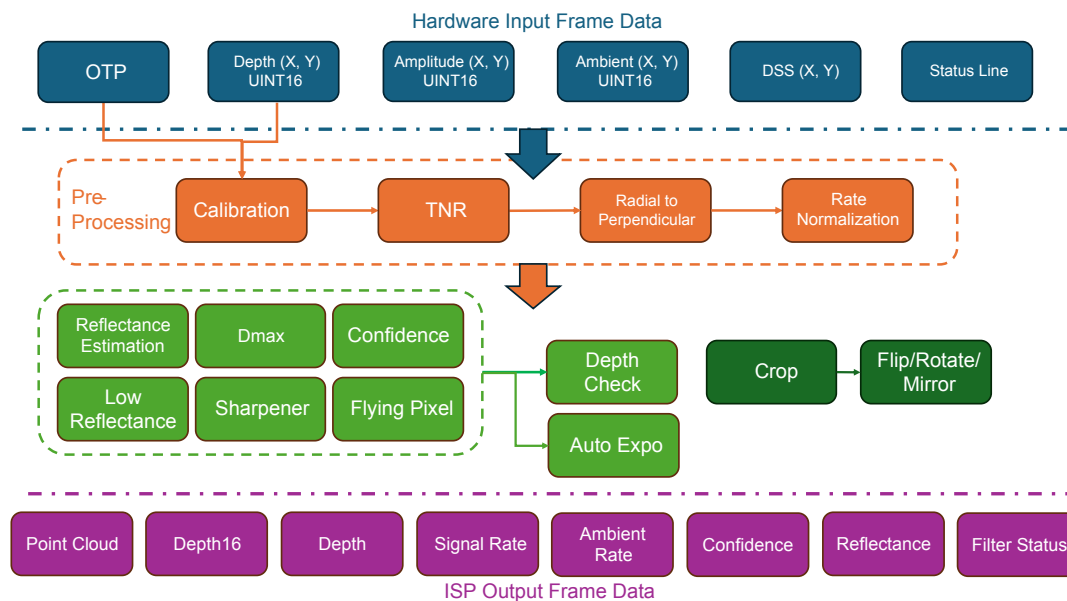


VL53L9CX postprocessing pipeline

Introduction

This document describes the postprocessing pipeline dedicated to the VL53L9CX device. Each algorithm block is described, along with the exposed parameters that can be tuned.

Figure 1. Pipeline overview



General tuning philosophy:

- Tune thresholds, enable flags, and operating modes first.
- Do not tune optical models, OTP-derived maps, or normalization math unless calibration owners explicitly request it.
- Parameters not listed in the *Exposed parameters* section can be modified. However, modifying the parameter is not the appropriate lever for tuning the image quality.
- Validate changes in both static scenes and motion scenes. Adjusting one tuning parameter may improve one aspect of system performance while degrading another.

1 Acronyms and abbreviations

Acronym	Definition
DSS	Dynamic SPAD selection
dToF	Direct Time-of-Flight
Float32	Floating-point over 32 bits
FMT	Final module testing
FP	Flying pixel
ISP	Internal signal processing
LSB	Least significant bit
LUT	Lookup table
MSB	Most significant bit
OTP	One-time programmable memory
R2P	Radial to perpendicular
S8	Signed int over 8 bits
S16	Signed int over 16 bits
S32	Signed int over 32 bits
SC	Step C
SF	Step final
SNR	Signal-to-noise ratio
SPAD	Single photon avalanche diode
Sy.x	Signed float encoded over x+y bits using the x bit for the floating value and y bit for the integer value
TNR	Temporal noise reduction
ToF	Time-of-Flight
U8	Unsigned int over 8 bits
U16	Unsigned int over 16 bits
U32	Unsigned int over 32 bits
Xtalk	Crosstalk

2 Input/output data format

2.1 OTP (one-time-programmable) data

This section details the information stored in OTP for the postprocessing calibration algorithm.

Figure 2. OTP calibration algorithm

Global Offset	0x06EC – 0x06ED
Per Pixel Offsets - Part 1	0x06F0 – 0x0AA0
Per Pixel Offsets - Part 2	0x0B28 – 0x0C62
Per Pixel Offsets - Part 3	0x0D18 – 0x0ED0
Defect SPAD	0x0AE0 – 0x0B07
Optical Offset	0x0B08 / 0x0B09
Residual Offset	0x0B0A / 0x0B0B
Bar Code	0x0C6C – 0x0C7F
Lens Calibration Data	0x0C80 – 0x0CA8
Traceability Code	0x0CE0 – 0x0CE7
I2C Slave ID	0x0CE8
I3C Instance ID	0x0CEC
Customer Section	0x0CF0 – 0x0D17
DSS_EFF_SPAD Context 1	0x0624 – 0x0337
DSS_EFF_SPAD Context 2	0x06A4 – 0x03B7

Optical centering data

The device provides lens-centering information.

- Optical offset information (X, Y), encoded in S2.0 (unity macro-SPAD) is provided because the hardware uses the information to center the receiver array during the exposure.
- Residual offset information (X, Y), encoded in S7.1 (unity SPAD) is used by the radial to perpendicular algorithm (refer to [Section 6.1](#))

Distance offsets

To compensate for internal delays per zone, the offset calibration is stored in the OTP.

- Global offset, encoded in S16 (mm)
- 2268 per zone offsets, encode in S6 (mm)

Lens calibration

The lens signal is not spread homogeneously over the array. The amplitude calibration map and associated offset are stored in the OTP and used by the rate normalization algorithm (refer to [Section 7: Rate normalization](#)).

- Amplitude calibration map (7x5) encoded in U8
- Amplitude scaler encoded in U8
- Calibration distance encoded in U16
- Calibration exposure encoded in U16 (number of shots of final exposure)
- Calibration reflectance encoded in U8 (%)

DSS configuration

The device provides the DSS lookup table (LUT), the efficiency table associated, and the dimensions of the tables.

Customer section

The customer section is a reserved bank for customers to store any relevant information required by the application. This 40-byte OTP memory can be programmed by the customer and read back by the application.

2.2

Status line

The status line is 100 bytes of metadata streamed with frames. This metadata contains dynamic information linked to the current frame. Check the datasheet (DS14879) for the full status line description. Below is the information required by the postprocessing library:

- Frame width/height
- Mode (Ranging/Intensity)
- Format (Wide/Square)
- Power mode (Regular/Low power/Ultralow power)
- Ambient attenuation
- Number of steps (6 or 7 steps)
- Current context (Context 1 / Context 2)
- Binning (2 / 4 / 8 / 12 / 16 / 24)
- DSS mode (Disable / Short / Long)
- Frame period
- Crop status (Enable/Disable, X/Y start, X/Y size)
- Number of shots (Step 1 / Step 4 or 5 / Step C / Step A)

2.3 Input frame data

Four frames with the hardware resolution are streamed out from the sensor.

- **Depth:** radial distances encoded in U15 (mm) for each zone. The MSB (most significant bit) gives extra information about the distance measured. 0 means that the distance is measured during the STEP4, while 1 means that the distance is measured during the STEP5.
- **Amplitude:** signal amplitude encoded in U16 for each zone
- **Ambient:** IR ambient encoded in U16 for each zone
- **DSS (dynamic SPAD selection):** DSS LUT index for each zone encoded on 4-bit, this is used to compute the amplitude and ambient rate

2.4 Output frame data

Depending on the applications and needs, the customer can output different frames depending on the selected options. The following section describes the different maps and formats. All maps are presented to the customer with a resolution corresponding to the cropping configuration. The cropping is either hardware (static configuration) or software.

Point cloud: The data is a list of points. The format for each point is 4 Float32 corresponding to the coordinates of the point (X, Y, Z) and the confidence. For more information about confidence encoding, refer to [Section 18: Point cloud](#).

DEPTH16: The main objective of the DEPTH16 is to minimize the volume of data to transmit.

The depth is encoded in millimeters over 13 bits and confidence on 3 bits (MSB). Due to the limitation of 13 bit the maximum distance is reduced $2^{13} - 1 = 8191$ mm. For more information about DEPTH16 encoding, refer to [Section 17: Depth16](#).

Depth: Distance encoded in millimeters in Float32.

Signal: Signal rate is encoded in count per second in Float32.

Ambient: Ambient rate is encoded in count per second in Float32.

Confidence: Confidence encoded in Float32.

Reflectance: Reflectance estimation is encoded in percentage in Float32.

Filter status: Filter status is encoded in Float32. Value 0 means the data is valid. Each bit gives the filter status information:

- Bit 0: Confidence filter
- Bit 1: Reflectance filter
- Bit 2: Sharpener filter
- Bit 3: Radial to perpendicular filter
- Bit 4: Flying pixel filter

Recommendations

When streaming more than one map is acceptable in terms of FPS, the following configuration is recommended:

- **Depth map** to provide depth information
- **Filter status map** to provide data validity information
- **Reflectance map** is optional, if useful for the application

When streaming only one map is required to maintain high FPS, the following configuration is recommended:

- **Depth16 map:** It contains both depth and confidence information
- **Reflectance map** is optional, if useful for the application

3 Overall pipeline configuration

3.1 Algorithm overview

The pipeline configuration:

- Defines setup-level flags that describe the hardware configuration and select the correct ISP behavior before block-level tuning starts.
- Groups the global ISP controls that enable, disable, or bypass whole processing stages.

3.2 Technical details

These parameters are not local image-quality adjustments for a single block. They are global ISP controls that describe the product setup or decide whether full algorithms stay active in the pipeline.

A wrong setting here can bias the behavior of later blocks even if their own thresholds are well tuned.

In particular, the cover glass presence flag changes how confidence is evaluated and which threshold adaptation path is relevant.

The bypass controls are listed here rather than in each algorithm section because they are pipeline-configuration choices, not normal tuning parameters.

3.3 Exposed parameters

Exposed params:

- `cover-glass` (default: `False`):
 - **type**: boolean.
 - **role**: declares whether the hardware stack includes a cover glass in front of the sensor.
 - **false**: use the nominal `no-coverglass` confidence behavior.
 - **true**: enable the `coverglass-aware` confidence behavior so the confidence computation and threshold adaptation account for the additional crosstalk risk.
 - **impact**: when enabled, confidence becomes more conservative in zones affected by cover glass crosstalk, which usually reduces false-valid detections at the cost of some coverage.
 - **Caution**: this should reflect the physical hardware configuration; it must not be used as a tuning shortcut.
- `bypass-tnr-algo`:
 - **enable**: bypasses the temporal filter, so no temporal noise reduction or flicker correction is applied.
 - **disable**: keeps the temporal filter active.
 - **risk**: Output usually becomes noisier and less stable from frame to frame.
- `bypass-r2p-algo`:
 - **enable**: bypasses the radial-to-perpendicular conversion and keeps the raw radial depth path.
 - **disable**: keeps the calibrated radial-to-perpendicular conversion active.
 - **risk**: Downstream geometric interpretation is no longer equivalent to the calibrated perpendicular output, and the PointCloud format is not available if bypassed.
- `bypass-r2p-filter`:
 - **enable**: keeps the radial-to-perpendicular conversion, but bypasses the perpendicular-distance clip validity check.
 - **disable**: keeps the radial clip active.
 - **risk**: Out-of-range values can propagate downstream and may later cause an issue.
- `bypass-refl-filter`:
 - **enable**: bypasses the reflectance filter, so pixels that would normally be rejected for implausible reflectance are kept.
 - **disable**: keeps the reflectance filter active.
 - **risk**: Unrealistic low-signal or extreme-reflectance pixels can remain in the output.

- `bypass-conf-filter`:
 - **enable**: bypasses the confidence filter, so pixels that would normally be rejected for insufficient confidence are kept.
 - **disable**: keeps the confidence filter active.
 - **risk**: Weak, unstable, or false-valid detections can reappear, especially under low signal, strong ambient, or crosstalk conditions.
- `bypass-fp-filter`:
 - **enable**: bypasses the flying-pixel filter and keeps its validity rejection out of the pipeline.
 - **disable**: keeps the flying-pixel filter active.
 - **risk**: Edge artifacts, mixed pixels, and isolated outliers can reappear.
- `bypass-sharpener-filter`:
 - **enable**: bypasses the sharpener filter, so pixels that would normally be rejected for blur or veiling-glare contamination are kept.
 - **disable**: keeps the sharpener filter active.
 - **risk**: Glare halos, blurred edges, and false-valid pixels can reappear.

Recommended use:

- Set this once during product or board initialization.
- Keep it aligned with the real optical stack of the target hardware.
- If the hardware has a cover glass, enable it before tuning confidence-related thresholds such as `conf-threshold-scaler` or `conf-threshold-scaler-cover-glass`.
- Use the bypass controls mainly for debug, comparison, or deliberate system-level experiments, not as substitutes for proper algorithm tuning.
- On platforms where the computing budget is tight, these bypass controls can also be used as system-level FPS levers, with the largest performance impact usually coming from `bypass-tnr-algo`, then `bypass-sharpener-filter`, then the other bypasses.

4 Calibration

4.1 Algorithm overview

Description: The calibration function makes it possible to correct internal delays that result in measurement errors.

Key features: As the available memory in OTP is limited, the offset map is stored in S6 (Signed integer on 6 bits) across three different memory banks. It is necessary to read and decode the 2268 per-pixel offsets (42x54) as well as the global offset, stored in S16 (signed integer over 16 bits). All offsets are specified in mm.

Applications: The algorithm reduces the distance error. The correction is applied on radial distance prior to the perpendicular conversion.

4.2 Exposed parameters

No tuning is expected; the calibration information is extracted in production.

5 TNR

5.1 Algorithm overview

The temporal noise reduction (TNR) module reduces temporal noise. It improves the stability of depth, amplitude and ambient measurements over time, while preserving real scene motion and minimizing flicker. This module also increases maximum distance ranging using increased confidence.

Per pixel, the module maintains up to two “candidates”, each representing a stable temporal solution (for example, short-range vs long-range or two alternative depths in flickering conditions). Each candidate stores filtered depth, amplitude, ambient, short/long flag, effective SPADs, a temporal counter and a flicker score.

The module combines three mechanisms:

1. **Temporal noise reduction (TNR):**

In static areas, new frames that are consistent with an existing candidate are temporally averaged with its history. The TNR counter increases and the temporal standard deviation decreases until a configurable maximum gain is reached.

2. **Flicker correction:**

When two candidates alternate (for example due to multipath, low signal-to-noise ratio (SNR) or unstable reflections), the module uses per-candidate scores, counters and exposure flags to choose a stable output and avoid visible toggling between them.

3. **Motion detection and reset:**

When motion is detected, the temporal history is reset locally to ensure fast convergence to the new state and avoid motion trails.

TNR and flicker correction can be enabled or disabled independently. Amplitude filtering and short/long behavior are configurable to match the system integration needs.

5.2 Exposed parameters

This section summarizes the parameters that impact behavior and how to tune them. Tuning is only needed for specific usage profiles (for example very dynamic scenes, very low SNR, or specific flicker conditions).

1. Global control

- **Overview:**
- Use this parameter section to enable/disable temporal processing and to reset the internal history when the operating mode changes.

Caution:

- *If TNR is disabled (bypass = True), no temporal noise reduction or flicker correction is applied, and the output may exhibit significantly higher frame-to-frame noise.*
- *If the memory is reset too often (resetMemory frequently triggered), the module cannot accumulate history, and the effective temporal gain is reduced.*

- `tnr-reset-memory`
 - **Role:** Reset temporal history and candidates.
 - **Use:** Use on mode changes or after large configuration changes (for example major exposure reconfiguration).

2. TNR parameters

- **Overview:**
- Use this parameter section to balance temporal smoothing versus responsiveness, and to control how aggressively frames are averaged over time.

Caution:

- *If temporal filtering is too weak (low maxCounter, low stdFactorDist/Amp), the noise reduction is limited, and residual temporal noise may remain visible in static scenes.*
- *If temporal filtering is too strong (high maxCounter, high stdFactorDist/Amp), the output may react slowly to actual scene changes and may temporarily “lag” behind fast depth or amplitude variations.*

- `tnr-max-counter` (default: 16)
 - **Role:** Sets the maximum effective temporal gain (beyond this, additional frames bring limited extra noise reduction).
 - **Effect:**
 - Lower → less smoothing, faster reaction to changes.
 - Higher → stronger smoothing in static areas, but slower reaction to subtle changes.
 - **Example use:** reducing maxCounter can help when the sharpener is disabled and glare becomes visible because TNR converges on the glare over time.

6 Radial to perpendicular

6.1 Algorithm overview

- Converts radial depth into perpendicular depth.
- Generates X/Y coordinates and point-cloud geometry.
- Applies radial distortion and optional parallax correction.

6.2 Technical details

- The module uses the calibrated optical center, effective focal length, crop position, binning, and distortion model to convert the measured radial depth into perpendicular depth.
- It computes the corrected Z distance on the optical axis and also generates distortion-corrected X and Y coordinates for point-cloud output.
- If parallax correction is enabled, the module applies an additional geometry correction to compensate Rx-Tx separation before finalizing the coordinates.
- It also outputs a validity map based on maxDistance, so zones beyond the allowed perpendicular range are marked invalid.
- If bypass is enabled, the geometric conversion is skipped and the host keeps the raw radial depth path for debug or custom processing.

6.3 Exposed parameters

No tuning is expected; the calibration information is extracted in production.

7 Rate normalization

7.1 Algorithm overview

The rate normalization module processes a **reference amplitude map** to compute the signal and ambient rate maps.

The normalization process includes three key steps:

1. A reference amplitude map is used for reflectance and signal uniformity.
2. **Ambient rescaling**: Correcting the ambient map for attenuation.
3. **Rate map computation**: Generating normalized signal and ambient photon rate maps.

7.2 Exposed parameters

No tuning is expected; the calibration information is extracted in production.

8 Reflectance

8.1 Algorithm overview

The **reflectance estimation algorithm** leverages the capabilities of a dToF (direct Time-of-Flight) sensor to compute a reflectance map of a scene. The sensor provides both **amplitude** and **depth** maps, which are used in combination with a **reference amplitude map** to estimate the Lambertian reflectance of objects in the scene.

The algorithm outputs:

- **Reflectance map:** A pixel-wise estimation of Lambertian reflectance.
- **Validation status:** Flags to indicate whether reflectance values are below a low threshold or above a high threshold, ensuring the reliability of the reflectance data.

This process is crucial for applications requiring accurate material property estimation, object recognition, or scene understanding. In addition to the sharpener algorithm, the low reflectance filter can be used to reject glare.

8.2 Exposed parameters

- `minimum-reflectance:`
 - **increase:** more low-signal, dark, or distant zones are rejected. Output becomes cleaner.
 - **decrease:** more dark targets pass and coverage increases.
 - **risk:** If set too low, it keeps an unrealistic low signal target. If set too high, it removes valid low reflectance targets too aggressively.

Recommended use:

- Tune `minimum-reflectance` to adjust minimum detectability of specified reflectance and to reject unwanted glare.
- If the sharpener is bypassed for some reason, this filter can be used as a first filter for any glare from objects.
- Leave the reflectance model unchanged unless characterization proves it is incorrect.

9 Confidence

9.1 Algorithm overview

The ISP provides the functionality to compute a confidence metric for use by the device, with or without cover glass. This functionality is used to determine if each pixel has enough signal compared to the current configuration (ambient, exposure, and so on).

Each confidence value is compared to the relevant threshold map generated based on final ranging conditions. The default thresholds are designed to ensure a false-positive rate of $\leq 1\%$. Default thresholds and scalers can be adjusted by the user as detailed below.

9.2 Exposed parameters

The confidence model selection depends first on the global **cover glass** flag that depends on the hardware setup.

- `conf-threshold-scaler`: Scale factor applied to the confidence threshold (with or without cover glass). Increasing this value tightens the filtering (fewer detections, fewer false positives), while decreasing it relaxes the filtering (more detections, potentially more false positives).
- `conf-threshold-scaler-cover-glass`: Scale factor applied to the cover glass contribution in the confidence computation. It adjusts the strength of the cover glass correction: the higher the cover glass signal, the higher this scaler should typically be set to properly compensate its effect.

Recommended use:

- Confidence thresholding is typically properly tuned in the delivered ISP.
- In normal use, no change is needed unless there is a clear application-specific reason to rebalance detection versus false positive.

10 DMax

10.1 Algorithm overview

- Estimates the maximum detectable distance per pixel.
- This maximum distance is computed for a given reflectance specified by **dmaxReflectance**.
- Used as a replacement distance when confidence fails and **dmaxEn** is active to give information that no object was found below this theoretical distance (based on ambient condition and reflectance target specified).

10.2 Exposed parameters

- `dmax-reflectance`:
 - **increase**: Higher reflectance means that a theoretical target is brighter. This helps to estimate the maximum distance. Therefore, the Dmax increases.
 - **decrease**: Lower reflectance means a darker theoretical target for estimating maximum distance. As a result, that Dmax decreases.

Recommended use:

- Tune only the target reflectance based on application usage. For example, if a target of 10% reflectance minimum must be found, then set this as `dmax-reflectance`.

11 Flying pixel

11.1 Algorithm overview

- Detects isolated depth outliers and edge-mixing artifacts, especially near discontinuities.
- Produces a validity map for **DistanceCheck**.

11.2 Technical details

- The filter validates local depth consistency around each pixel and can add SNR-based rejection under weak-signal conditions.
- It mainly affects edge cleanliness, mixed-pixel behavior, and isolated outlier suppression.

11.3 Exposed parameters

Recommended use:

- No tuning available to user.

12 Sharpener

12.1 Algorithm overview

Description:

The sharpener algorithm identifies and marks false targets caused by veiling-glare.

The algorithm segments the image into groups of data. It analyzes the grouped data and gives a score to each data in the group. A score below 1 means the data is unreliable/corrupted and should be discarded.

Key features: The signal threshold factor scales with the amount of data to be filtered out.

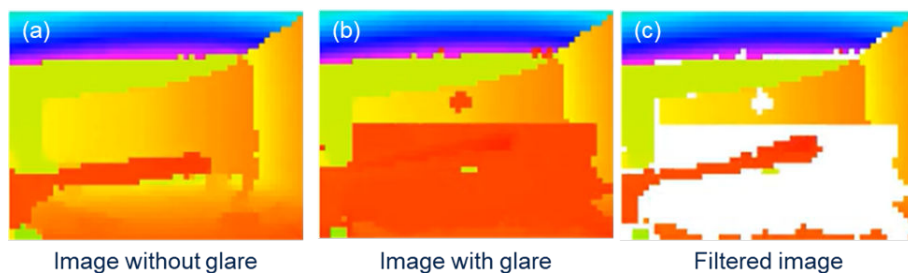
Applications: The algorithm is particularly effective at filtering out data corrupted by veiling glare.

This algorithm is internally tuned to minimize the number of pixels that introduce glare artifacts, even if that means discarding some pixels that can still be valid in some situations.

12.2 Technical details

The VL53L9CX can reduce the glaring effect of an aggressor, such as a very bright object in the field of view. Glare causes objects returning low signals to be corrupted, and returns the same distance as the aggressor.

Figure 3. Glare examples



The images above are examples of a scene with a two-sided object: one side is highly reflective. The object is in front of two screens in an open-space office. Image (a) shows that the nonreflective side does not cause glare. In image (b) the reflective side is an aggressor and causes significant glare. In image (c) the sharpener filters out the corrupted pixels.

By making groups of pixels using the measured depth, the aggressor is grouped with the corrupted pixels. The glare is modeled using a Gaussian shape that adapts to the population in the group. The VL53L9CX compares the amount of signal of an object in the scene with the glare caused by the main aggressor of the group. If the score is over 1, then the signal received is that of an actual object and not veiling glare from the aggressor.

Recovery mode allows for pixels to be restored if they are likely to correspond to a real object rather than pure glare.

12.3 Exposed parameters

This section outlines the key parameters used to adjust the behavior of the sharpener filter. Each parameter influences how the filter processes image data, allowing fine control over the balance between preserving good pixels and removing pixels in an area affected by veiling glare.

- `sharp-optimization:`
 - **optim:** default mode with larger grouping range threshold.
 - **double_sharp:** enables the dedicated double-sharp mode. It reduces the sharpener grouping range thresholds to 200 mm; it is a more specific mode for close double-edge and can improve target detection when a bright object and the real target are close in distance.
 - **risk:** double sharp mode is more precise in terms of segmentation but at the cost of time and memory.
- `sharp-signal-threshold-factor:`
 - **default:** 0.05.
 - **increase:** more pixels are treated as weak relative to the dominant group signal, so more blur or glare is flagged.
 - **decrease:** fewer pixels are rejected and more coverage is preserved.
 - **first tuning:** if the sharpener discards too many pixels, tune this parameter first before changing sharpener mode, sigma, or recovery settings.
 - **practical guidance:** 0.03 has been tested as a useful relaxation point that usually restores valid pixels without keeping too much glare, but it still keeps more glare than the default value and the customer must be aware of that tradeoff.
 - **extreme case:** 0.01 is very risky. At this value, glare is kept almost every time it occurs, while valid pixels are almost no longer removed by the sharpener.
 - **risk:** a value set too high creates holes or over-filtering around targets. Too low leaves halos and blur leakage.
- `sharp-recovery-mode:`
 - **enable:** allows recovery of some pixels that would otherwise be discarded.
 - **disable:** keeps the sharpener in a stricter reject-only behavior.
 - **risk:** enabling can reintroduce glare-contaminated pixels. Disabling can remove recoverable valid structure.
- `sharp-recovery-signal-threshold-factor:`
 - **default:** 4.0.
 - **increase:** recovery becomes more conservative, so fewer pixels are restored.
 - **decrease:** recovery becomes more permissive, so more pixels are restored.
 - **first tuning for recovery:** if recovery mode is enabled but not enough previously filtered pixels are restored, tune this parameter first.
 - **practical guidance:** the default value has been tuned to avoid recovering glare, but it may also fail to recover some pixels that were previously filtered even though they can be valid.
 - **aggressive case:** 1.0 is a very aggressive value. It should recover nearly everything that was previously filtered, but it can also recover some glare and must therefore be used with care.
 - **risk:** too high leaves unnecessary holes. Too low brings back glare artifacts.

Recommended use:

- This is one of the most useful application-tuning blocks.
- If too many pixels are discarded by the sharpener, start with `sharp-signal-threshold-factor` before tuning other sharpener parameters.
- If recovery mode is used, start with `sharp-recovery-signal-threshold-factor` before changing other recovery-related behavior.
- Tune together with reflectance and flying-pixel filtering because all three can remove edge or halo artifacts.

13 DistanceCheck

13.1 Algorithm overview

This module applies validity maps from various filters. They are processed in the following order:

1. Confidence
2. Reflectance threshold
3. Sharpener
4. R2P (theoretical distance clip)
5. Flying pixel

If a pixel is marked as invalid (validity != 0):

- if **dmaxCheck**= True, then if filtered by confidence, the depth value is replaced by dmax; but if filtered by R2P, the depth value is replaced by distance_clip
- if **dmaxCheck**= False and **replaceDistance**=False, then depth value is left as is
- if **dmaxCheck**= False and **replaceDistance**=True, then depth value is replaced by 12000

13.2 Technical details

Each pixel status is accumulated and if invalid, it specifies which filter marked the pixel as invalid by toggling a bit sequence from LSB (confidence) to MSB (n-th filter, currently flying pixel). The final filter status is the integer value of the binary inverse. This results in $2^n - 1$ integer filter codes.

The current filter codes are outlined in the following table.

Table 1. Filter status codes

Code	Description
0	Pixel valid (no filter)
1	Confidence
2	Reflectance
3	Conf+Refl
4	Sharpener
5	Sharp+Conf
6	Sharp+Refl
7	Sharp+Refl+Conf
8	R2P
9	R2P+Conf
10	R2P+Refl
11	R2P+Refl+Conf
12	R2P+Sharp
13	R2P+Sharp+Conf
14	R2P+Sharp+Refl
15	R2P+Sharp+Refl+Conf
16	Flying Pixel
17	FP+Conf
18	FP+Refl
19	FP+Refl+Conf
20	FP+Sharp

Code	Description
21	FP+Sharp+Conf
22	FP+Sharp+Refl
23	FP+Sharp+Refl+Conf
24	FP+R2P
25	FP+R2P+Conf
26	FP+R2P+Refl
27	FP+R2P+Refl+Conf
28	FP+R2P+Sharp
29	FP+R2P+Sharp+Conf
30	FP+R2P+Sharp+Refl
31	FP+R2P+Sharp+Refl+Conf

13.3 Exposed parameters

`dmax-check`: Bool indicates if an invalid value filtered by confidence and r2p must be replaced by the estimated distance:

- either by Dmax computed when confidence filtering (Note: dmax algo must not be bypassed)
- either by the r2p clip value when r2p filtering

`replace-distance`: Bool indicates if an invalid value must replace invalid data or not. If bool is false then the reported invalid distance is kept.

14 Auto exposure

14.1 Algorithm overview

- Predicts exposure based on two target profiles:
 - Range target: Find exposure to reach ranging conditions for a given reflectance.
 - Power target: Find exposure to maintain power limit.
- Uses DMax for range mode and signal/ambient rate for power mode. Target reflectance is specified in DMax configuration. Target range is specified in this module.

14.2 Technical details

- Its effect is global because it shifts signal level, noise behavior, motion sensitivity, and power consumption together.

14.3 Exposed parameters

- `smart-mode`:
 - **range**: exposure is adjusted to reach a target range.
 - **power**: exposure is adjusted to hit a power budget.
 - **risk**: the wrong mode optimizes the wrong system objective.
- `smart-target-range`:
 - **increase**: integration tends to increase and long-range coverage improves.
 - **decrease**: operation becomes faster and safer, but reach is reduced.
 - **risk**: too high increases motion sensitivity, saturation risk, and glare sensitivity. Too low underuses available range. The distance prediction is for a given dmax-reflectance.
- `smart-target-power`:
 - **increase**: the algorithm allows more exposure budget and usually more signal.
 - **decrease**: power is constrained more aggressively.
 - **risk**: too high wastes power and can worsen motion behavior. Too low reduces usable coverage.
- `set-integration-limits`:
 - **enable**: clamps exposure updates to the allowed integration-time window.
 - **disable**: lets the algorithm run without applying those host-side limits.
 - **risk**: disabling can allow settings that do not match the intended frame-rate or system envelope.
- `min-integration-time`:
 - **increase**: prevents very short exposures.
 - **decrease**: allows shorter exposures in bright scenes.
 - **risk**: too high wastes power or saturates bright scenes. Too low can reduce stability in some operating corners.
- `max-integration-time`:
 - **increase**: low-signal scenes can integrate longer.
 - **decrease**: exposure growth is limited earlier.
 - **risk**: too high increases latency and motion artifacts. Too low blocks useful low-light operation.

Recommended use:

- This is a strong system-level tuning block.
- Tune it together with confidence and TNR because exposure changes shift both noise and motion behavior.
- Do not use auto exposure to fix static range bias, and avoid forcing long integration when motion artifacts are already the limiting factor.

15 Crop

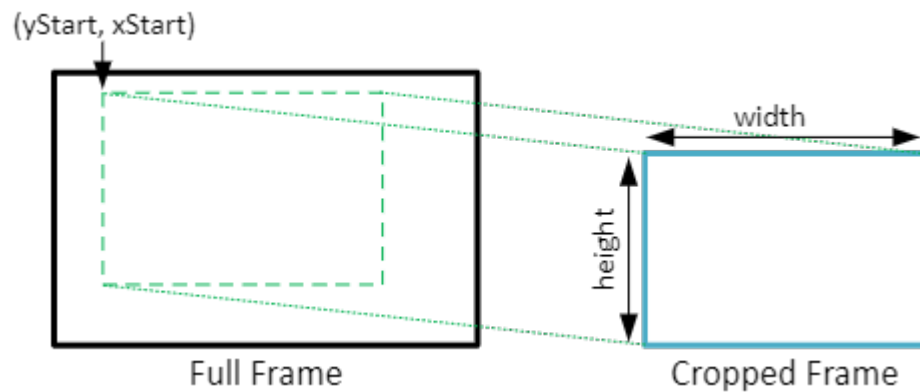
15.1 Algorithm overview

The crop section enables final maps prior output to be cropped to an arbitrary user-selectable subresolution. By specifying the final output height and width as well as the start coordinates in the form of (yStart, xStart). This crop applies to any relevant map available to the host.

15.2 Technical details

The crop is done by specifying the final output height and width as well as the start coordinates in the form of (yStart, xStart). This crop applies to any relevant map available to the host.

Figure 4. Crop illustration



15.3 Exposed parameters

`sw-crop-enable`: Bool to select if software crop needs to be done.

`sw-crop-x-offset`: Int of the x coordinate to start the cropping.

`sw-crop-y-offset`: Int of the y coordinate to start the cropping.

`sw-crop-x-size`: Int of the x cropped size.

`sw-crop-y-size`: Int of the y cropped size.

Note: The output dimension can differ from default dimension based on cropping selection.

16 Flip rotate mirror

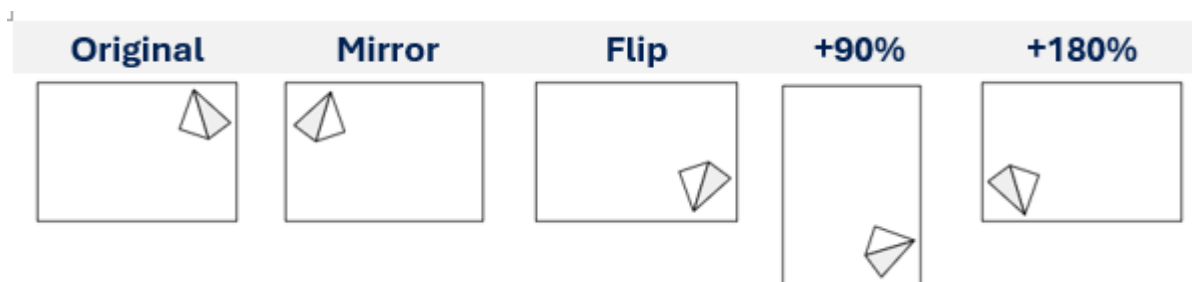
16.1 Algorithm overview

Flip, rotate, and mirror sections enable final maps prior output to be rotated, flipped or mirrored by 90° increment, although the model provides the capability to rotate at an arbitrary angle.

16.2 Technical details

This applies to any relevant map made available to the host. All maps which are cropped are also rotated.

Figure 5. Rotation and flip/mirror examples



16.3 Exposed parameters

Rotation: String that can be set to '+90', '-90' or '180'.

Flip: String that can be set to 'none', 'mirror' or 'flip'.

Note: The output dimension can differ from the default dimension based on rotation selection. For example, a rotation of 90° changes (x,y) dimensions to (y,x).

17 Depth16

17.1 Algorithm overview

The component Depth16 converts the final distance map and a 3-bit confidence version according to the Android DEPTH16 standard into the DEPTH16 format. It clips range values above 13 bits. In other words, values greater than 8191 mm are clipped to 8191 mm and marked as invalid in confidence.

This output is optional and must be explicitly enabled by the host.

17.2 Technical details

The confidence-mapping default is outlined in the table below and is computed alongside the main confidence and the per pixel confidence threshold C_{th} , which is internally available as a per pixel map confidence_thr from confidence. Other options are available and can be selected using the format variable. The default confidence mapping (close to android depth16 format definition) is the following.

Table 2. DEPTH16 confidence mapping

DEPTH16 3 bit code	Binary	Definition	VL53L9 Confidence
0	000	100%	$Conf > 12C_{th}$
7	111	85%	$10C_{th} \leq Conf < 12C_{th}$
6	110	70%	$8C_{th} \leq Conf < 10C_{th}$
5	101	60%	$6C_{th} \leq Conf < 8C_{th}$
4	100	45%	$4C_{th} \leq Conf < 6C_{th}$
3	011	30%	$2C_{th} \leq Conf < 4C_{th}$
2	010	15%	$C_{th} \leq Conf < 2C_{th}$
1	001	0%	$Conf < C_{th}$ or detection limit or filtered

"The confidence value is an estimate of correctness for this sample. It is encoded in the three most significant bits of the sample, with a value of zero representing 100% confidence, a value of one representing 0% confidence, a value of two representing 1/7, a value of three representing 2/7, and so on." (<https://developer.android.com/reference/android/graphics/ImageFormat#DEPTH16>).

Additionally, two mappings are provided. One provides information on DMax unique to ST's ToF sensors, and the other is completely customized to VL53L9CX as a template for custom mappings. These are shown in [Table 3. DEPTH16 confidence and mapping](#) and [Table 4. DEPTH16 custom mapping](#). [Table 3](#) shows the differentiation between bad confidence and any filter. [Table 4](#) makes further use of the *filterStatus* map with a custom filter encoding.

Table 3. DEPTH16 confidence and mapping

DEPTH16 3 bit code	Binary	Definition	VL53L9CX Confidence
0	000	100%	$\text{Conf} > 12C_{th}$
7	111	85%	$10C_{th} \leq \text{Conf} < 12C_{th}$
6	110	70%	$8C_{th} \leq \text{Conf} < 10C_{th}$
5	101	60%	$6C_{th} \leq \text{Conf} < 8C_{th}$
4	100	45%	$3C_{th} \leq \text{Conf} < 6C_{th}$
3	011	15%	$C_{th} \leq \text{Conf} < 3C_{th}$
2	010	Invalid (confidence or r2p)	$\text{Conf} < C_{th}$ or detection limit
1	001	Invalid	Filtered

Table 4. DEPTH16 custom mapping

DEPTH16 3 bit code	Binary	Definition	VL53L9CX Confidence
0	000	Valid	Valid
7	111	Invalid	Any other filter combination
6	110	Invalid	Sharpener + FP
5	101	Invalid	Sharpener + Refl
4	100	Invalid	FP
3	011	Invalid	Refl
2	010	Invalid	Sharpener
1	001	Invalid	$\text{Conf} < C_{th}$

17.3 Exposed parameters

`depth16-format`: Int to determine which of the three formats to use. The default value is 0 to select the default Android DEPTH16. The values 1 and 2 correspond respectively to [Table 2](#) and [Table 3](#).

18 Point cloud

18.1 Algorithm overview

The point cloud algorithm converts the depth map into a point cloud. Each pixel in the distance map is positioned in space using {X, Y, Z} coordinates, with the sensor center as the origin point.

Note: Disabling r2p does not work if point cloud output is enabled.

18.2 Exposed parameters

No tuning is expected; the calibration information is extracted in production.

Revision history

Table 5. Document revision history

Date	Version	Changes
14-Sep-2026	1	Initial release

Contents

1	Acronyms and abbreviations	2
2	Input/output data format	3
2.1	OTP (one-time-programmable) data	3
2.2	Status line	4
2.3	Input frame data	5
2.4	Output frame data	5
3	Overall pipeline configuration	6
3.1	Algorithm overview	6
3.2	Technical details	6
3.3	Exposed parameters	6
4	Calibration	8
4.1	Algorithm overview	8
4.2	Exposed parameters	8
5	TNR	9
5.1	Algorithm overview	9
5.2	Exposed parameters	10
6	Radial to perpendicular	11
6.1	Algorithm overview	11
6.2	Technical details	11
6.3	Exposed parameters	11
7	Rate normalization	12
7.1	Algorithm overview	12
7.2	Exposed parameters	12
8	Reflectance	13
8.1	Algorithm overview	13
8.2	Exposed parameters	13
9	Confidence	14
9.1	Algorithm overview	14
9.2	Exposed parameters	14
10	DMax	15
10.1	Algorithm overview	15
10.2	Exposed parameters	15
11	Flying pixel	16
11.1	Algorithm overview	16

11.2	Technical details	16
11.3	Exposed parameters	16
12	Sharpen	17
12.1	Algorithm overview	17
12.2	Technical details	17
12.3	Exposed parameters	18
13	DistanceCheck	19
13.1	Algorithm overview	19
13.2	Technical details	19
13.3	Exposed parameters	20
14	Auto exposure	21
14.1	Algorithm overview	21
14.2	Technical details	21
14.3	Exposed parameters	21
15	Crop	22
15.1	Algorithm overview	22
15.2	Technical details	22
15.3	Exposed parameters	22
16	Flip rotate mirror	23
16.1	Algorithm overview	23
16.2	Technical details	23
16.3	Exposed parameters	23
17	Depth16	24
17.1	Algorithm overview	24
17.2	Technical details	24
17.3	Exposed parameters	25
18	Point cloud	26
18.1	Algorithm overview	26
18.2	Exposed parameters	26
	Revision history	27

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved