



VisionLabs LUNA KIOSK

v.2.5.2

None

| | |
|---|----|
| 1. Glossary | 4 |
| 2. Introduction | 5 |
| 3. System requirements | 6 |
| 3.1 Minimum system requirements for x64 architecture | 6 |
| 3.2 Minimum system requirements for ARM architecture | 6 |
| 3.3 Supported cameras | 7 |
| 3.3.1 Connection requirements | 7 |
| 4. System architecture | 8 |
| 5. Algorithm | 9 |
| 5.1 Bestshot selection | 9 |
| 5.2 Monitor camera status | 11 |
| 6. Component description | 13 |
| 6.1 RSEngine component | 13 |
| 6.1.1 VisionLabs LUNA SDK component | 13 |
| 6.1.2 RealSense2 SDK component | 14 |
| 6.1.3 Component: VLS LUNA CAMERA 3D SDK | 14 |
| 6.1.4 Component VLS LUNA CAMERA 2D SDK | 15 |
| 6.1.5 Camera functions | 15 |
| 6.1.6 Camera Monitoring component | 18 |
| 6.2 RSE Server component | 19 |
| 6.3 WebSocket Client component | 24 |
| 7. System setup | 25 |
| 7.1 System setup on Windows | 25 |
| 7.2 System settings on Ubuntu 24.04 x64 and Debian 10 x64 | 25 |
| 7.3 Logging | 26 |

| | |
|---|----|
| 8. Switching camera types | 27 |
| 8.1 Quick start: Launch commands | 27 |
| 8.1.1 Changing the default camera | 27 |
| 8.2 Configuration differences | 28 |
| 8.2.1 Common settings | 28 |
| 8.2.2 IntelRealSense-specific settings | 28 |
| 8.2.3 VLSSLunaCamera3D-specific settings | 28 |
| 8.2.4 VLSSLunaCamera2D-specific settings | 28 |
| 9. Appendix 1: General configuration parameters | 29 |
| 10. Appendix 2. Status codes and error descriptions | 49 |

1. Glossary

| Term | Definition |
|-------------------------------|---|
| Bestshot | The frame of the video stream in which the face is captured in the best angle for further use in a face recognition system |
| Bbox | A rectangle that bounds the image space with a detected face |
| Intersection Over Union (IOU) | Parameter determines the coefficient of intersection of two detections |
| JSON | A text-based data exchange format based on JavaScript |
| Liveness | Software method that enables you to confirm whether a person in one or more images is "real" or a fraudster using a fake ID (printed face photo, video, paper or 3D mask) |
| MessagePack (MsgPack) | A fast and compact binary serialization format for data exchange, a more efficient alternative to JSON |
| Attributes | Gender and age of a person determined automatically by the system |
| Detection | Actions to find areas of the image containing faces |
| Spoofing attack | Substitution of a real person for a fake image (for example, a photograph) to deceive the system |

2. Introduction

This document contains information about VisionLabs LUNA KIOSK and describes how the components work.

VisionLabs LUNA KIOSK (hereinafter referred to as System) is a set of libraries that provide the possibility of realizing real-time operation to perform face detection in a frame, check the vitality of a person and transfer data to an external system.

The System is designed for:

- receiving and processing a color video stream from a video recording device,
- checking the image quality,
- selecting the bestshot,
- face detection by machine calculation method on two images,
- checking the presented image by Liveness-algorithms,
- protection against image spoofing by depth map analysis,
- subsequent transfer of the bestshot to device integration systems.

3. System requirements

To successfully install and run the system, your hardware must meet the minimum requirements specified below.

3.1 Minimum system requirements for x64 architecture

| Required Resource | Recommended |
|---------------------|--|
| Processor | Intel(R) Core(TM) i3-10110U |
| RAM | 4GB or more |
| Hard disk drive | HDD or SSD at least 1,4 B |
| Operating system | <ul style="list-style-type: none">• Windows 10 (64 bit)• Ubuntu 24.04 x64• Debian 10 x64 |
| Instruction Support | Advanced Vector Extensions 2 (AVX2) |

To run the application on Windows, install the [Visual C++ Redistributable package](#).

3.2 Minimum system requirements for ARM architecture

| Required Resource | Recommended |
|-------------------|---------------------------|
| Processor | Rockchip RK3588S |
| RAM | 4GB or more |
| Hard disk drive | HDD or SSD at least 128GB |
| Operating system | Armbian 23 (aarch64) |

3.3 Supported cameras

The system is optimized and tested for reliable operation with the following 3D and IR cameras:

- **Intel® RealSense™ Depth Cameras:**

- D415
- D435
- D435j
- D455

- **VLS LUNA CAMERA:**

- VLS LUNA CAMERA 3D / VLS LUNA CAMERA 3D Embedded
- VLS LUNA CAMERA 2D

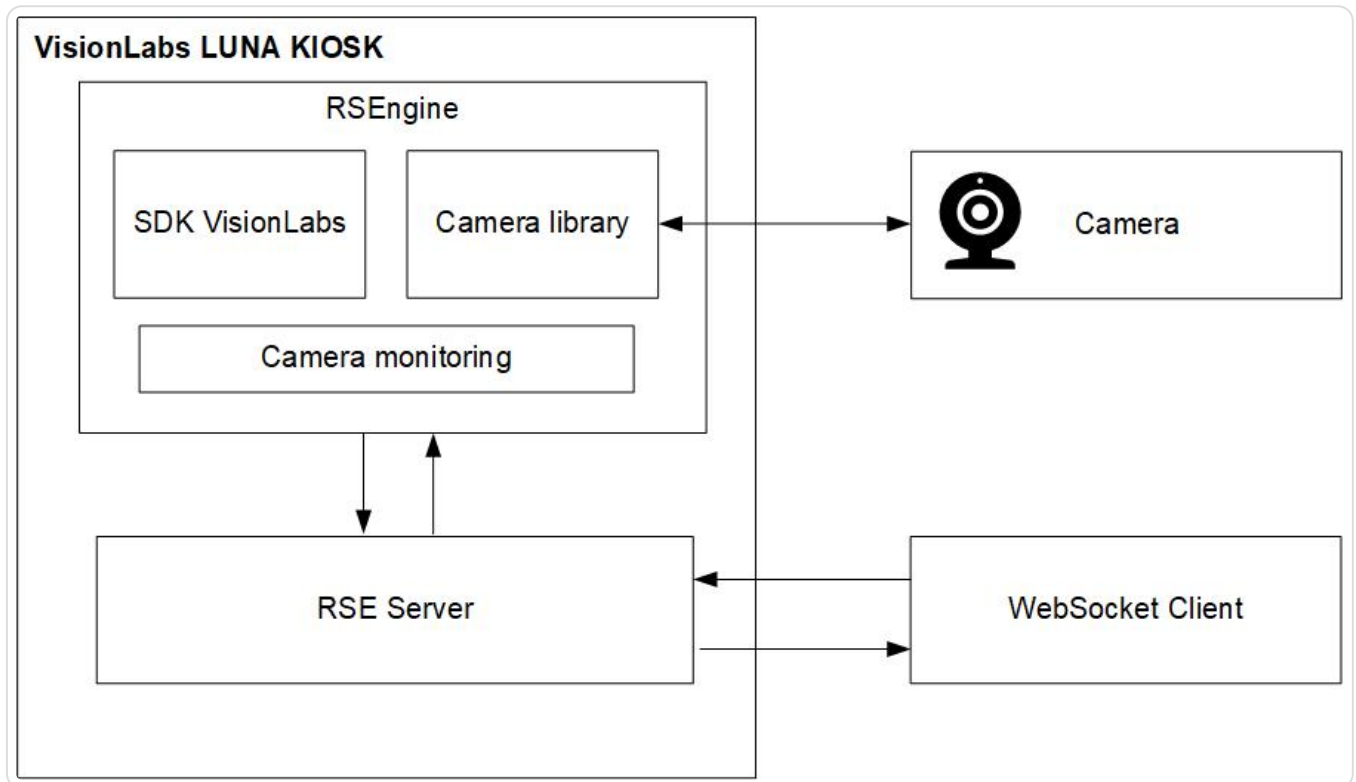
For information about VLS LUNA CAMERA 3D / VLS LUNA CAMERA 3D Embedded, please contact your VisionLabs representative.

3.3.1 Connection requirements

For stable performance, connect all supported cameras using USB 3.0.

4. System architecture

A high-level diagram of the System architecture is shown below (Figure 1).



System Solution Architecture Diagram

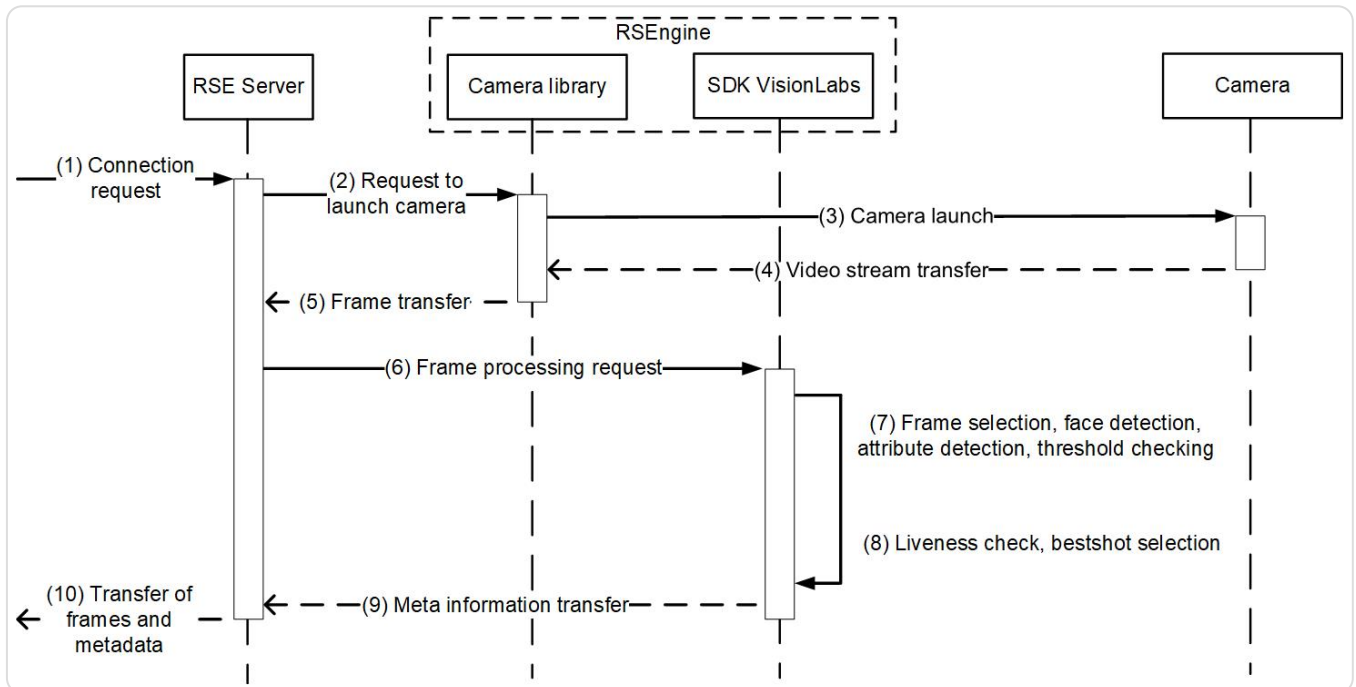
The system consists of the following components:

- RSEngine is a component that includes:
 - [VisionLabs SDK](#) library from the VisionLabs development kit for image processing;
 - libraries:
 - [RealSense2 SDK](#)—for work with Intel RealSense camera;
 - [VLS LUNA CAMERA 3D SDK](#)—for work with VLS LUNA CAMERA 3D camera;
 - [VLS LUNA CAMERA 2D SDK](#)—for work with VLS LUNA CAMERA 2D camera;
 - Camera monitoring, designed to check the status of the camera;
- RSE Server—WebSocket server for interaction with external system [WebSocket Client](#).

5. Algorithm

5.1 Bestshot selection

The interaction of components when selecting the bestshot is shown below (Figure 2).



Interaction diagram of System components when selecting the bestshot

An explanation for the figure is presented below (Table 3).

Table 3. The description of the interaction diagram of the System components when selecting the bestshot

| Step | Description |
|------|--|
| (1) | <p>The RSE Server receives a WebSocket connection request from a client.</p> <p>Example request:</p> <pre>GET ws://127.0.0.1:4444/ --establishing a WebSocket connection.</pre> <pre>0 --content of the message to start the session</pre> |

| Step | Description |
|------|---|
| (2) | <p>RSE Server sends a request to RSEngine to launch the camera (to the camera library).</p> <p>Depending on which camera is connected, <code>IntelRealSense</code>, <code>VLS LUNA CAMERA 3D</code> or <code>VLS LUNA CAMERA 2D</code>, RSE Server passes the request to the appropriate RSEngine library: <code>RealSense2 SDK</code>, <code>VLS LUNA CAMERA 3D SDK</code>, <code>VLS LUNA CAMERA 2D SDK</code>.</p> |
| (3) | Camera library launches the camera |
| (4) | The camera library receives RGB, IR, Depth video streams from the camera, splits them into frames and analyzes |
| (5) | <p>The camera library passes a set of frames to the RSE Server. Depending on the <code>cs_communication</code> parameter, different data is transferred:</p> <ul style="list-style-type: none"> - <code>cs_communication = msg-pack</code> : all three frames (RGB, IR, Depth) is transferred - <code>cs_communication = json</code> : only RGB frame is transferred |
| (6) | RSE Server sends a request to frame processing (performed on each frame) to the VisionLabs SDK |
| (7) | <p>SDK VisionLabs performs:</p> <ul style="list-style-type: none"> • selection of every third frame from the incoming video stream (the number of frames is not adjustable in the settings); • face detection on each frame; • determination of attributes and parameters of detected faces; • checking threshold values of the received parameters (head position, image quality, etc.). <p>If all checks are passed, the process continues (go to step 8).</p> <p>If at least one check fails, SDK VisionLabs sends a request to the camera for new frames to perform a second check as long as there is detection (return to step 4)</p> |

| Step | Description |
|------|--|
| (8) | <p>VisionLabs SDK performs a Liveness check and compares the resulting value of the Liveness score to the threshold one.</p> <p>If the received Liveness value is higher than the threshold, then the current frame becomes the bestshot.</p> <p>If the received Liveness value is below the threshold, SDK VisionLabs sends a request to the camera to to receive new frames to perform a second check as long as there is detection (return to step 4)</p> |
| (9) | If the Liveness check is successful, the received bestshot and facial attributes are sent to the RSE Server |
| (10) | RSE Server converts the selected bestshot and meta-information into MessagePack and sends it to the client in an external system |

5.2 Monitor camera status

The interaction of components when monitoring the state of the camera is shown below (Figure 3).

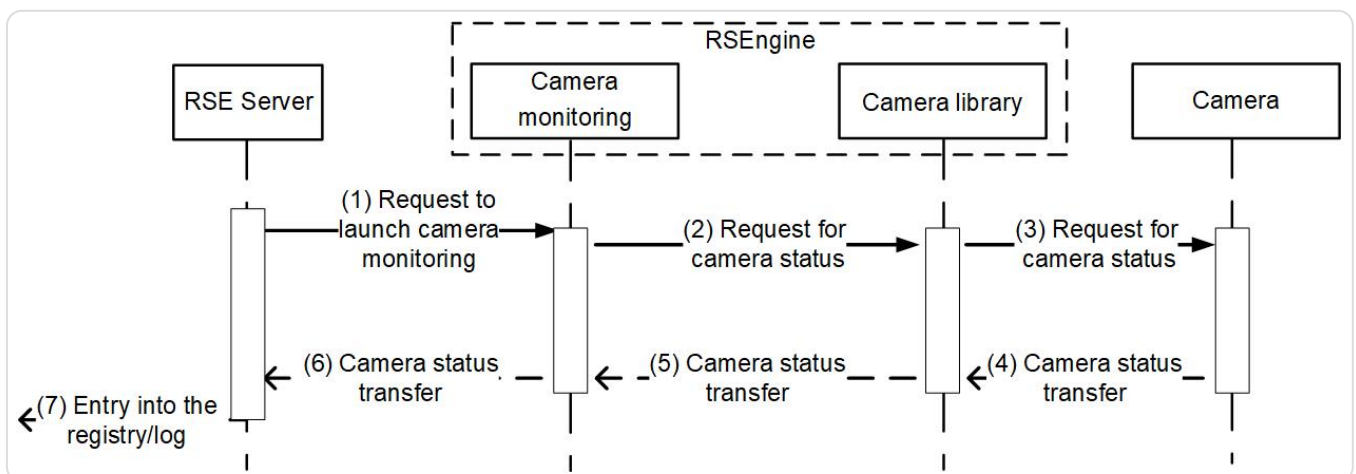


Diagram of component interaction when monitoring camera status

An explanation for the figure is presented below (Table 4).

Monitoring is started by default once every 300 seconds, you can change the duration in the `camera-monitoring` and `camera-monitoring-delay` parameters in the `rsengine.conf` file.

Table 4. The description of the interaction diagram of the System components when monitoring the camera state

| Step | Description |
|------|--|
| (1) | RSE Server sends a request to RSEngine to launch monitoring the camera |
| (2) | Camera Monitoring sends a request to the camera library for camera status— RealSense2 SDK, VLS LUNA CAMERA 3D SDK or VLS LUNA CAMERA 2D SDK depending on which camera is connected IntelRealSense, VLS LUNA CAMERA 3D or VLS LUNA CAMERA 2D SDK |
| (3) | The camera library transfers a request for camera status |
| (4) | The camera library receives the camera status |
| (5) | The camera library passes the camera status to the camera monitoring |
| (6) | RSEngine transfers camera status information to RSE Server |
| (7) | RSE Server entries camera status data to the registry (on Windows) or to the ./logs working folder (on Ubuntu 24.04 x64, Debian 10 x64 and Armbian 23) |

6. Component description

6.1 RSEngine component

RSEngine provides the interaction of the following libraries within the system:

- VisionLabs LUNA SDK
- RealSense2 SDK
- VLS LUNA CAMERA 3D SDK
- VLS LUNA CAMERA 2D SDK

This integration ensures efficient and reliable communication between these libraries, enabling advanced functionalities for image and video processing.

6.1.1 VisionLabs LUNA SDK component

VisionLabs LUNA SDK is a comprehensive software development kit that includes specialized libraries and neural networks designed for advanced image analysis. Its key capabilities include:

- **Face detection:** Identifying faces in images and locating key facial landmarks.
- **Best shot selection:** Automatically selecting the highest-quality frame from a video stream for further processing.
- **Image attribute estimation:** Analyzing image attributes for further Liveness estimations.
- **Liveness estimation:** Evaluating faces in images using Liveness algorithms to prevent spoofing attacks.

Note: All estimations described below are performed to ensure that the image meets Liveness requirements. These estimations are internal. Results are only displayed in cases of errors, such as when an image or face attribute is unsuitable for Liveness estimation. For more details on error codes and their descriptions, see [Appendix 2: Status Codes and Error Descriptions](#).

All estimations described below are performed to ensure that the image meets Liveness requirements. These estimations are internal, and the results are not transmitted externally. The result of the check can only be shown in case of an error, if any attribute of the image/face is not suitable for the Liveness estimation (see the description of errors in “Appendix 2. Status codes and error descriptions”);

6.1.2 RealSense2 SDK component

RealSense2 SDK is a component that provides the following functionalities:

- **Image acquisition:** Receives incoming images from Intel RealSense cameras.
- **Parameter configuration:** Allows you to configure detection parameters to suit specific requirements.
- **Camera control:** Enables turning the camera on or off and adjusting various settings, such as:
 - Laser backlight brightness
 - Auto exposure
 - Brightness levels
- **Automatic connection management:**
 - Automatically updates the connection with the camera.
 - If the connection is lost, the system attempts to reconnect to the camera.
 - In case reconnection fails, a soft reset of the connection cable is performed.
 - If all recovery operations are unsuccessful, the issue will be logged in the camera status report within the system logs.

6.1.3 Component: VLS LUNA CAMERA 3D SDK

VLS LUNA CAMERA 3D SDK is a component that provides the following functionalities:

- **Image acquisition:** Receives incoming images from VLS LUNA CAMERA 3D or VLS LUNA CAMERA 3D Embedded devices.
- **Parameter configuration:** Allows you to configure detection parameters to meet specific requirements.
- **Camera control:** Provides the ability to turn the camera on or off and adjust various settings, such as:
 - Laser illumination brightness
 - Auto exposure
 - Brightness levels

6.1.4 Component VLS LUNA CAMERA 2D SDK

VLS LUNA CAMERA 2D SDK is a component that provides the following functionalities:

- **Image acquisition:** Receives incoming images from VLS LUNA CAMERA 2D infrared cameras.
- **Parameter configuration:** Allows you to configure detection parameters to meet specific requirements.
- **Camera control:** Provides the ability to turn the camera on or off as needed.
- **Frame rotation adjustment:** Enables changing the rotation angle of the camera's video frame.

6.1.5 Camera functions

Face detection

The detector employs advanced face detection algorithms to address the following tasks:

- **Face detection:** Identifying faces within an image.
- **Key point localization:** Locating five key facial landmarks: two for the eyes, one for the tip of the nose, and two for the corners of the mouth.
- **Detection quality estimation:** Evaluating the probability that the detected object is indeed a face, ensuring high accuracy and reliability.

Image quality estimation

The quality of an image is evaluated based on the following parameters:

- Blurriness
- Lightness or overexposure
- Darkness or Underexposure

Mouth estimation

The mouth estimation evaluates the following parameters:

- **Open:** Indicates whether the mouth is open.
- **Occluded:** Detects if the mouth is blocked or covered by an external object.
- **Smiling:** Identifies the presence of a smile.

Eye state estimation

The eye state estimation evaluates the following parameters:

- **Closed:** Eyes are shut.
- **Open:** Eyes are open.
- **Occluded:** Eyes are covered, for example, by sunglasses or other objects.

Head pose estimation

The head pose estimation determines a person's head rotation angles in 3D space, specifically along the pitch, yaw, and roll axes:

- **Pitch:** This angle measures the vertical tilt of the head. It limits the head rotation along the X-axis.
- **Yaw:** This angle measures the horizontal rotation of the head. It limits the head rotation along the Y-axis.
- **Roll:** This angle measures the lateral tilt of the head. It limits the head rotation along the Z-axis.

Depth Liveness estimation

The "vitality" of a person in the image is verified using a depth map.

The process involves analyzing a 16-bit depth matrix. It contains detailed information about the distances of scene objects (such as faces) relative to the camera's viewpoint. This analysis helps determine whether the subject is a live person or a spoof, such as a photograph or mask.

IR Liveness estimation

The "vitality" of a person in the image is verified through infrared (IR) image analysis. This process ensures that the subject is a live human and not a spoof, such as a printed photo or mask.

Note: The camera must be equipped with infrared illumination to perform this check effectively.

LivenessOneShotRGB estimation

The LivenessOneShotRGB estimation determines whether a person's face is real or fake by detecting and mitigating various types of spoofing attacks. These include:

- **Printed photo attack:** One or more printed photos of another person are used.
- **Video replay attack:** A pre-recorded video of another person is displayed to trick the camera.
- **Printed mask attack:** An imposter cuts out a face from a photo and uses it to cover their own face.
- **3D mask attack:** An imposter wears a 3D mask designed to resemble the face of another person.

ESTIMATION CONFIGURATION

Regardless of the platform, you can configure the LivenessOneShotRGB settings via the `faceengine.conf` file in the `LivenessOneShotRGBEstimator::Settings` section:

| Parameter | Description |
|--------------------------|---|
| <code>version</code> | Specifies the algorithm version (<code>10</code> or <code>11</code>). |
| <code>deny2XLmode</code> | Enables or disables the <code>2XL</code> or <code>XL</code> mode. |

On Linux, in the `rsengine.conf` file, you can configure the following parameters:

| Parameter | Description |
|---|---|
| <code>liveness-depth-osl</code> | Enables or disables LivenessOneShotRGB estimation. The default value is <code>1</code> (enabled). |
| <code>liveness-depth-osl-threshold</code> | Specifies the LivenessOneShotRGB threshold value. The default value is <code>0.7</code> . |

On Windows, in the registry under `HKLM\Software\VisionLabs\RSEServer`, you can configure the following parameters:

| Parameter | Description |
|--|---|
| <code>LivenessDepthOSL</code> | Enables or disables LivenessOneShotRGB estimation. The default value is <code>1</code> (enabled). |
| <code>LivenessDepthOSLThreshold</code> | Specifies the LivenessOneShotRGB threshold value. The default value is <code>0.7</code> . |

DepthLivenessEstimator estimation

DepthLivenessEstimator performs liveness estimation for a face depth warp image. It exposes different threshold parameters that let you configure the estimator for your specific use case.

ESTIMATION CONFIGURATION

Regardless of the platform, you can configure the DepthLivenessEstimator settings via the `faceengine.conf` file in the `DepthLivenessEstimator::Settings` section:

| Parameter | Description | Default value |
|----------------------------------|---|---------------|
| <code>maxDepthThreshold</code> | Maximum depth distance (in mm) in the [0..inf] range. | 3000 |
| <code>minDepthThreshold</code> | Minimum depth distance (in mm) in the [0..maxDepthThreshold] range. | 100 |
| <code>zeroDepthThreshold</code> | Percentage of zero pixels in the [0..1] range. | 0.66 |
| <code>confidenceThreshold</code> | Score threshold in the [0..1] range above which person is considered to be alive. | 0.5 |

Pixel filtering logic

- Pixels with depth values greater than `maxDepthThreshold` are set to zero and excluded from estimation.
- Pixels with depth values less than `minDepthThreshold` are set to zero and excluded from estimation.
- If the percentage of zero-valued pixels in the input image exceeds `zeroDepthThreshold`, the entire image is rejected. In this case, all pixels are set to zero, regardless of any valid data present. The estimator returns a score close to 0.









6.1.6 Camera Monitoring component

Camera monitoring is used to check the status of the camera.

Camera monitoring queries the following camera parameters:

- Firmware data
- Operating status of infrared cameras—on/off
- RGB camera operating status—on/off
- Camera serial number
- Operation status of the entire camera—on/off
- Camera temperature
- Date of the last update

An example of the contents of the registry in the monitoring section is shown below (Figure 4).

| Name | Type | Data |
|--|-----------|----------------------|
|  (Default) | REG_SZ | (value not set) |
|  Camera | REG_SZ | Intel RealSense D415 |
|  Firmware | REG_SZ | 05.13.00.50 |
|  IR1 | REG_SZ | ON |
|  IR2 | REG_SZ | ON |
|  LastSuccessfulUpdate | REG_SZ | 2022-03-03 16:54:29 |
|  LastUpdate | REG_SZ | 2022-03-03 16:54:29 |
|  RGB | REG_SZ | ON |
|  Serial | REG_SZ | 907112061033 |
|  Status | REG_SZ | ON |
|  Temperature | REG_DWORD | 0x0000001f (31) |

Example of registry contents in the monitoring section

6.2 RSE Server component

The RSE Server is a WebSocket server that processes commands from external systems.

The RSE Server accepts requests and sends responses via WebSocket.

Request Format:

- Operation request code (1 byte)
- Additional payload (MessagePack or string)

Example of a request:

`GET ws://127.0.0.0.1:4444/` —establishing a WebSocket connection.

`0` —content of the message to start the session.

Response Format:

- Operation response code (1 byte)
- Additional payload (MessagePack or string)

Only one request can be processed at a time.

Depending on the type of integration required (selected at the discretion of the external system developer), you can configure RSE Server in the following ways:

- RSE Server expects requests (presented in Table 5) to connect to the camera from the external system—set the `cs_communication = msg-pack` parameter;
- RSE Server starts the process of receiving video stream and face detection process as soon as WebSocket connection is established —set the `cs_communication = json` parameter.

Table 5. Description of requests to RSE Server

| Request Name | Request Code | Description | Payload | Possible responses to the request |
|-------------------|--------------|---|---------|--|
| RSE_START_CAPTURE | 0 | Starts the process of receiving a video stream and the process of detecting faces | No | <code>RSE_CAPTURE_OK</code> (54) , <code>RSE_CAPTURE_META</code> (55) |
| RSE_STOP | 1 | Stops all running processes | No | <code>RSE_STOP_OK</code> (50) |

Depending on the type of integration selected (chosen at the discretion of the external system developer), the server response can be presented in two formats:

- if the external system developer has set the `cs_communication = msg-pack` parameter, each response will arrive in `msg-pack` format and will contain the `messageType` field with the response code and some additional data fields (payloads) described in Table 6;
- if the external system developer has set the `cs_communication = json` parameter, each response will arrive in `json` format and will be categorized into the message types described in Table 7.

Table 6. Responses to RSE Server requests with MessagePack response format

| Answer title | Code | Description ** | Payload |
|------------------|------|------------------------------|--|
| RSE_CAPTURE_OK | 54 | Captured set of video frames | <ul style="list-style-type: none">— rgbFrame—RGB frame in uint8 array format;— rgbFrameWidth—RGB frame width in pixels in int format;— rgbFrameHeight—RGB frame height in pixels in int format;— irFrame—IR frame in uint8 array format;— depthFrame—Depth frame in uint8 array format |
| RSE_CAPTURE_META | 55 | Metadata of detected persons | <ul style="list-style-type: none">— gotBestshot—indicator of whether bestshot was received, in bool format, returns:<ul style="list-style-type: none">• true—if bestshot was received;• false—if bestshot was not received• failureReason—the status or error code for liveness checks in int format (see error description in "Appendix 2. Status codes and error descriptions";)– bestshot—RGB frame, in uint8 array format:<ul style="list-style-type: none">• if gotBestshot=True, the response is an RGB frame, that passes all checks;• if gotBestshot=False, the field is blank |

| Answer title | Code | Description ** | Payload |
|--------------------|------|---|------------|
| RSE_STOP_OK | 50 | All processing has stopped. RSE Server is ready for new requests | No payload |
| RSE_UNKNOWN | 51 | The request was not recognized | No payload |
| RSE_INTERNAL_ERROR | 52 | An error occurred while processing the request | No payload |
| RSE_BUSY | 53 | Request denied because server is busy | No payload |

Table 7. Responses to RSE Server requests with JSON response format

| Message type | Description | Payload |
|--------------|---|---|
| visual | The type of response that is used for broadcasting the video stream to the user | <ul style="list-style-type: none">– msg_type—the type of message returned (visual);– img_b64—base64 camera frame;– metadata—parameters of the returned image:<ul style="list-style-type: none">• frame_size—image dimensions:• h—height of the image in pixels;• w—width of the image in pixels;• detections—coordinates of the detected person:<ul style="list-style-type: none">• h—height of the frame of the detected face;• w—width of the detected face frame• x—coordinates of the upper left corner of the detected face frame• y—coordinates of the upper left corner of the detected face frame• progress—displays the stages of Liveness of the detected person's background checks (percentage);• track_id—track identifier |

| Message type | Description | Payload |
|--------------|--|---|
| bestshot | <p>The type of response when a person is successfully found.</p> <p>This frame can be used for subsequent processing (e.g. in an external face recognition system)</p> | <ul style="list-style-type: none"> – msg_type—the type of message returned (bestshot); – img_b64—face from camera frame in base64 format; – metadata—parameters of the returned image: <ul style="list-style-type: none"> • frame_size—image dimensions: • h—height of the image in pixels; • w—width of the image in pixels; • detections—coordinates of the detected person: • h—height of the frame of the detected face; • w—width of the detected face frame • x—coordinates of the upper left corner of the detected face frame • y—coordinates of the upper left corner of the detected face frame • progress—displays the stages of Liveness of the detected person's background checks (percentage); • track_id—track identifier |

6.3 WebSocket Client component

WebSocket Client is an external component for interacting with RSE Server.

WebSocket Client is a JavaScript library for communicating with RSE Server via WebSocket. It uses the minimized binary serialization format [MessagePack](#) as a protocol library to encode and decode messages if the server returns responses in MessagePack format.

7. System setup

This section provides general information regarding System setup and logging.

The System allows you to customize the following parameters:

- General parameters (for more details see. ["Appendix 1: General configuration parameters"](#)).
- Image capture parameters (change in the `rsengine.conf` file);
- Face detection parameters (change in the `rsengine.conf` file);
- Parameters of IOU check execution - check of BBox face intersection on IR and RGB images (change in `rsengine.conf` file).

Image capture parameters, face detection parameters, and IOU verification parameters are fixed by default and cannot be changed by the user (administrator). Setting of these parameters is performed only by the copyright holder (VisionLabs LLC).

There are other .conf files included in the System distribution. It is not recommended to change parameters in these files, as it may disrupt the System operation. The System can be configured only within the instructions below.

7.1 System setup on Windows

The System is configured through the Windows Registry.

Settings received by the server from the client are saved until the System is restarted.

When configuring the System via the Windows Registry, write the settings in the following path:

```
\\*\* HKEY\LOCAL_MACHINE \ SOFTWARE \ VisionLabs \ RSEServer \*\*
```

The configuration parameters are described in Table 8 and Table 9.

When changing configuration settings, the new configurations overwrites the previous configurations.

To change parameters in the registry, find the corresponding parameter, make changes and apply them.

7.2 System settings on Ubuntu 24.04 x64 and Debian 10 x64

The System is configured by changing data in the client configuration files in the distribution (`server.conf` and `rsengine.conf`).

To apply the client configuration settings, make changes to the `server.conf` and `rsengine.conf` files (Table 8, Table 9) and restart RSE Server.

To change parameters, make changes to the appropriate file and apply the changes.

7.3 Logging

RSE Server includes a robust logging mechanism that captures operational and diagnostic data for monitoring, debugging, and maintenance. Logs are stored in both the console and log files.

The following log files are available:

| File | Description |
|-------------------|---|
| <i>server.log</i> | Stores RSE Server logs under Windows. |
| <i>stdout.log</i> | Contains all non-error messages, including informational, debugging, and other non-critical logs. |
| <i>stderr.log</i> | Exclusively contains error messages. |

The combination of *stdout.log* and *stderr.log* reflects the content of the *server.log* file, but separates standard messages from errors for easier analysis and troubleshooting.

The *stdout.log* and *stderr.log* files become available if the `log-level` parameter is set to `3`.

Note: Prior to v. 2.2.0, the `verboseLogging` parameter in the *faceengine.conf* file controlled logging verbosity. Starting from v. 2.2.0, the `verboseLogging` parameter is deprecated. Instead, you can configure logging levels using the `log-level` parameter. See [Appendix 1: General configuration parameters](#).

8. Switching camera types

The camera type is configured using the `ProcessorStrategy` parameter, which is written to the Windows Registry as:

```
ProcessorStrategy (REG_SZ) = IntelRealSense | VLSLunaCamera3D | VLSLunaCamera2D
```

8.1 Quick start: Launch commands

To configure and launch the service with a specific camera type, run one of the following commands from the script directory:

```
:: Intel RealSense (by default)
installService.bat ProcessorStrategy=IntelRealSense

:: VL SLuna 3D
installService.bat ProcessorStrategy=VLSLunaCamera3D

:: VL SLuna 2D
installService.bat ProcessorStrategy=VLSLunaCamera2D
```

You can also run the script without any parameters — it will use the default strategy:

```
installService.bat
```

In this case, the default parameter will be used: `ProcessorStrategy=IntelRealSense`.

8.1.1 Changing the default camera

To change the default camera type:

1. Open `installService.bat` in a text editor. 2. Locate the line:

```
set ProcessorStrategy=IntelRealSense
```

3. Replace `IntelRealSense` with your preferred default:

- `IntelRealSense` - `VLSLunaCamera3D` - `VLSLunaCamera2D`

Tip: You can also enable extended message display mode when running the script:

```
installService.bat ProcessorStrategy=VLSLunaCamera2D quietReg=false
```

By default, `quietReg` is set to `true`. If you need to change this behavior, edit line number 10 (`set quietReg=true`).

8.2 Configuration differences

The installation script writes a core set of registry keys for all strategies, plus strategy-specific settings.

8.2.1 Common settings

These keys are written regardless of camera type if ROI (Region of Interest) is enabled:

```
RoiEnable (REG_DWORD) 1
RoiX (REG_DWORD) 160
RoiY (REG_DWORD) 0
RoiWidth (REG_DWORD) 320
RoiHeight (REG_DWORD) 480
```

8.2.2 IntelRealSense-specific settings

Additional keys for Intel RealSense cameras:

```
IRStreamDarknessCheck (REG_DWORD) = 0
LivenessDepthOSL (REG_DWORD) = 1
LivenessDepthOSLThreshold (REG_DWORD) = 70
RgbIrMatch (REG_DWORD) = 0
RgbIRMatchThreshold (REG_DWORD) = 120
RGBCoordinatesTransfer (REG_DWORD) = 1
IOULivenessThreshold (REG_DWORD) = 50
```

8.2.3 VLSSLunaCamera3D-specific settings

Only the general ROI settings are applied; no additional keys are specified.

8.2.4 VLSSLunaCamera2D-specific settings

One additional key:

```
FrameRotation (REG_DWORD) 0
```

9. Appendix 1: General configuration parameters

General parameters are changed in the registry (for Windows) files `server.conf` (Table 8) and `rsengine.conf` (Table 9). Windows registry configuration parameters are shown below (Table 10).

Table 8. Common configuration parameters in the `server.conf` file

| Parameter name | Data type | Silent value | Description |
|---------------------------------|-----------|-------------------------------------|---|
| <code>data-path</code> | string | <code>./data.</code> | <p>Path RSE Server data directory.</p> <p>It is not recommended to change Parameter value.</p> |
| <code>rsengine-conf-path</code> | string | <code>./client/rsengine.conf</code> | <p>Path to RSEngine library config file.</p> <p>Relevant for setting up the System on Ubuntu 24.04 x64, Debian 10 x64, Armbian 23 and Windows OS using configuration files</p> |
| <code>cs-communication</code> | string | <code>msg-pack</code> | <p>The type of interaction between the server and the client. Depends on the selected System configuration.</p> <p>Can take the following values:</p> <ul style="list-style-type: none">• <code>json</code> : the data type returned by the server is JSON;• <code>msg-pack</code> : the data type returned by the server is MessagePack |
| <code>bestshot-format</code> | string | <code>jpg</code> | <p>The returned image format of the bestshots.</p> <p>Selected based on the requirements of external software. Can accept the following values:</p> <ul style="list-style-type: none">• <code>jpg</code> : JPEG format;• <code>png</code> : PNG format |

| Parameter name | Data type | Silent value | Description |
|-----------------------------------|--------------------------|--------------------------|--|
| <code>save-bestshot-ondisk</code> | <code>int</code> | <code>0</code> | <p>Saving bestshots to disk to the directory <code>save-bestshot-path/SaveBestshotPath</code>.</p> <ul style="list-style-type: none"> • <code>0</code> –do not save; • <code>1</code> –save. |
| <code>save-bestshot-path</code> | <code>string</code> | <code>./bestshots</code> | <p>The directory for saving the best shots when the <code>save-bestshot-ondisk/SaveBestshotonDisk</code> variable is activated.</p> |
| <code>encrypt-bestshot</code> | <code>int</code> | <code>0</code> | <p>Encryption of the bestshots when saving them. Not used in this version of the System.</p> <ul style="list-style-type: none"> • <code>0</code> –do not encrypt; • <code>1</code> –encrypt. |
| <code>server-host</code> | <code>string</code> | <code>127.0.0.1</code> | <p>IP address of the server on which to run to accept websocket connection.</p> <p>When using one copy, specify localhost; when using several running copies, specify the main server.</p> |
| <code>server-port</code> | <code>int</code> | <code>4444</code> | <p>Port on which RSE Server accepts connections</p> |
| <code>log-level</code> | <code>int, [0, 3]</code> | <code>1</code> | <p>The logging level filters log messages and has the following levels from 0 to 3:</p> <ul style="list-style-type: none"> • <code>0</code> –disable logging; • <code>1</code> –logging information about work; • <code>2</code> –logging warnings and information about work; • <code>3</code> –logging errors, warnings and operation information. |

| Parameter name | Data type | Silent value | Description |
|-----------------------------------|---------------------|---------------------|--|
| <code>log-path</code> | <code>string</code> | <code>./logs</code> | Path to a writable directory for storing server logs |
| <code>log-file-rotation</code> | <code>int</code> | <code>0</code> | Daily log rotation <ul style="list-style-type: none"> • <code>0</code> –disabled; • <code>1</code> –enabled. |
| <code>continuous-bestshots</code> | <code>int</code> | <code>0</code> | Continue to receive bestshots in the session, even if bestshot has already been received <ul style="list-style-type: none"> • <code>0</code> –disabled; • <code>1</code> –enabled. |

Table 9. Common configuration parameters in the `rsengine.conf` file

In the `rsengine.conf` file, the unique settings for each camera are separated into blocks.

| Parameter name | Data type | Silent value | Description |
|---------------------------------|---------------------|-----------------------------|---|
| <code>processor-strategy</code> | <code>string</code> | <code>IntelRealSense</code> | Camera System Mode: <ul style="list-style-type: none"> • <code>IntelRealSense</code> –working with Intel RealSense camera; • <code>VLSLunaCamera3D</code> –working with VLS LUNA CAMERA 3D; • <code>VLSLunaCamera2D</code> –working with VLS LUNA CAMERA 2D. |
| <code>camera-monitoring</code> | <code>int</code> | <code>1</code> | Parameter enables/disables monitoring of the camera status. <ul style="list-style-type: none"> • <code>0</code> –turns off monitoring; • <code>1</code> –turns on monitoring |

| Parameter name | Data type | Silent value | Description |
|-------------------------|-----------|--------------|---|
| camera-monitoring-delay | int | 300 | Parameter sets the request frequency for the camera status from the monitoring service |
| rgb-ir-match | int | 0 | <p>Using RGB and IR camera data to perform checks</p> <p>It is not recommended to change this parameter.</p> <p>0 – off;</p> <p>1 – on.</p> |
| check-eyes | int | 1 | <p>Using information about eye position and status for checking Liveness.</p> <p>0 –off;</p> <p>1 –on.</p> |
| check-mouth | int | 0 | <p>Using information about mouth status for checking Liveness.</p> <p>0 –off;</p> <p>1 –on.</p> |
| liveness-depth | int | 1 | <p>Using information about face volumetric on image when checking Liveness.</p> <p>It is not recommended to change this parameter.</p> <p>0 –off;</p> <p>1 –on.</p> |
| liveness-depth-osl | int | 1 | |

| Parameter name | Data type | Silent value | Description |
|---|--------------------|------------------|---|
| <code>liveness-depth-threshold</code> | <code>float</code> | <code>0.0</code> | <p>The minimum threshold for the Liveness value when checking face in the volumetric image.</p> <p>Threshold is set between 0.0 and 1.0 (0...100 for Windows), where:</p> <ul style="list-style-type: none"> • <code>0</code> –no check • <code>0.1</code> –high probability of missing a fake • <code>1</code> –low probability of missing a fake, high probability to miss a real person |
| <code>liveness-depth-osl-threshold</code> | <code>float</code> | <code>0.7</code> | <p>Minimum threshold for Liveness value while face checking</p> <p>Parameter is chosen analytically by the developers and is not recommended to be changed. The threshold is set in the range from 0.0 to 1.0 (0...100 for Windows), where:</p> <ul style="list-style-type: none"> • <code>0</code> –no check • <code>0.1</code> –a high probability of missing a fake • <code>1</code> –low probability of missing a fake, high probability to miss a real person |
| <code>rgb-coordinates-transfer</code> | <code>int</code> | <code>1</code> | <p>Transferring the Bbox coordinates of a face from an RGB image to an IR image for further processing.</p> <p><code>0</code> –off;</p> <p><code>1</code> –on.</p> |

| Parameter name | Data type | Silent value | Description |
|------------------------|-----------|--------------|---|
| iou-liveness-threshold | float | 0.5 | <p>The threshold for using IOU when building a Bbox. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| quality-threshold | float | 0.8 | <p>Minimum threshold for evaluating image quality before checking Liveness. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set between 0.0 and 1.0 (0...100 for Windows).</p> |
| rgb-ir-match-threshold | float | 1.2 | <p>Face comparison threshold from IR and RGB images. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set between 0.0 and 1.0 (0...100 for Windows).</p> |
| margin | int | 10 | <p>Minimum space between the face box and the frame boundaries in pixels.</p> <p>The face must be at least 10 pixels from the frame border when performing a Liveness check, so that face information is not lost.</p> <p>10...100 pixels</p> |

| Parameter name | Data type | Silent value | Description |
|----------------------|-----------|--------------|--|
| suspicious-threshold | float | 0.6 | <p>Minimum image quality threshold at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| light-threshold | float | 0.9 | <p>Minimum threshold for checking the quality of face illumination at image at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| dark-threshold | float | 0.93 | <p>Minimum threshold for checking the quality of darkness on the face. image at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| blur-threshold | float | 0.94 | <p>Minimum threshold for checking the quality of blurred face image at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |

| Parameter name | Data type | Silent value | Description |
|---------------------------------------|------------------|-----------------|--|
| <code>yaw-threshold</code> | <code>int</code> | <code>15</code> | Maximum head tilt angle relative to the camera axis,, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed. |
| <code>pitch-threshold</code> | <code>int</code> | <code>15</code> | Maximum head rotation angle relative to the camera axis, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed. |
| <code>roll-threshold</code> | <code>int</code> | <code>10</code> | Maximum head rotation angle relative to the camera axis, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed. |
| <code>autoexp-rgb</code> | <code>int</code> | <code>1</code> | Enable auto exposure mode for RGB images. It is not recommended to disable this setting. <code>0</code> –off; <code>1</code> –on. |
| <code>autoexp-ir</code> | <code>int</code> | <code>1</code> | Enable auto exposure mode for IR image. It is not recommended to disable this setting. <code>0</code> –off; <code>1</code> –on. |
| <code>ir-stream-darkness-check</code> | <code>int</code> | <code>0</code> | Check for lack of image illumination in IR image. |

| Parameter name | Data type | Silent value | Description |
|-----------------------------|------------------|------------------|---|
| <code>roi-enable</code> | <code>int</code> | <code>1</code> | <p>Cropping the original frame to reduce the area of interest to improve the recognition quality. The preset parameters below limit the central part of the frame, where the face is least distorted.</p> <p>It is not recommended that this setting be turned off.</p> <p><code>0</code> –off; <code>1</code> –on.</p> |
| <code>roi-x</code> | <code>int</code> | <code>160</code> | <p>Horizontal coordinate of the upper left corner of the area of interest. Specified from the upper left corner of the frame.</p> <p>It is not recommended to change this parameter.</p> |
| <code>roi-y</code> | <code>int</code> | <code>0</code> | <p>Vertical coordinate of the top left corner of the area of interest. Set from the upper left corner of the frame.</p> <p>It is not recommended to change this parameter.</p> |
| <code>roi-width</code> | <code>int</code> | <code>320</code> | |
| <code>roi-height</code> | <code>int</code> | <code>480</code> | <p>Height of the area of interest. It is not recommended to change this parameter.</p> |
| <code>frame-rotation</code> | <code>int</code> | <code>0</code> | <p>The rotation angle of the camera frame.</p> <p>Possible values: <code>0</code> , <code>90</code> , <code>180</code> , <code>270</code> . For the location of the camera depending on the rotation angle of the video frame, see the diagram below (Figure 5)</p> |

| Parameter name | Data type | Silent value | Description |
|--------------------------|------------------|----------------|--|
| <code>liveness-ir</code> | <code>int</code> | <code>1</code> | <p>IR Liveness check.</p> <p><code>0</code> –off;</p> <p><code>1</code> –is on</p> |

Table 10. Windows Registry Configuration Settings

| Parameter name | Data type | Silent value | Description |
|------------------------------|---------------------|--|--|
| <code>DataPath</code> | <code>REG_SZ</code> | <code>{Path to system files}/data</code> | <p>Path to RSE Server data directory.</p> <p>It is not recommended to change the value of Parameter.</p> |
| <code>CSCommunication</code> | <code>REG_SZ</code> | <code>msg-pack</code> | <p>The type of interaction between the server and the client. Depends on the selected System configuration.</p> <p>Can take the following values:</p> <ul style="list-style-type: none"> • <code>json</code> : the data type returned by the server is JSON; • <code>msg-pack</code> : the data type returned by the server is MessagePack |
| <code>BestshotFormat</code> | <code>REG_SZ</code> | <code>jpg</code> | <p>The returned image format of the bestshots.</p> <p>Selected based on the requirements of external software. Can accept the following values:</p> <ul style="list-style-type: none"> • <code>jpg</code> : JPEG format; • <code>png</code> : PNG format |

| Parameter name | Data type | Silent value | Description |
|---------------------------------|------------------------|--------------------------|--|
| <code>SaveBestshotonDisk</code> | <code>REG_DWORD</code> | <code>0</code> | <p>Saving the bestshots to disk to the directory <code>save-bestshot-path/SaveBestshotPath</code>.</p> <ul style="list-style-type: none"> • <code>0</code> –do not save; • <code>1</code> –save. |
| <code>save-bestshot-path</code> | <code>string</code> | <code>./bestshots</code> | <p>The directory for saving the best shots when the <code>save-bestshot-ondisk/SaveBestshotonDisk</code> variable is activated.</p> |
| <code>EncryptBestshot</code> | <code>REG_DWORD</code> | <code>0</code> | <p>Encryption of the bestshots when saving them. Not used in this version of the System.</p> <ul style="list-style-type: none"> • <code>0</code> –do not encrypt; • <code>1</code> –encrypt. |
| <code>ServerHost</code> | <code>REG_SZ</code> | <code>127.0.0.1</code> | <p>IP address of the server on which to run to accept websocket connection.</p> <p>When using one copy, specify localhost; when using several running copies, specify the main server.</p> |
| <code>ServerPort</code> | <code>REG_DWORD</code> | <code>4444</code> | <p>The port on which the RSE Server accepts connections</p> |

| Parameter name | Data type | Silent value | Description |
|------------------------|-----------|----------------|--|
| LogLevel | REG_DWORD | 1 | <p>The logging level filters log messages and has the following levels from 0 to 3:</p> <ul style="list-style-type: none"> • 0 –disable logging; • 1 –logging information about work; • 2 –logging warnings and information about work; • 3 –logging errors, warnings and operation information. |
| LogPath | REG_SZ | C:\RSE\logs | Path to a writable directory for storing server logs |
| Processor | REG_SZ | IntelRealSense | <p>Camera System Mode:</p> <ul style="list-style-type: none"> • IntelRealSense –working with Intel RealSense camera; • VLSLunaCamera3D –working with VLS LUNA CAMERA 3D; • VLSLunaCamera2D –working with VLS LUNA CAMERA 2D. |
| Camera Monitoring | REG_DWORD | 1 | <p>Parameter enables/disables monitoring camera status.</p> <ul style="list-style-type: none"> • 0 –turns off monitoring; • 1 –turns on monitoring. |
| Camera MonitoringDelay | REG_DWORD | 300 | Parameter sets the request frequency for the camera status from the monitoring service. |

| Parameter name | Data type | Silent value | Description |
|------------------|-----------|--------------|---|
| RgbIrMatch | REG_DWORD | 0 | <p>Using RGB and IR camera data to perform checks</p> <p>It is not recommended to change this parameter.</p> <p>0 – off;</p> <p>1 – on.</p> |
| CheckEyes | REG_DWORD | 1 | <p>Using information about eye position and status for checking Liveness.</p> <p>0 –off;</p> <p>1 –on.</p> |
| CheckMouth | REG_DWORD | 0 | <p>Using information about mouth status for checking Liveness.</p> <p>0 –off;</p> <p>1 –on.</p> |
| `LivenessDepth | REG_DWORD | 1 | <p>Using information about face volumetric on image when checking Liveness.</p> <p>It is not recommended to change this parameter.</p> <p>0 –off;</p> <p>1 –on.</p> |
| LivenessDepthOSL | REG_DWORD | 1 | <p>LivenessOneShotRGB estimation. We do not recommend to change this parameter.</p> <p>0 –off;</p> <p>1 –on.</p> |

| Parameter name | Data type | Silent value | Description |
|--------------------------------|-----------|--------------|--|
| `LivenessDepth | REG_DWORD | 0 | <p>The minimum threshold for the Liveness value when checking face in the volumetric image.</p> <p>Threshold is set between 0.0 and 1.0 (0...100 for Windows), where:</p> <ul style="list-style-type: none"> • 0 –no check • 0.1 –high probability of missing a fake • 1 –low probability of missing a fake, high probability to miss a real person |
| `LivenessDepth OSLThreshold | REG_DWORD | 0.7 | <p>Minimum threshold for Liveness value while face checking</p> <p>Parameter is chosen analytically by the developers and is not recommended to be changed. The threshold is set in the range from 0.0 to 1.0 (0...100 for Windows), where:</p> <ul style="list-style-type: none"> • 0 –no check • 0.1 –a high probability of missing a fake • 1 –low probability of missing a fake, high probability to miss a real person |

| Parameter name | Data type | Silent value | Description |
|----------------------------|-----------|--------------|---|
| RGBCoordinates Transfer | REG_DWORD | 1 | <p>Transferring the Bbox coordinates of a face from an RGB image to an IR image for further processing.</p> <p>0 –off; 1 –on.</p> |
| IOULiveness | REG_DWORD | 50 | <p>The threshold for using IOU when building a Bbox. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| Quality Threshold | REG_DWORD | 80 | <p>Minimum threshold for evaluating image quality before checking Liveness. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set between 0.0 and 1.0 (0...100 for Windows).</p> |
| RgbIRMatch Threshold | REG_DWORD | 120 | <p>Face comparison threshold from IR and RGB images.</p> <p>Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set between 0.0 and 1.0 (0...100 for Windows).</p> |

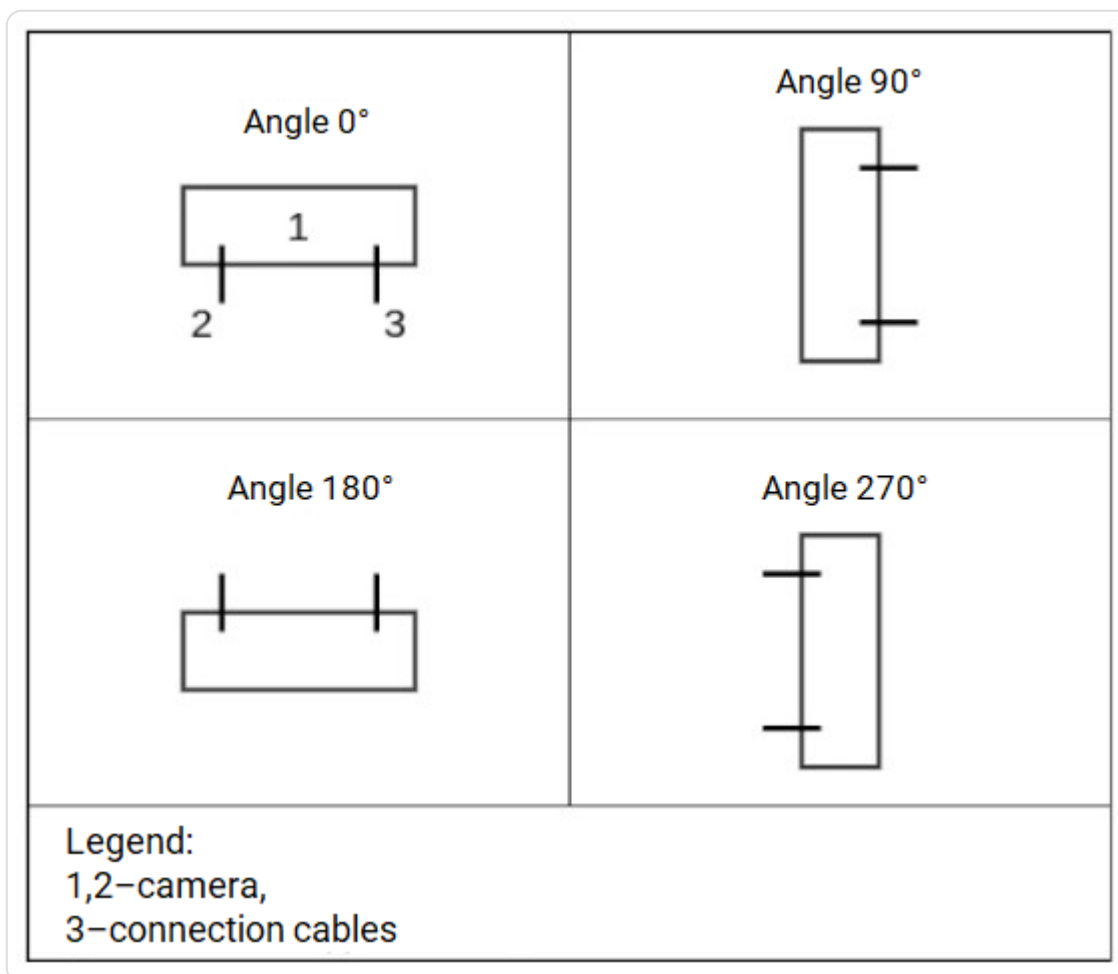
| Parameter name | Data type | Silent value | Description |
|----------------------------------|------------------------|-----------------|--|
| <code>`Margin</code> | <code>REG_DWORD</code> | <code>20</code> | <p>Minimum space between the face box and the frame boundaries in pixels.</p> <p>The face must be at least 10 pixels from the frame border when performing a Liveness check, so that face information is not lost.</p> <p>10...100 pixels</p> |
| <code>SuspiciousThreshold</code> | <code>REG_DWORD</code> | <code>60</code> | <p>Minimum image quality threshold at which the Liveness check will be performed</p> <p>Parameter is chosen analytically by the developers and is not recommended to be changed.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| <code>LightThreshold</code> | <code>REG_DWORD</code> | <code>90</code> | <p>Minimum threshold for checking the quality of face illumination at image at which the Liveness check will be performed.</p> <p>Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |

| Parameter name | Data type | Silent value | Description |
|----------------|-----------|--------------|--|
| DarkThreshold | REG_DWORD | 93 | <p>Minimum threshold for checking the quality of darkness on the face. image at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| BlurThreshold | REG_DWORD | 94 | <p>Minimum threshold for checking the quality of blurred face image at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to change.</p> <p>The threshold is set in the range of 0.0 to 1.0 (0...100 for Windows).</p> |
| YawThreshold | REG_DWORD | 15 | <p>Maximum head tilt angle relative to the camera axis,, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> |
| PitchThreshold | REG_DWORD | 15 | <p>Maximum head rotation angle relative to the camera axis, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed.</p> |

| Parameter name | Data type | Silent value | Description |
|---------------------------|-----------|--------------|---|
| RollThreshold | REG_DWORD | 10 | Maximum head rotation angle relative to the camera axis, at which the Liveness check will be performed. Parameter is chosen analytically by the developers and is not recommended to be changed. |
| AutoexpRGB | REG_DWORD | 1 | Enable auto exposure mode for RGB images. It is not recommended to disable this setting. 0 –off; 1 –on. |
| AutoexpIR | REG_DWORD | 1 | Enable auto exposure mode for IR image. It is not recommended to disable this setting. 0 –off; 1 –on. |
| IRStream DarknessCheck | REG_DWORD | 0 | Check for lack of image illumination in IR image. |
| RoiEnable | REG_DWORD | 1 | Cropping the original frame to reduce the area of interest to improve the recognition quality. The preset parameters below limit the central part of the frame, where the face is least distorted. It is not recommended that this setting be turned off. 0 –off; 1 –on. |

| Parameter name | Data type | Silent value | Description |
|----------------------|-----------|--------------|---|
| RoiX | REG_DWORD | 160 | Horizontal coordinate of the upper left corner of the area of interest. Specified from the upper left corner of the frame. It is not recommended to change this parameter. |
| RoiY | REG_DWORD | 0 | Vertical coordinate of the top left corner of the area of interest. Set from the upper left corner of the frame. It is not recommended to change this parameter. |
| RoiWidth | REG_DWORD | 320 | Width of the area of interest. It is not recommended to change this parameter. |
| RoiHeight | REG_DWORD | 480 | Height of the area of interest. It is not recommended to change this parameter. |
| LogFileRotation | REG_DWORD | 0 | Daily log rotation <ul style="list-style-type: none"> 0 –disabled; 1 –enabled. |
| Continuous Bestshots | REG_DWORD | 0 | Continue to receive bestshots in session even if bestshot has already been received 0 –off; 1 –on. |

| Parameter name | Data type | Silent value | Description |
|----------------|-----------|--------------|---|
| FrameRotation | int | 0 | <p>The rotation angle of the camera frame.</p> <p>Possible values: 0 , 90 , 180 , 270 . For the location of the camera depending on the rotation angle of the video frame, see the diagram below (Figure 5)</p> |
| LivenessIr | int | 1 | <p>IR Liveness check.</p> <p>0 –off;</p> <p>1 –is on</p> |



Location of the VLS LUNA CAMERA 2D camera depending on the rotation angle of the video frame

10. Appendix 2. Status codes and error descriptions

Status codes and descriptions of `failureReason` errors in the `RSE_CAPTURE_META` response payloads when performing a Liveness check are summarized in Table 11.

Codes are common to msg-pack and JSON responses.

Table 11. Status codes and description of `failureReason` errors in `RSE_CAPTURE_META` response

| Status Code | Description |
|-------------|--|
| 0 | There are no errors, the frame passes checks |
| 1 | Incorrect RGB frame |
| 2 | Incorrect Depth frame |
| 3 | Incorrect IR frame |
| 4 | Face wasn't detected |
| 5 | Face wasn't detected (face too small in frame) |
| 6 | The detected face doesn't pass one of the configuration parameters |
| 7 | It is impossible to estimate the face in the frame by 5 points |
| 10 | The face in the frame is cropped |
| 11 | Face rotated—eye distance is too short |
| 13 | Failed to normalize RGB frame |
| 14 | Failed to normalize Depth frame |
| 15 | Failed to normalize IR frame |
| 16 | Incorrect head position |
| 17 | Eyes are closed |
| 18 | Neutral face expression of the mouth muscles is necessary |
| 19 | Liveness frame Depth check was failed |
| 20 | Liveness IR frame check was failed |
| 21 | Poor frame quality |

| Status Code | Description |
|-------------|---|
| 22 | RGB frame is too bright |
| 23 | RGB frame is too dark |
| 25 | Image is blurry |
| 26 | LivenessOneShotRGB estimation failed |
| 28 | Liveness check of RGB and IR frames comparison is in progress |
| 29 | Liveness check IR frame without illumination |
| 31 | Failed to recognize face on the frame |
| 32 | Failed to detect face on IR frame |
| 33 | Low quality Depth frame |
| 34 | Several faces with overlapping BBox zones detected |