Release Notes Evoke 1.4.0

Exported on 7th April 2021

Table of Contents

1 New Features

1.1 Object Evaluation

The Evoke tracking algorithm requires all objects to have a unique pattern of markers (with the exception of template objects in a proximity group). This allows Evoke to start tracking the same frame an object becomes visible, which is particularly helpful in cases of frequent occlusion. However, the uniqueness requirement can be difficult to fulfill when considering all possible orientations of the objects, particularly when you have a large number of them.

The new object evaluation feature compares each object enabled in the **Tracking** panel with every other enabled object, showing a warning when there is a match between a pair of objects. The warning tooltip indicates which other object (or objects) it may be confused with, and the match distance (in millimeters). The match distance is the approximate average distance the object markers would have to move to exactly match the other object (or part of it). A match distance of 0 indicates the object pair is identical (or to be more precise, the fraction of markers specified by the object's entrance threshold is identical to a pattern of markers on the other object).

Entrance threshold has a significant effect on object evaluation. The entrance threshold specifies the fraction of object markers that must be detected in order to start tracking. If the entrance threshold is low it is easier to start tracking the object, but more likely that the tracking can be initialized incorrectly. For example, if you have a 10-marker object with an entrance threshold of 0.3 or below, tracking can start as soon as any 3 markers are detected in the correct pattern (3 is the minimum number required to track). This means that the object could be confused with any other object that has the same or similar pattern of 3 markers, regardless of how different they are as a whole. In general, it is recommended to set entrance threshold to 1 (require all object markers), unless the object geometry makes it difficult for the system to detect all the markers simultaneously. This may be the case for an object resting on the ground for instance; any markers on the underside may not be visible to the cameras.

This feature has several preference options (see **Settings | Preferences... | User | Subject Store**):

- Enable object evaluation - if enabled, object evaluation runs automatically in the background when any object is loaded or changed
  - This is enabled by default, but as the computational cost can be high we recommend it is disabled while running a live experience
- Object evaluation step limit - this controls the maximum amount of work done per object pair comparison
  - Increasing the step limit may allow the evaluation to find more potential problems, at the cost of increased computation time
- Object confusion warning threshold (mm) - a warning will be shown if the match distance between a pair of objects falls below this threshold
  - You may wish to adjust this threshold, depending on the accuracy of your volume calibration and the size of your objects

1.2 Improvements to the System panel

The **System** panel has been improved for 1.4, making it more responsive and providing more helpful feedback, especially for large systems.

Device category headers are now always visible, and include a summary of device status and warnings. For Pulsars, this includes battery charge level and number of devices on charge.

System and device parameters have been streamlined to remove some unused options. System file settings have moved to the top menu bar, next to view file settings. It is now possible to remove disconnected or selected devices from the system.

Most application configuration files are now stored in JSON format rather than XML. Configuration files created in Evoke 1.3.x or earlier will be converted on import into Evoke 1.4.

1.3 Improvements to the Processing panel

The **Processing** panel now offers a search facility and other controls that determine the visibility of the component properties.

Many parameters have been moved to the 'Advanced' section, leaving only the most important in the default view (click 'Show Advanced' to see all parameters). Proximity grouping parameters have also been moved to their own category.

Some parameter names have been revised (functionality is unchanged):

- In the 'Object Tracking' section:
    - 'Minimum Object Marker Separation' is now called 'Minimum Object Marker Separation (mm)'
    - 'Halve Processing Rate Post Centroid Tracking' is now called 'Halve Processing Rate'
- In the 'Camera Healing' section:
    - 'Enable Auto Heal Bump' is now called 'Enable Auto Bump Healing'
    - 'Force Auto Bump Healing Enabled with Strobes' is now called 'Enable Healing with Strobes'
- In the 'Reconstruction' section:
    - 'Minimum Cameras to Start Trajectory' is now called 'Unlabeled Recon. Min. Cams'
    - 'Enabled' is now called 'Enable Unlabeled reconstructions'

1.4 Calibrate Selected Cameras

The **Camera Calibration** panel now has an additional option for wand waves. 'Start Wave (All)' will start collecting wand wave data for calibrating all cameras, as in previous versions. In advanced options, there is now an additional option to 'Start Wave (Selected)', which will collect data to calibrate only the selected cameras.

1.5 Camera Calibration Improvements

Improvements have been made to the initial calibration of pairs of cameras, and subsequent merging of sets of cameras.

This increases the robustness of camera calibration, fixing occasional failures to calibrate in cases where there is limited overlap in the camera fields of view. This is particularly helpful in larger systems, with many cameras and lower density of camera coverage.

1.6 Load Subjects Via .mcp File

For an easy way to quickly load subjects and props from an existing scene, you can now load an *.mcp* capture file directly into Evoke.

To quickly load subjects and props from an .mcp:

1. At the top of the **Tracking** panel, click the **Load tracking configuration** button .
2. From the Load Tracking Configuration dialog box, select or browse to the required *.mcp* file.
   The objects and characters are displayed in the Tracking panel as normal.

1.7 Vantage+ Compatibility

Evoke 1.4 supports the use of the Vantage+ firmware upgrade (Firmware 725 and later), enabling you to use **High Speed** mode with your Vantage cameras without having to change the field of view (FOV) or lens. When you capture optical data, subsampling (selectively reducing the pixel count) enables you to run at high camera frame rates without reducing the FOV (frame size).

Previously, to run Vantage cameras at higher frame rates while maintaining their maximum resolution, windowing was used, which reduces the size of the FOV.

Now, in new **High Speed** mode, you can run your Vantage cameras at higher frames rates while maintaining the FOV. You can change frame rates during capture and you don't need to set up your cameras again when you increase the frame rate, as the FOV is unchanged.

Note that because the higher speeds are achieved through subsampling (removing some pixels from the frames), some reduction in resolution is incurred.

2 Changes

- Entrance threshold is now 1 by default, meaning that all object markers must be detected in order to start tracking an object
  - This improves reliability of tracking in general by reducing the likelihood of confusion between objects
  - For some objects it may not always be possible to see all markers (e.g. a ball when lying on the floor)
    - In these cases, entrance threshold should be set to a lower value
- Frame numbers are now counted from 1 (rather than 0) for the first frame
  - This is a display change, the wire format for DSSDK output is unchanged (counting from 0)
- Smart objects are automatically excluded from tracking if they are linked to connected device that is on charge, and the 'Markers On While Charging' parameter is disabled
  - It is still recommended to disable smart objects that aren't in use or are otherwise not trackable, but this relatively common case is now handled automatically
- The Camera Calibration 'Set Origin' tooltip now gives more information about the reason when the operation can't be completed
- The Camera view filter option 'Show Orientation' now has 'Snap to 90°' enabled by default
- Evoke Python API can now be installed directly from the Evoke install folder (`C:\Program Files\Vicon\Evoke1.4\SDK\Python` by default)
- Vicon terminology updates
  - The 'Master' device is now referred to as the 'Sync Source' device
  - The Beacon 'Master ID' parameter is now the 'Host ID' parameter
  - The Beacon 'whitelist' is now referred to as the 'allow list'
  - The Pulsar 'Pairing Master Device ID' is now the 'Pairing Host Device ID' parameter

3 Fixes

- Fixed 'Auto Bump Heal' always attempting to bump heal in order of device ID, which meant that it was sometimes impossible to heal cameras with adequate data because there was a lower-numbered camera that couldn't be healed
- Fixed preference to disable the recovery confirmation dialog when auto-saved settings are detected on start
- Fixed Evoke closing if Control app authorization is required on start
- Fixed 'Do not show again' option on out of date version warning dialog
- Improved performance when changing object parameters, particularly when characters are also loaded (this was especially apparent when changing parameters rapidly via API)
- Evoke now always reports Pulsar commands queued via API immediately, rather than allowing an indeterminate period where smart objects had not yet queued parameter changes to their linked devices
- CharacterFromClustersServices.SetSlotObject now accepts an empty string as the object name, which may be used to clear slots
- LogServices.Log now properly handles user-defined log categories
- RadioDeviceServices.ConnectionRefusedList now returns Result.Ok and an empty list if there haven't been any refused connections, rather than an error code

4 Evoke API 1.7.3

- Added object evaluation services.
- Added support for object confusion warnings to BasicObjectServices and SmartObjectServices.
- Added CameraCalibrationServices.StartWandWaveSelected.
- RadioDeviceServices: deprecated RadioWhitelist*, added equivalent RadioAllowList* functions. Deprecated MasterId, added equivalent HostId function.
- Added sample code for Pulsar communication and command synchronization

5 Pulsar Reprogramming Tool 1.0.5

Pulsar firmware 727 is now included in the installer:

- Fix for Pulsars that stop connecting to a Beacon when the Beacon is rebooted while the Pulsar is connecting
- Fix Pulsars failing to retry to connect to a Beacon when first attempt fails
- Fix issue that prevented Pulsars connecting to Beacons within a certain MAC address range