VzPeopleCountSDK

User Guide

Contents

Release Record

Version	Instruction	Date
V1.0.0	First edition release	20201107
V1.1.0	Add Vz_PCSetMaxDetectDistance, Change Vz_PCSetDoorOpenState	20210114

1 Overview

VzpeopleCountSDK is a customer flow algorithm software development kit based on Vzense TOF camera developed by Vzense, which is currently applicable to Linux/Arm Linux. Developers can get a count of how long customers stay in front of the camera, where they stay and how many of them stay through the SDK. It provides a series of friendly APIs and simple application examples for developers.

2 SDK Module Diagram

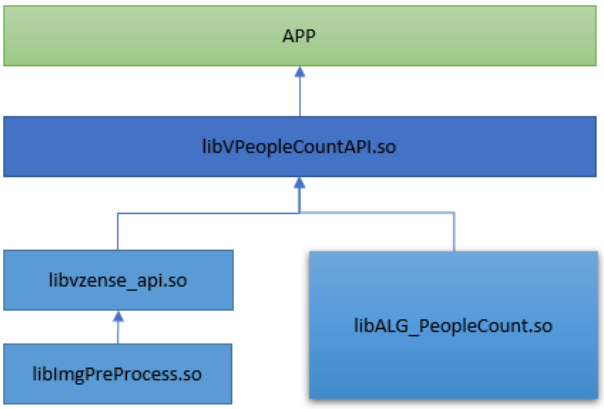

Fig1.SDK Module Diagram

3 API Call Sequence

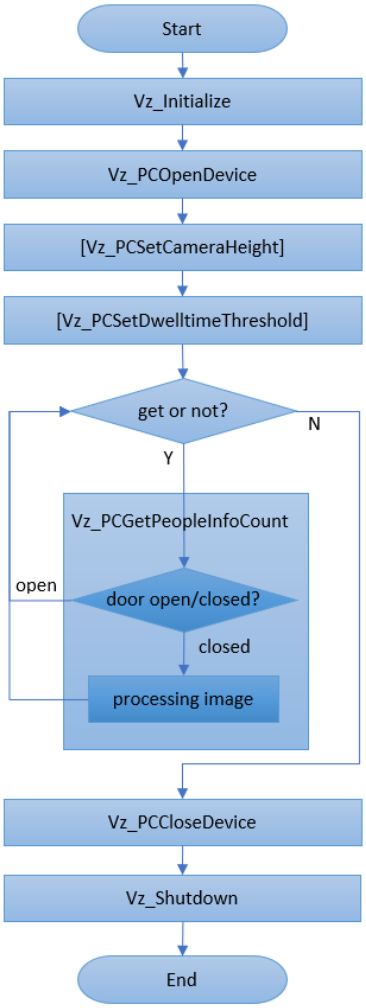

Fig2. API Call Sequence

PS:[xx] is an optional configuration item.

4 Algorithm Parameter Settings

4.1 Camera Mounting Height

The camera mounting height can be set by calling the *Vz_PCSetCameraHeight* function and are saved to the configuration file (path: ./ Alg_PCConfig.ini). The values set in *Vz_PCSetCameraHeight* take effect in real time before Vz_OpenDevice, otherwise it will take effect in the next time the application is started. Alternatively, you can set the camera mounting height by directly modifying the ALG:CameraHeight configuration option in the configuration file. The next time the program starts, the setting takes effect.

4.2 Dwell time threshold

The dwell time threshold can be set by calling the *Vz_PCSetDwelltimeThreshold* function. The values set in *Vz_PCSetDwelltimeThreshold* take effect in real time and are saved to the configuration file (path: ./ Alg_PCConfig.ini). Alternatively, you can directly modify the ALG:Dwelltimethreshold configuration item in the configuration file to set the dwell time threshold. The next time the program starts, the setting takes effect.

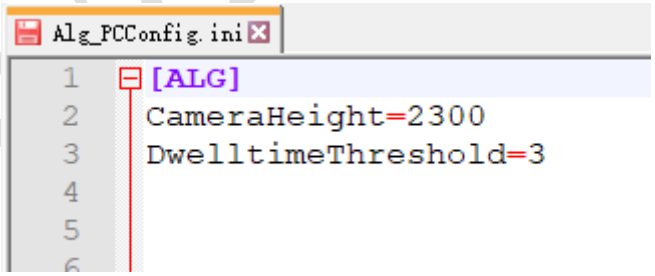

Fig3. Configuration File

5 API Introduction

5.1 Enum Type

5.1.1 VzReturnStatus

Description:

Return status of API

Enumerator:

VzRetOK: Succeed

VzRetNoDeviceConnected: No depth camera connected or the camera connected abnormally. Please check HW connection or try to plug out and plug camera in again.

VzRetInvalidDeviceIndex: The input device index is invalid

VzRetDevicePointerIsNull: The device structure pointer is null

VzRetInvalidFrameType: The input frame type is invalid

VzRetFramePointerIsNull: The output frame is empty

PsRetReadNextFrameError: Error when capturing the next image frame

VzRetInputPointerIsNull: An input pointer parameter is null.

PsRetCameraNotOpened: Camera is not opened

PsRetInvalidParams: Parameter is invalid

VzRetCurrentVersionNotSupport: This feature is not supported in the current version.

VzRetUpgradeImgError: There is an error in the upgrade file.

VzRetUpgradeImgPathTooLong: Upgrade file path length greater than 260.

VzRetUpgradeCallbackNotSet: Ps2_SetUpgradeStatusCallback is not invoked.

VzRetNoAdapterConnected : There is no adapter connected.

VzRetDoorWasOpend: The door has been opened.

PsRetOthers: Other error

5.1.2 VzDeviceStatus

Description:

Indicates the current status of the device

Enumerator:

DEVICE_NORMAL: The device is in normal working condition.

DEVICE_UPGRADE_BEGIN: The device is in the upgrade state and ready to upgrade.

DEVICE_UPGRADE_IMG_COPY: Finish copying firmware files to the device.

DEVICE_UPGRADE_IMG_CHECK_DOING: The device is checking the integrity of the firmware file.

DEVICE_UPGRADE_IMG_CHECK_DONE: The device completes the firmware file integrity check.

DEVICE_UPGRADE_DOING: The device is burning firmware file.

DEVICE_UPGRADE_RECHECK_DOING: The device is being rechecked.

DEVICE_UPGRADE_RECHECK_DONE: The device upgrade review completed.

DEVICE_UPGRADE _DONE: The device upgrade completed.

5.2 Struct Type

5.2.1 VzPCFrame

Description:

The Image information is used for debugging.

Members:

pFrameData: The pointer which points to the image buffer.

dataLen: The length of pFrameData, in bytes.

width: The width of the frame, in pixels.

height: The height of the frame, in pixels.

5.2.2 VzPeopleInfo

Description:

Information about the person being identified from the image.

Members:

headPostion[2]: Pixel coordinates of the center point of the head of the person identified from the image.

distance: The distance of the person identified from the image from the device.

dwell_time: The time a person identified from the image dwells in front of the device.

duration_time: The time a person identified from the image since detected(second).

reserved: reserved Fields

id: identifier of the detected person

5.2.3 VzPeopleInfoCount

Description:

Information about the person being identified from the image.

Members:

peopleInfo[20]: Information(Up to 20) about the person being identified from the image.Reference VzPeopleInfo.

validPeopleCount: The count of people identified from the current image.

dwellPeopleCount: The count of people who stayed longer than the dwell time threshold.

frame: The Image information is used for debugging. Available only if Vz_PCSetShowImg is set to true, otherwise empty.

5.3 API

5.3.1 Vz_PCInitialize

Prototype:

VzReturnStatus Vz_PCInitialize(void)

Description:

Initializes the API on the device. This function must be invoked before any other Vzense APIs.

Parameters:

void

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.2 Vz_PCShutdown

Prototype:

VzReturnStatus Vz_PCShutdown(void)

Description:

Shuts down the API on the device and clears all resources allocated by the API.

After invoking this function, no other Vzense APIs can be invoked.

Parameters:

pDeviceHandler [out]: the handle of the device on which to open.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.3 Vz_PCOpenDevice

Prototype:

VzReturnStatus Vz_PCOpenDevice(VzDeviceHandler* pDeviceHandler)

Description:

Opens device. The device must be subsequently closed using

Vz_CloseDevice().

Parameters:

pDeviceHandler [out]: the handle of the device on which to open.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.4 Vz_PCCloseDevice

Prototype:

VzReturnStatus Vz_PCCloseDevice (VzDeviceHandler* pDeviceHandler)

Description:

Closes the device that was opened using Vz_OpenDevice.

Parameters:

pDeviceHandler [in/out]: The handle of the device to close.After that, *pDeviceHandler will be set to NULL and can no longer be used.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.5 Vz_PCSetMaxDetectDistance

Prototype:

VzReturnStatus Vz_PCSetMaxDetectDistance(const int maxValue)

Description:

Set the furthest detection distance. This function is called before Vz_OpenDevice and can take effect immediately. Called after Vz_OpenDevice, it takes effect the next time the application is started.

Parameters:

maxValue [in]: the furthest detection distance(500mm~5100mm).

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.6 Vz_PCSetCameraHeight

Prototype:

Sets the mounting height of the camera. This function is called before Vz_OpenDevice and can take effect immediately. Called after Vz_OpenDevice, it takes effect the next time the application is started.

Description:

Sets the mounting height of the camera.

Parameters:

height [in]: camera mounting height(1900mm~2100mm).

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.7 Vz_PCSetDwelltimeThreshold

Prototype:

VzReturnStatus Vz_PCSetDwelltimeThreshold(const uint16_t threshold)

Description:

Sets the threshold for person dwell time.

Parameters:

threshold [in]: the threshold for person dwell time.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.8 Vz_PCGetPeopleInfoCount

Prototype:

VzReturnStatus Vz_PCGetPeopleinfoCount(const VzDeviceHandler deviceHandler, VzPeopleInfoCount* pPeopleInfoCount)

Description:

Get information about the person detected.

Parameters:

deviceHandler[in]: The handler of the device on which to get information about the person detected.

pPeopleInfoCount [out]: The pointer to a buffer in which to store information about the person detected.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.9 Vz_PCSetLowpowerModeEnable

Prototype:

VzReturnStatus Vz_PCSetLowpowerModeEnable(const VzDeviceHandler deviceHandler, bool enable)

Description:

Set to enable or disable the low power mode.

Parameters:

deviceHandler [in]: The handler of the device on which to set low power mode.

enable [in]: true to enable the low power mode, false to disable the low power mod.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.10 Vz_PCSetDoorOpenState

Prototype:

VzReturnStatus Vz_PCSetDoorOpenState(const VzDeviceHandler deviceHandler, bool isOpened)

Description:

Switch the open state of the door.

Parameters:

deviceHandler [in]: The handler of the device on which to set low power mode.

isOpened [in]: true means the door is open, false means the door is not open.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.11 Vz_PCRegDeviceHotplugStateCallbackFunc

Prototype:

VzReturnStatus Vz_PCRegDeviceHotplugStateCallbackFunc(const PtrDeviceHotplugStateCallback pCallback, const void* pUserData)

Description:

Registers the Camera State Callback Function.

Parameters:

pCallback [in]: The pointer to the callback function. See ::PtrDeviceHotplugStateCallback.

pUserData [in]: The Pointer to the user data which will be passed to the app via PtrDeviceHotplugStateCallback.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.12 Vz_PCStartUpgradeFirmWare

Prototype:

VzReturnStatus Vz_PCStartUpgradeFirmWare(const VzDeviceHandler deviceHandler, const char* pImgPath)

Description:

Starts upgrading device firmware.

Parameters:

deviceHandler [in]: The handler of the device on which to start upgrading device firmware.

pImgPath [in]: The pointer to the memory device firmware path.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.13 Vz_PCRegUpgradeStateCallbackFunc

Prototype:

VzReturnStatus Vz_PCRegUpgradeStateCallbackFunc(const VzDeviceHandler

deviceHandler, const PtrUpgradeStateCallback pCallback , const void* pUser-Data)

Description:

Registers the callback function to return the device firmware update status.

Parameters:

deviceHandler [in]: The handler of the device on which to return to firmware upgrade status.

pCallback [in]: The pointer to the callback function. See ::PtrUpgradeState-Callback.

pUserData [in]: The Pointer to the user data which will be passed to the app via PtrUpgradeStateCallback.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.14 Vz_PCSetShowImg

Prototype:

VzReturnStatus Vz_PCSetShowImg(bool isShow)

Description:

Sets whether to return images for debugging.

Parameters:

isShow [in]: true means it returns the image, false means it doesn't.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.

5.3.15 Vz_PCSetSaveOfflineDataState

Prototype:

VzReturnStatus Vz_PCSetSaveOfflineDataState(bool isSaved)

Description:

Sets whether to save offline data.

Parameters:

isSaved [in]: true means it save offline data, false means it doesn't.

Returns:

VzRetOK: Succeed

Others: Failed, refer to VzReturnStatus.