.
- New: MimMorphic M_DILATE/M_ERODE special optimization for structural elements filled with 0s.
- New: MimMorphic M_DILATE/M_ERODE special optimization for 1D structural elements filled with 0.
- New: optimized MP predefined IIR filters (MimConvolve).

5. Board specifics

- Fix [ODC]: McodeStream(M_SAVE_REPORT) was not working properly on Odyssey platforms.
- Fix [ODC]: a memory leak in MmodPreprocess on Odyssey when the kernel size of the filter was large (using kernel mode and high filter smoothness).

6. Miscellaneous

- Predefined MIL contexts (%MIL_PATH%\..\Contexts\):
No new context.
- Calibration grid samples (%MIL_PATH%\..\images):
No new sample.
- Enhanced DispD3D DLL to display 3D depth maps.
- A Matrox M-Series graphics cards can now be used as a fingerprint for, and storage of, a MIL license. Note that the Matrox display driver must be started before the first call to MappAlloc.

Section 9: Differences between MIL 9 Processing Pack 1 and MIL 9

Table of Contents for Section 9

1. Overview
2. New processing functionality and improvements
 - 2.1. Color module (New!)
 - 2.2. 3dmap module (New!)
 - 2.3. Metrology module
 - 2.4. Code Reader module
 - 2.5. Edge Finder module
 - 2.6. Calibration module
 - 2.7. Blob module
 - 2.8. String Reader module
 - 2.9. Pattern Matching module
 - 2.10. Model Finder module
 - 2.11. Registration module
 - 2.12. Mim and general processing functions
 - 2.13. Interactive utilities
3. Fixed bugs
 - 3.1. Metrology module
 - 3.2. Code reader module
 - 3.3. Blob module
 - 3.4. String Reader module
 - 3.5. Edge Finder module
 - 3.6. Model Finder module

- 3.7. Ocr module
 - 3.8. Pattern Matching module
 - 3.9. Registration module
 - 3.10. Mim and general functions
 - 3.11. Interactive utilities
 - 4. Mim functions and performance optimizations
 - 5. Board specifics
 - 6. Miscellaneous
-

1. Overview

- MIL 9 Processing Pack 1 includes new processing functionality, performance optimizations and general improvements.
- MIL 9 Processing Pack 1 also includes new ActiveMIL processing controls. For more information, see the ActiveMIL readme file.
- MIL 9 is the minimum requirement for all upcoming Processing Packs until the next major release.

2. New processing functionality and improvements

1- Color module (New!)

- New: a color module to perform color vision processing, including:
 - supervised color matcher tools,
 - color separation operators,
 - and color distance operators.
 The color tools can be parametrized using various controls. Refer to the MIL Help file for more details.

2- 3dmap module (New!)

- New: a 3dmap module to generate 3D data, including:
 - camera-laser scanning devices calibration tools
 - camera-laser scanning devices depth map generation tools
 - camera-laser scanning devices fill holes function
 - multi-camera point triangulation tool.

3- Metrology module

- New: constructed feature construction methods (bisector between two lines, intersection point between extended features,
- New: A local frame constructed feature can now be constructed from a single point. The orientation of the local frame can be set using MmetControl and is relative to the parent frame.
- Improvement: of the drawing of an infinite radial region.

4- Code Reader module

- New: code verifications for 1D codes (specifications: ISO 15416, 15417, 15420, 16388, 16390)
- New: code verifications for 2D codes (specifications: ISO 24724 (RSS), 15438 (PDF417 & Trunc. PDF417), 24723 (Composite), 24728 (MicroPDF417))
- New: M_APERTURE_MODE control flag to set the type of aperture to apply when performing a code verification operation. When enabled, the aperture mode can be set to an absolute size value or to a relative size to the code module size.
- New: M_RELATIVE_APERTURE_FACTOR control flag to set the relative aperture factor to the module size for optimal code verification operation.
- New: M_ABSOLUTE_APERTURE_SIZE control flag to set the absolute diameter size for optimal code verification operation.
- New: M_MAXIMUM_CALIBRATED_REFLECTANCE to set the reference "maximum calibrated value" used in the scan reflectance profiles analysis.
- New: M_MINIMUM_CALIBRATED_REFLECTANCE to set the reference "minimum calibrated value" used in the scan reflectance profile analysis.
- New: M_LINEAR_COMPONENT combination flag to retrieve results of the linear component of a composite code. For example: McodeGetResult(.., M_NUMBER_OF_ROWS + M_LINEAR_COMPONENT, ..)
- New: M_2D_COMPONENT combination flag to retrieve results of the 2D component of a composite code.
- New: M_NUMBER_OF_CODEWORDS to retrieve the number of codewords in the symbol after a read or a verify operation for all 2D codes except for non-ECC200 Data Matrix symbols.
- New: M_NUMBER_OF_ERROR_CORRECTION_CODEWORDS to retrieve the number of error correction codewords in the symbol after a read or a verify operation for all 2D codes except non-ECC200 Data Matrix symbols.
- New: M_NUMBER_OF_ERRORS to retrieve the number of errors found in the symbol after a read or a verify operation for all 2D codes except for non-ECC200 Data Matrix symbols.
- New: M_NUMBER_OF_ERASURES to retrieve the number of erasures found in the symbol after a read or a verify operation for all 2D codes except for non-ECC200 Data Matrix symbols.
- Updated: M_OVERALL_SYMBOL_GRADE now returns a numeric value instead of a letter, in order to respect ISO 15415 and ISO 15416 specifications.
- New: M_NUMBER_OF_ROWS to retrieve the number of rows verified in the symbol for 1D codes and RSS codes.
- New: M_NUMBER_OF_SCANS_PER_ROW to retrieve the number of scan reflectance profiles for a given row for 1D codes and RSS codes.
- New: M_NUMBER_OF_SCANS to retrieve the total number of scan reflectance profiles analyzed for 1D codes and RSS codes.
- New: M_ROW_OVERALL_GRADE to retrieve the overall grade per for 1D codes and RSS codes.
- New: M_SCAN_DECODE_GRADE to retrieve per profile if the data was decoded correctly.

- New: M_SCAN_SYMBOL_CONTRAST to retrieve per profile the symbol contrast(SC).
- New: M_SCAN_SYMBOL_CONTRAST_GRADE to retrieve per profile the grade based on the symbol contrast (SC) parameter.
- New: M_SCAN_REFLECTANCE_MINIMUM to retrieve per profile the minimum reflectance (Rmin) parameter
- New: M_SCAN_REFLECTANCE_MINIMUM_GRADE to retrieve per profile the grade based on the minimum & maximum reflectance (Rmin & Rmax) parameters.
- New: M_SCAN_REFLECTANCE_MAXIMUM to retrieve per profile the maximum reflectance (Rmax) parameter.
- New: M_SCAN_EDGE_CONTRAST_MINIMUM to retrieve per profile the minimum edge contrast (ECmin) parameter.
- New: M_SCAN_EDGE_CONTRAST_MINIMUM_GRADE to retrieve per profile the grade based on the minimum edge contrast (ECmin) parameter.
- New: M_SCAN_MODULATION to retrieve per profile the modulation (MOD) parameter.
- New: M_SCAN_MODULATION_GRADE to retrieve per profile the grade based on the modulation (MOD) parameter.
- New: M_SCAN_ERN_MAXIMUM to retrieve per profile the element reflectance non-uniformity maximum (ERNmax) parameter.
- New: M_SCAN_DEFECTS to retrieve per profile the defects parameter (based on ERNmax and SC).
- New: M_SCAN_DEFECTS_GRADE to retrieve per profile the grade based on the defects parameter.
- New: M_SCAN_DECODABILITY to retrieve per profile the decodability parameter (V).
- New: M_SCAN_DECODABILITY_GRADE to retrieve per profile the grade based on the decodability parameter (V).
- New: M_SCAN_REFLECTANCE_PROFILE_GRADE to retrieve per profile the scan reflectance profile grade, which is equal to the lowest grade of any scan profile grade.
- New: M_SCAN_PRINT_CONTRAST_SIGNAL to retrieve per profile the print contrast signal (PCS) of the scan profile.
- New: M_SCAN_PROFILE_START_X to retrieve the x coordinate of the start of a scan profile.
- New: M_SCAN_PROFILE_START_Y to retrieve the y coordinate of the start of a scan profile.
- New: M_SCAN_PROFILE_END_X to retrieve the x coordinate of the end of a scan profile.
- New: M_SCAN_PROFILE_END_Y to retrieve the y coordinate of the end of a scan profile.
- New: M_SCAN_QUIET_ZONE to retrieve the worst quiet zone for Code 128, EAN/UPC, Code 39, Interleaved 2 of 5, Codabar and Code 93.
- New: M_SCAN_QUIET_ZONE_GRADE to retrieve the grade based on the quiet zone size for Code 128, EAN / UPC, Code 39, Interleaved 2 of 5, Codabar and Code 93.
- New: M_SCAN_GUARD_PATTERN to retrieve the maximum interior guard pattern element size for Rss codes, and composite codes containing a Rss code.
- New: M_SCAN_GUARD_PATTERN_GRADE to retrieve the grade based on the interior guard pattern elements size for Rss codes, and composite codes containing an Rss code.
- New: M_SCAN_WIDE_TO_NARROW_RATIO to retrieve the wide to narrow ratio for Code 39, Interleaved 2 of 5 and Codabar.
- New: M_SCAN_WIDE_TO_NARROW_RATIO_GRADE to retrieve the grade based on the wide to narrow ratio for Code 39, and Interleaved 2 of 5 and Codabar.
- New: M_SCAN_INTERCHARACTER_GAP to retrieve the maximum intercharacter gap in the scan profile for Code 39 and Codabar.
- New: M_SCAN_INTERCHARACTER_GAP_GRADE to retrieve the grade based on the maximum intercharacter gap for Code 39 and Codabar.
- New: M_CODEWORD_YIELD to retrieve the codeword yield result for the symbol for MicroPDF417, PDF417 and Truncated PDF417.
- New: M_CODEWORD_YIELD_GRADE to retrieve the grade based on the codeword yield for MicroPDF417, PDF417 and Truncated PDF417 codes.
- New: M_UNUSED_ERROR_CORRECTION_GRADE to retrieve the grade based on the unused error correction is now supported for MicroPDF417, PDF417 and Truncated PDF417 codes.
- Updated: M_START_STOP_PATTERN_GRADE to retrieve the grade based on the analysis of the start/stop pattern (or RAPs) in the symbol for MicroPDF417, PDF417 and Truncated PDF417 codes. Additionally, this result is now a numeric value. See ISO 15415 for more details.
- Updated: M_DECODABILITY_GRADE to retrieve the grade based on the codeword print quality assessment based on the decodability grade for MicroPDF417, PDF417 and Truncated PDF417 codes only. For linear symbologies, see M_SCAN_DECODABILITY_GRADE instead.
- Updated: M_DEFECTS_GRADE to retrieve the grade based on the codeword print quality assessment based on the defects grade for MicroPDF417, PDF417 and Truncated PDF417 codes only. For linear symbologies, see M_SCAN_DEFECTS_GRADE instead.
- Updated: M_MODULATION_GRADE to retrieve the grade based on the codeword print quality assessment based on the modulation grade for MicroPDF417, PDF417 and Truncated PDF417 codes only. For linear symbologies, see M_SCAN_MODULATION_GRADE instead.
- New: M_CODEWORD_DECODABILITY_GRADE to retrieve the decodability grade of each codeword in the symbol.
- New: M_CODEWORD_DEFECTS_GRADE to retrieve the defects grade of each codeword in the symbol.
- New: M_CODEWORD_MODULATION_GRADE to retrieve the modulation grade of each codeword in the symbol.
- New: M_DRAW_SCAN_PROFILES drawing option to draw the scan profiles examined after a verify operation.
- New: M_DRAW_REFLECTANCE_PROFILE drawing option to draw the scan reflectance profiles analyzed by the verify operation.
- New: M_STRING_SIZE_MIN control flag to specify the minimum string length requirement of a code read.

- New: M_STRING_SIZE_MAX control flag to specify the maximum string length requirement of a code read.
- New: M_BEARER_BAR control flag to specify the presence of bearer bars around the code in order to read at unknown search angles.
- New: Generation of a verification report can be done through McodeStream with the operation M_SAVE_REPORT.
- New: BC412 now supports the M_ECC_CHECK_DIGIT error correction.

5- Edge Finder module

- New: M_FERET_ANGLE_SEARCH_START and M_FERET_ANGLE_SEARCH_END to set the angle range limits used to compute the Feret min and max diameters and angles.

6- Calibration module

- New: M_3D_ROBOTICS calibration context to fully calibrate the system camera mounted on a robotic arm.
- New: M_ROBOT_BASE_COORDINATE_SYSTEM coordinate system to describe the robot base coordinate system (M_3D_ROBOTICS context only)
- New: M_NUMBER_OF_CALIBRATION_POSES, M_GLOBAL_AVERAGE_PIXEL_ERROR, M_GLOBAL_AVERAGE_WORLD_ERROR, M_GLOBAL_MAXIMUM_PIXEL_ERROR, and M_GLOBAL_MAXIMUM_WORLD_ERROR inquiries (M_3D_ROBOTICS context only).
- New: McallList and McalGrid M_ACCUMULATE mode to accumulate robot poses (M_3D_ROBOTICS context only).
- New: McalInquireSingle to retrieve information about individual poses (M_3D_ROBOTICS context only).
- New: McalGrid supports a new type of calibration grid (M_CHESSBOARD_GRID).
- New: M_TRANSFORM_FILL_MODE McalControl flag to set the McalTransformImage operation to the new modes M_FIT, M_CLIP, M_CORRECT_LENS_DISTORTION_ONLY, or M_USER_DEFINED.
- New: M_TRANSFORM_IMAGE_WORLD_POS_X and M_TRANSFORM_IMAGE_PIXEL_SIZE_X control flag to control the offset and scaling behavior of McalTransformImage when setting the M_TRANSFORM_FILL_MODE to M_USER_DEFINED.
- New: McalTransformImage M_EXTRACT_LUT_(X,Y) mode to retrieve the transformation warping LUT.
- New: McalDraw function to draw various calibration results (extracted grid points, calibration grid area, calibration grid filled area,...)
- New McalTransformCoordinate3DList to transform list of points between the various 2D and 3D coordinate systems (M_TSAI_BASED and M_3D_ROBOTICS contexts only).
- New: M_DISPLACE_RELATIVE_COORD McalList and McalGrid option. This mode can be used to find the 3D posture of a known object (M_3D_ROBOTICS contexts only)
- Deprecation: M_LOCATE_CAMERA_ONLY is renamed M_DISPLACE_CAMERA_COORD.
- New: the calibration module now supports Chessboard calibration grids.

7- Blob module

- New: the calculation operation stops when the max number of blob, set using the M_MAX_BLOBS control value, is reached.
- New: M_MAX_BLOBS_END result value to retrieve if the M_MAX_BLOBS limit has been reached.
- New: MblobGetRuns supports M_ALL to retrieve the runs for all blobs.
- New: M_INPUT_SELECT_UNITS flag in order to provide world units entries to MblobSelect function.
- Improvement: improved robustness to lack of memory exceptions.

8- String Reader module

- New: fontless application specific contexts (ex. ANPR context) which do not require the explicit definition of fonts. Refer to the MIL Help file for more details.
- Improvement: improved robustness to lack of memory exceptions.
- Improvement: robustness has been improved when reading punctuations.

9- Pattern Matching module

- New: MpatStream function to stream pattern matching contexts.
- Improvement: Mpat model size limit has been removed.

10- Model Finder module

- New: GEOMETRIC_CONTROLLED contexts can now search within an angle range setting the M_SEARCH_ANGLE_RANGE control value to M_ENABLE.

11- Registration module

- New: M_SUPER_RESOLUTION control to perform super-resolution image composition.
- New: M_SR_PSF_TYPE, M_SR_PSF_RADIUS, and M_SR_SMOOTHNESS control flags to setup the super-resolution operation.
- New: M_TRANSFORMATION_TYPE now supports M_TRANSLATION_ROTATION_SCALE.

12- Mim and general processing functions

- New: MimThin M_BINARY2 binary thinning method.

13- Interactive utilities

- New: support the subpixel drawing of the line approximation result in the Edge Finder GUI.

3. Fixed bugs

1- Metrology module

- Fix: MmetGetResult could crash when retrieving result of non calculated features.
- Fix: potential memory leak when releasing a context or or crash when re-setting the reference template when the reference template is defined from an edge result buffer.
- Fix: the status of a measured feature was failing is the coverage was equal to the min coverage limit.
- Fix: when a feature is calculated using a non infinite region, then changing to infinite region and re-calculating leads to invalid results.
- Fix: an angularity tolerance with a range 0-360 was always returning a fail status instead of a pass status.
- Deprecation: MmetControl M_INDEX selection flag is renamed M_OCCURRENCE.
- Fix: the point construct using the max distance method between a circle or an arc and another feature was sometimes returning two points instead of one.
- Fix: when using calibration, accuracy of edgel orientation has been improved.
- Fix: MmetCalculate could crash when a NULL context ID was provided.
- Fix: MmetCalculate could crash when a NULL system ID was provided.
- Fix: inner/outer fit segment was not always returning the expected result.

2- Code Reader module

- Fix: RSS codes decoding was expecting quiet zones, but only the guard patterns are required.
- Fix: the timeout end result did not return the appropriate M_STATUS_TIMEOUT_END status.
- Fix in some rare situation for 1D codes, the reading operation could fail when setting the angle range to 180 degrees.
- Fix: some RSS Expanded codes could be decoded as RSS Expanded Stacked codes.
- Fix: CC-A MicroPDF417 symbol could fail decoding some codewords.
- Fix: for 1D codes, the position result could be false when the code angle was near a multiple of 90 degrees.
- Fix: UCC-EAN128 /PDF417 Composite codes generated by McodeWrite was not always having the mandatory quiet zones size.
- Fix: in rare situations, PDF417 error correction mechanism could lead to false read.
- Fix: writing a composite code with a very long string could lead to a GPF.
- Fix: Pharmacode position could be offsetted when scan reverse was enabled or when reading multiple models.
- Fix: MicroPDF417 decoding has been improved when the microPDF417 is part of a composite code.
- Fix: the cell number X was incorrect for Truncated PDF417 symbols.
- Fix: reading a code at very low speed or with multiple code models could enable the scan reverse, wrongly overriding the user value.
- Fix: the maximum cell size limitation for 2D matrix code (DM, QR and Maxicode) has been removed.
- Fix: general bug fixes have improved the reading of composite code containing a RSS code, Maxicode, PDF417, Truncated PDF417, composite codes with CC-C, Data Matrix ECC200.
- Fix: general bug fixes have improved the verification of MicroPDF417, and composite codes with CC-C.

3- Blob module

- Fix: potential MblobSelect crash when merging blobs with a selection criterion using the blob area feature.
- Fix: MblobControl(M_TIMEOUT, M_DEFAULT) was not deactivating the a previously set timeout.

4- String Reader module

- Fix: the homogeneity score was not always retruning 100% for a string composed of a single character.
- Fix: the drawing of the underscore character was not respecting the baseline value.
- Fix: some string experts error reports were ambiguous and have been clarified.

5- Edge Finder module

- Fix: MedgeGetNeighbors was not working properly when the M_SEARCH_RADIUS_MAX control was not previously set.
- Fix: MedgeGetNeighbors has been improved for large search radius values.
- Fix: the follwing combination [crest context, filter type Shen, filter mode kernel], the calculation of the edges could generate a crash.
- Fix: the drawing of the chain approximation segments was erroneous under Windows Vista 64 bits.

6- Model Finder module

- Fix: a potential crash when matching in an edge result buffer calculated using an M_CREST edge context type.
- Fix: potential crash during a match when the M_DETAIL_LEVEL control is set to M_LOW.
- Fix: the search region was erroneous when setting a pixel scale !=1 for a DXF based CAD model.
- Fix: small radius arcs from DXF based models were not properly extracted.

7- OCR module

- Fix: potential crash of a read operation when using an M_GENERAL OCR context type with the M_TARGET_CHAR_SPACING set to M_SAME or to a user defined amount of spaces.
- Fix: in very rare situations, two successive readings could lead to different results.

8- Pattern Matching module

- Fix: parameter checking fixes for MpatControl and MpatInquire functions.

9- Registration module

- Fix: potential memory leak in MregSave/MregRestore/MregStream of a result element.

10- Mim and general processing functions

- Fix: in rare situation, a potential crash in MimWarp when using M_WARP_LUT mode has been fixed.
- Fix: MimEdgeDetect accuracy has been improved when using float buffers.
- Fix: small verticals Mpat models (size X < 4) are now supported.
- Improvement: performance of multi-thread MIL applications is improved when the processing involves calls to modules (blob, ocr,...)

11- Interactive utilities

- Fix: when multiple context were opened, adding a feature in one context was adding the feature in all context.
- Fix: context calibration information was lost when cancelling the feature properties dialog.

4. Mim functions and performance optimizations

- Improvement: MimLocatePeakID, MimLocateEvent, and MimStat reoptimization and general optimizations of the image processing functions.
- New: MIL now supports multiprocessing optimizations for multi-core and multi-processor machines.
- New: M_MP_USE flag to enable / disable the use of MP capability either globally using MappControl() or the at thread level using MthrControl().
- New: M_MP_MAX_CORES_PER_THREAD flag to set globally a maximum number of cores to use in any processing thread using MappControl()
- New: M_MP_MAX_CORES flag to set at the thread level a maximum number of cores to use using MthrControl()
- New: M_MP_CORES_NUM flag to retrieve the number of cores available to the application using MappInquire()
- New: M_MP_MAX_CORES_EFFECTIVE to retrieve the effective number of cores used for MP processing by the given thread using MthrInquire()
- New: the following image processing functions have been optimized for multiprocessing usage (MP):
 - MbufBayer
 - MbufClear
 - MbufCopy
 - MbufCopyClip
 - MbufCopyColor2d
 - MbufCopyColor
 - MbufCopyCond
 - MbufCopyMask
 - MimArith
 - MimArithMultiple
 - MimBinarize
 - MimClip
 - MimConnectMap
 - MimConvert
 - MimConvolve (except IIR predefined filters)
 - MimDilate/MimErode
 - MimEdgeDetect
 - MimFindExtreme
 - MimHistogram
 - MimLutMap
 - MimMorphic (except AreaOpen and AreaClose operations)
 - MimOpen/MimClose
 - MimProject
 - MimRank
 - MimResize
 - MimRotate
 - MimShift
 - MimThick
 - MimThin
 - MimTransform
 - MimWarp
 - MimFlip
 - MimLocateEvent
 - MimLocatePeakID
 - MimStat
 - MimTranslate
 - MimDeinterlace
 - MimPolarTransform
 - JPEG and JPEG2000 Compression

5. Board specifics

- Fix [ODC]: The system hangs when a wrong template id is passed to a Metrology context allocated on an odyssey system.

6. Miscellaneous

- Predefined contexts

The new %MIL_PATH%\..\Contexts\ directory includes the following predefined MIL contexts:

```
+ Ocr context: SEMI_M12-92.mfo
+ Ocr context: SEMI_M13-88.mfo
+ Ocr context: SEMI.mfo
+ Ocr context: MICR_CMC_7.mfo
+ Ocr context: MICR_E_13B.mfo
+ Ocr context: OCR_A_BT.mfo
+ Ocr context: OCR_B_BT.mfo
+ Code context: SEMI_T1-95r0303.mco
+ Code context: SEMI_T2-0298E.mco
+ Code context: SEMI_T7-0303.mco
+ String Reader context: FONTLESS_ANPR.msr
+ String Reader context: FONTLESS_EUROPEAN_ANPR.msr
+ String Reader context: FONTLESS_MACHINE_PRINT.msr
```

- Calibration grid samples

Calibration grid samples are included in the %MIL_PATH%\..\images directory:

```
+ CircleCalibrationGrid_15x16_Letter.pdf
+ CircleCalibrationGrid_18x22_Letter.pdf
+ ChessboardCalibrationGrid_15x16_Letter.pdf
+ ChessboardCalibrationGrid_18x22_Letter.pdf
```

Section 10: Differences between MIL 9 and MIL 8 Processing Pack 4

Table of Contents for Section 10

1. Overview
2. New processing functionality and improvements
 - 2.1. Metrology module
 - 2.2. Code Reader module
 - 2.3. Calibration module
 - 2.4. Blob module
 - 2.5. String Reader module
 - 2.6. Measurement module
 - 2.7. Registration module
 - 2.8. Pattern Matching module
 - 2.9. Mim and general processing functions
 - 2.10. Interactive utilities
3. Fixed bugs
 - 3.1. Metrology module
 - 3.2. Code Reader module
 - 3.3. Calibration module
 - 3.4. Blob module
 - 3.5. String Reader module
 - 3.6. Measurement module
 - 3.7. Edge Finder module
 - 3.8. Registration module
 - 3.9. Model Finder module
 - 3.10. OCR module
 - 3.11. Pattern Matching module
 - 3.12. Mim and general functions
 - 3.13. Interactive utilities
4. Mim functions and performance optimizations
5. Board specifics

1. Overview

- MIL 9 includes new processing functionality, performance optimizations and general improvements.
- MIL 9 also includes new ActiveMIL processing controls. For more information, see the ActiveMIL readme file.
- MIL 9 is the minimum requirement for all upcoming Processing Packs until the next major release.

2. New processing functionality and improvements

1- Metrology module

- New: the diameter of a circle feature can be retrieved.
- New: the circle radius and arc radius tolerances can be defined.
- Improvement: the MmetAddFeature BuildOperation parameter can now be inquired.
- Improvement: the MmetAddFeature FeatureLabelArray parameter can now be inquired.

2- Code Reader module

- New: the numerical values of the parameters estimated during a call to McodeVerify can be retrieved.

- New: support for M_FOREGROUND_ANY for linear 1D codes and microPDF417.
- Improvement: high accuracy support for position results when using adaptive threshold.
- Improvement: the accuracy of geometric results (position, length, height, width, cell size) has been improved.
- Improvement: general performance improvements in decoding and more specifically, for QR codes.
- Improvement: the estimation of the position of RSS codes has been improved.
- Improvement: reading composite codes at an angle is more robust
- Improvement: reading RSS Expanded Stacked has been improved when at low speed.
- Improvement: the position of the corners of the read code can be retrieved.

3- Calibration module

- New: M_GRID_CORNER_HINT_X/Y flag which allows for a user defined position to help solve the grid orientation ambiguity when the grid is reverted or heavily rotated (ex. 45 degrees).
- New: M_CALIBRATION_STATUS to retrieve information regarding the failure of a calibration.
- New: M_Y_AXIS_UP to inquire about the orientation of the Y-axis.
- New: M_TSAI_BASED calibration method for full 3D camera positioning with the estimation of the extrinsic and intrinsic parameters of the camera.
- New: M_CALIBRATION_PLANE control to set the associated calibration grid coordinate system (M_TSAI_BASED method only).
- New: M_DISTORTION_RADIAL_1 to retrieve the estimated lens radial distortion (M_TSAI_BASED method only).
- New: M_FOCAL_LENGTH to retrieve the estimated effective pinhole focal length in horizontal pixel units (M_TSAI_BASED method only).
- New: M_CCD_ASPECT_RATIO to set the aspect ratio of the CCD (M_TSAI_BASED method only).
- New: M_PRINCIPAL_POINT_X/Y to set the position of the optical center (M_TSAI_BASED method only).
- New: M_LINK_TOOL_AND_CAMERA to set a rigid link between the tool coordinate system and the camera coordinate system (M_TSAI_BASED method only).
- New: M_LOCATE_CAMERA_ONLY McalList/McalGrid flag to re-calibrate the camera position relative to a new grid system (M_TSAI_BASED method only).
- New: M_TOOL_POSITION_X/Y/Z to translate the tool coordinate system relative to the absolute coordinate system.
(note: in MIL 9 Processing Pack 1, the name has changed to M_DISPLACE_CAMERA_COORD)
- New: McalSetCoordinateSystem() to move the position of a coordinate system relative to another coordinate system using various transformation methods (M_TSAI_BASED method only).
- New: McalGetCoordinateSystem() to inquire the position of a coordinate system relative to another coordinate system (M_TSAI_BASED method only).
- New: M_CALIBRATION_CHILD_OFFSET_X/Y to specify the child offsets when associating a calibration to a buffer or to a child buffer.

4- Blob module

- New: M_LABELED_TOUCHING flag which allows the detection of blobs from labels, where touching blobs with different labels are considered to be distinct blobs.

5- String Reader module

- New: the bounding box of the string can be retrieved.
- New: the bounding box of the individual characters can be retrieved.
- Improvement: a label can be associated to a string model.
- Improvement: a label can be associated to a font model.

6- Measurement module

- New: the measurement regions can be sub-divided into sub regions for a better fit of the position and the angle of the edge.

7- Registration module

- New: a score of overlapped regions of two registered images can be enabled and retrieved.
- New: support for alpha-blending of the overlapped regions.

8- Pattern Matching module

- New: M_SAVE_SUMS control for the MpatSetSearchParameter() function to enable the saving of results of the individual sums involved in the correlation formula.
- New: additional flags for MpatGetResult to retrieve the results of the individual sums involved in the correlation formula (M_SUM_I, M_SUM_II, M_SUM_III, M_SUM_IV, M_SUM_V, M_SUM_MM).
- New: additional flags for MpatGetResult to retrieve the number of pixels used to compute the correlation (M_NUMBER_OF_PIXELS).

9- Mim and general processing functions

- New: MimRestore, MimSave and MimStream functions to save and restore Mim contexts.

- New: MimConvert matrix multiplication for user defined color conversions.
- New: added support for Magnitude and Phase in MimTransform reverse FFT operation.
- New: MimTransform new M_POLAR method to transform cartesian system values from/to polar system values.
- New: new 2x2 average Bayer demosaicing method.
- New: ability to stop a focus operation for MdigFocus.
- Improvement: MbufBayer(M_ADAPTIVE, M_COLOR_CORRECTION) now supports 16 bits buffers.
- Improvement: MimTransform(M_FFT, M_REVERSE) with floating-point sources and integer destinations accepts the M_SATURATION flag.

10- Interactive utilities

- New: TSAI based calibration mode has been added to the calibration dialog.
- New: the Metrology module now supports interactive dialogs.

3. Fixed bugs

1- Metrology module

- Fix: the drawing of labels for line feature, edgel feature and parallelism tolerance was erroneous when calibration was used.
- Fix: a geometric ambiguity could have lead to an erroneous distance tolerance status.
- Fix: the results of edgel features were not flushed when no new edgel features were found in the following image.
- Fix: the result of the coverage of a circle was not updated to the measured value.
- Fix: the result of the general status was not updated when the context did not contain any tolerance definitions.
- Fix: a geometric construction could lead to an ambiguity, resulting in erroneous min/max distance tolerance calculations.
- Fix: inner fit and outer fit of segment features have been improved when using metrology with calibration.
- Fix: the feature status was M_FAIL when the measured coverage value was equal to the min coverage value of the tolerance range.
- Fix: the maximum distance between a circle/arc and another feature could have lead to a non-unique solution.
- Fix: MmetStream(M_SAVE/M_LOAD/M_INQUIRE_SIZE_BYTE) lead to a failure when using an M_NULL system ID.
- Fix: MmetGetResult applied to a feature that was not found lead to a fatal failure.
- Fix: MmetCalculate applied to an M_NULL context ID lead to a fatal failure.
- Fix: an MmetGetResult call with invalid parameters gave invalid error messages.

2- Code Reader module

- Fix: when a timeout was reached, the status of a Data Matrix read operation was erroneous.
- Fix: the position estimation of MicroPDF417 codes was erroneous.
- Fix: the position estimation of Maxicode codes with the adaptive threshold method was erroneous.
- Fix: the threshold value was not updated when reading a maxicode with a user defined threshold value.
- Fix: reading a code at 90 degrees with adaptive threshold was not working properly.
- Fix: the high accuracy position of an RSS code was not valid when the angle was outside the range of [0,90].
- Fix: the returned position for M_PDF417 and M_TRUNCATED_PDF417 code types was always 0.
- Fix: there was an offset in the position after a successful read of 2D code types when the presearch was enabled.
- Fix: the result of the cell size has been improved when reading discrete linear codes.
- Fix: the comparison of the user supplied string in McodeVerify has been fixed.
- Fix: the M_ECC_CORRECTED_NUMBER result was erroneous.
- Fix: McodeInquire of the default value of the M_POSITION_ACCURACY control returned M_DEFAULT instead of M_LOW.
- Fix: when reading a code with angle detection and high position accuracy mode, the position result was not calculated properly in rare situations.
- Fix: McodeVerify of matrix codes (Data Matrix & QRCode) had a bad Unused ECC grades value in rare situations.
- Fix: McodeRead and McodeInquire on a same code context could not be used safely in different threads.

3- Calibration module

- Fix: a floating-point overflow could have occurred when an M_INVALID_POINT was present.
- Deprecation: M_OUTPUT_COORDINATE_SYSTEM has been renamed to M_OUTPUT_UNITS.
- Deprecation: M_CAMERA_POSITION_X/Y/Z controls are deprecated.
- Documentation: The former camera coordinate system has been replaced by tool coordinate system. The new camera coordinate system now refers to the coordinate system attached to the pinhole of the camera.

4- Blob module

- Fix: multiple calls to MblobCalculate with M_BLOB_IDENTIFICATION set to M_LABELED could have lead to erroneous blob results.
- Fix: the accuracy of blob features has been improved for large buffers.
- Fix: the results of contact points retrieved using MblobGetResult were inconsistent with the results retrieved from MblobGetResultSingle.

5- String Reader module

- Fix: internal fixes were applied to improve the reading of accentuated characters.
- Fix: in rare situations the module could have generated an invalid floating point operation.

6- Measurement module

- Fix: the edge strength result was erroneous.
- Fix: certain edges with very low contrast were ignored when they should not have been.
- Fix: edge extremity results will now be returned in an order consistent with the measurement box orientation.

7- Edge Finder module

- Fix: in rare situations, computation of the equivalent IIR filter could have lead to an invalid memory access.
- Fix: a potential division by 0 in the ellipse fit has been fixed.
- Fix: MedgeGetNeighbors with M_SEARCH_RADIUS_MAX control set to M_INFINITE was not properly interpreted.
- Fix: in rare situations, MedgeCalculate could be slowed down by inefficient memory access.

8- Registration module

- Fix: in some situations an overflow could have happened which affected the robustness of MregCalculate.
- Fix: a potential memory leak in MregCalculate has been fixed.
- Fix: failure to register one image prevented the registration of other images during a call to MregCalculate.

9- Model Finder module

- Fix: M_SEARCH_OFFSET_X/Y control was not properly updated when set to M_INFINITE.
- Fix: in extremely rare situations, MmodFind could have lead to an infinite loop.
- Fix: the weighted mask was not correctly supported when using M_GEOMETRIC_CONTROLLED.

10- OCR module

- Fix: MocrInquire did not properly support the M_TYPE_MIL_ID flag.
- Fix: a parameter verification issue when running in unicode has been fixed in the MocrVerifyString function.

11- Pattern Matching module

- Fix: MpatAllocResult with a large number of entries lead to a fatal failure.
- Fix: in rare situations the score result was lower than expected.
- Fix: MpatAllocAutoModel of small models lead to a fatal failure.

12- Mim and general processing functions

- Fix: the B component of BGR32 was lost in the MimArith(ADD) operation when the source and destination were in different types of memory (Host and Video).
- Fix: the accuracy of MimProject(M_90_DEGREE) has been improved for large 8-bits buffers.
- Fix: an internal rounding of the user-supplied MinIntensity threshold value in MimLocatePeakId has been corrected.
- Fix: MimFree lead to a crash when called with an invalid id.
- Fix: MimAllocResult() with the M_DEFAULT flag as the second parameter caused a "Not enough host memory to allocate buffer" error.
- Fix: MbufExportSequence was not working properly when dealing with images of different sizes.
- Fix: MimConvolve with a separable kernel and transparent overscan has been improved.
- Fix: MbufBayer(M_ADAPTIVE) saturation has been fixed.
- Fix: MimArith(M_NOT) has been fixed for small buffers whose size x equals 8.

13- Interactive utilities

- Fix [All]: under some conditions, the selection of results was not working properly.
- Fix [Edge]: the update of drawing annotations was sometimes done in the wrong display.
- Fix [Metrology]: drawing annotations were not updated properly when a sub-item was selected.
- Fix [Metrology]: controls for missing features have been added.
- Fix [Metrology]: an incorrect behavior which occurred when drawing and zooming the active edgels in the template display buffer has been fixed
- Fix [Metrology]: the addition of a new feature or tolerance was applied

to all open contexts.

- Fix [Metrology]: the behavior of the scrollbars for the template display buffer has been improved.
- Fix [StringReader]: the "Operations" group was not updated properly

4. Mim functions and performance optimizations

- General optimizations of the image processing functions.

5. Board specifics

- New [GPU]: GPU accelerations of image processing operations (For more information, see the GPU readme file).
- New [Helios]: MbufBayer with M_AVERAGE_2X2 has been optimized for the Matrox Helios.
- New [Helios]: MimLocatePeakId has been optimized for the Matrox Helios.
- Fix [Helios]: a potential memory leak has been fixed when using the blob module for the Matrox Helios system.
- New [Helios]: MimArithMultiple(M_OFFSET_GAIN) is accelerated by the Matrox Helios for 8 and 16 bits buffers.
- New [Odyssey]: MbufBayer with M_AVERAGE_2X2 has been optimized for the Matrox Odyssey.
- New [Odyssey]: MimLocatePeakId has been optimized for the Matrox Odyssey.
- New [Odyssey]: MimConvert with matrix conversion has been optimized for the Matrox Odyssey.
- New [Odyssey]: MimArithMultiple(M_OFFSET_GAIN) is accelerated by the Matrox Odyssey for 8 and 16 bits buffers.
- Fix [Odyssey]: MimResize has been improved for the Matrox Odyssey.
- Fix [Odyssey]: calibration propagation mechanism has been fixed for the Matrox Odyssey.