



## VIVOTEK NETWORK DEVELOPMENT PLATFORM

fisheyesw  
Version 2.0.0.0  
2014/12/10

© 2014 VIVOTEK Inc. All Right Reserved

VIVOTEK may make changes to specifications and product descriptions at any time, without notice.

The following is trademarks of VIVOTEK Inc., and may be used to identify VIVOTEK products only: VIVOTEK. Other product and company names contained herein may be trademarks of their respective owners.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from VIVOTEK Inc.

# TABLE OF CONTENTS

## 1. Overview

1.1. Introduction .....	3
1.2. Getting Started with fisheyesw Module .....	3
1.3. File Structure .....	3

## 2. Programmer's Guide

2.1. Using fisheyesw Module .....	4
Enable VIVOTEK Fisheye Camera Watermark .....	4
Dewarp using Rectilinear Projection .....	4
Dewarp using Panorama Projection .....	4

## 3. Sample Code

3.1. dewarprect .....	6
3.2. dewarppano .....	7
3.3. dewarpclippano .....	8

## 4. API Reference

4.1. Enumeration .....	10
FEDEWARPTYPE .....	11
FEMOUNTTYPE .....	12
FELENSTYPE .....	13
FEOPTIONFLAG .....	14
FEVPIXELFORMAT .....	16
4.2. Data Structure .....	17
FEOPTION .....	18
FEPOINT20 .....	
FERECT21 .....	
FEVPICTURE .....	22
4.3. API Definition .....	23
Fisheye_Initial .....	24
Fisheye_Release .....	25
Fisheye_SetOption .....	26
Fisheye_OneFrame .....	27
Fisheye_PositionToYawPitchRoll .....	28

# 1. Overview

## 1.1. Introduction

This document describes the properties and methods supported by the VIVOTEK fisheyesw module.

## 1.2. Getting Started with fisheyesw Module

The main purpose of fisheyesw module is to dewarp the video frames which are captured from VIVOTEK fisheye cameras.

## 1.3. File Structure

FILE	DESCRIPTION
\$(SDKPATH)\doc\	This manual document
\$(SDKPATH)\include\	Header files
\$(SDKPATH)\lib\x86\	Binary files and import library files (32-bit)
\$(SDKPATH)\lib\x64\	Binary files and import library files (64-bit)
\$(SDKPATH)\sample\	Sample projects
\$(SDKPATH)\sample\image	Test images

# 2. Programmer's Guide

## 2.1. Using fisheyesw Module

Programmers can use fisheyesw module to dewarp video frames which are captured from VIVOTEK fisheye camera. The fisheyesw module supports both rectilinear and panorama projections as dewarp methods. Programmers can easily combine different dewarp contents to provide flexible layout of scenes in surveillance environment. This module is cross-platform compatible on Windows(x86/x64), MacOS(x86/x64), and Linux(x86/x64).

The minimum system requirement of Windows platform is Windows XP. The minimum system requirement of MacOS is OS X 10.6.

### Enable VIVOTEK Fisheye Camera Watermark

The fisheyesw module only supports video frames that contain VIVOTEK watermark information. The watermark information is composed of white pixels on the bottom left and right corners of the video frame.

If there is no watermark on video frames, please send the following CGI command with VIVOTEK camera web browser to enable watermark:

[http://CAMERA\\_IP/cgi-bin/admin/setparam.cgi?videoin\\_c0\\_enablewatermark=1](http://CAMERA_IP/cgi-bin/admin/setparam.cgi?videoin_c0_enablewatermark=1)

### Dewarp using Rectilinear Projection

The rectilinear projection preserves more geometry characteristics view of objects in the scene. After initializing the fisheyesw module, you will need to choose the mount type of fisheye camera. Set FE\_DEWARP\_RECTILINEAR as the dewarp type and you can get the dewarp scene using rectilinear projection.

### Dewarp using Panorama Projection

The panorama projection provides near-complete view in the scene. After initializing the fisheyesw module, you will need to choose the mount type of fisheye camera. Set FE\_DEWARP\_FULLVIEWPANORAMA as the dewarp type and you can get the dewarp scene using panorama projection. Moreover, the fisheyesw module provides a dual-view panorama projection in ceiling and floor mount type. You can use DEWARP\_DUALVIEWPANORAMA as the as the dewarp type to get the dual-view panorama.

# 3. Sample Code

This section contains sample projects provided by fisheyesw.

To demonstrate how to use fisheyesw library and the fisheye ePTZ control, we use Simple DirectMedia Layer(SDL2) library to access keyboard and mouse event. Also, we use it to present the dewarpe result on the window. As a result, please check the SDL2 library is available in your platform before running these samples.

All samples are cross-platform on Windows, MacOS and Linux, but only support x86 platform.

## How to Run the Sample

1. In Windows, launch Visual Studio and open the \$(SAMPLENAME).sln solution under \$(SDKPATH)\sample\\$(SAMPLENAME). Build the solution and run the sample.
2. In Mac OS, launch XCode and open \$(SAMPLENAME).xcodeproj under \$(SDKPATH)\sample\\$(SAMPLENAME). Build the solution and run the sample.
3. In Linux, locate the folder to \$(SDKPATH)\sample\\$(SAMPLENAME)\vcproj and make the file then type“./build.sh”. The sample will run after being successfully built. Pease check the SDL2 library is available in your platform before running these samples.

## 3.1. dewarprect

### DESCRIPTION

This sample loads a Bitmap image and displays the rectilinear dewarped scene in client window It also shows how to use keyboard and mouse event to perform ePTZ control.

### SAMPLE CODE

#### STEP 1. Initialize fisheyesw module

```
SCODE scRet = Fisheye_Initial(&hFEctx, LIBFISHEYE_VERSION);
```

#### STEP 2. Set the options

In this sample, we use a SDL surface from Bitmap as the input and allocate a output buffer. First of all, initialize the values of fisheye option and set default value of Scale field with 100 which means 1x zoom factor. Secondly, we set the mount type of fisheye camera and FE\_DEWARP\_RECTILINEAR as the dewarp type. After fill these fields in option, you need to set the option flags to indicate what specific fields are filled with new values. Finally, use Fisheye\_SetOption to update the option.

#### STEP 3. Event loop

In the event loop, this sample use keyboard and mouse event to control the Yaw, Pitch, Roll and Scale. After applying these parameters via Fisheye\_SetOption, use Fisheye\_Oneframe to dewarp the input image and update the dewarped scene to the output buffer. Here, in order to show the dewarped scene, we update SDL texture with new output pixel data and use SDL render to present the scene on client window.

#### STEP 4. Release fisheyesw module

```
Fisheye_Release(&hFEctx);
```

### TIPS

1. The pixel format must be the same in InVPicture and OutVPicture.
2. Before first Fisheye\_OneFrame is called, InVPicture, OutVPicture, FOVCenter, FOVRadius and OutROI fields must be set with valid values.
3. Scroll filed is not available in rectilinear projection.

## 3.2. dewarppano

### DESCRIPTION

This sample loads a Bitmap image and displays the panorama dewarped scene in client window It also shows how to use keyboard to perform ePTZ control.

### SAMPLE CODE

#### STEP 1. Initialize fisheyesw module

```
SCODE scRet = Fisheye_Initial(&hFECtx, LIBFISHEYE_VERSION);
```

#### STEP 2. Set the options

In this sample, we use a SDL surface from Bitmap as the input and allocate a output buffer. First of all, we initialize the values of fisheye option and set default value of Sacle field with 100 which means 1x zoom factor. Secondly, we set the mount type of fisheye camera and FE\_DEWARP\_FULLVIEWPANORAMA as the dewarp type. After fill these fields in option, you need to set the option flags to indicate what specific fields are filled with new values. Finally, use Fisheye\_SetOption to update the option.

#### STEP 3. Event loop

In the event loop, this sample use keyboard event to control the Scroll. After applying these parameters via Fisheye\_SetOption, use Fisheye\_Oneframe to dewarp the input image and update the dewarped scene to the output buffer. Here, in order to show the dewarped scene, we update SDL texture with new output pixel data and use SDL render to present the scene on client window.

#### STEP 4. Release fisheyesw module

```
Fisheye_Release(&hFECtx);
```

### TIPS

1. The pixel format must be the same in InVPicture and OutVPicture.
2. Before first Fisheye\_OneFrame is called, InVPicture, OutVPicture, FOVCenter, FOVRadius and OutROI fields must be set with valid values.
3. Yaw, Pitch, Roll, Scale fields are not available in panorama projection.
4. Scroll control is only available in ceiling and floor mount type.

## 3.3. dewarpclippano

### DESCRIPTION

This sample loads a Full-HD Bitmap image and displays the clip panorama dewarped scene in client window. The ePTZ control is not available in this dewarp type.

### SAMPLE CODE

#### STEP 1. Initialize fisheyesw module

```
SCODE scRet = Fisheye_Initial(&hFECtx, LIBFISHEYE_VERSION);
```

#### STEP 2. Set the options

In this sample, we use a SDL surface from Bitmap as the input and allocate a output buffer. First of all, we initialize the values of fisheye option and set default value of Sacle field with 100 which means 1x zoom factor. Secondly, we set the mount type of fisheye camera and FE\_DEWARP\_CLIPVIEWPANORAMA as the dewarp type. After fill these fields in option, you need to set the option flags to indicate what specific fields are filled with new values. Finally, use Fisheye\_SetOption to update the option.

#### STEP 3. Event loop

After applying these parameters via Fisheye\_SetOption, use Fisheye\_Oneframe to dewarp the input image and update the dewarped scene to the output buffer. Here, in order to show the dewarped scene, we update SDL texture with new output pixel data and use SDL render to present the scene on client window.

#### STEP 4. Release fisheyesw module

```
Fisheye_Release(&hFECtx);
```

### TIPS

1. The pixel format must be the same in InVPicture and OutVPicture.
2. Before first Fisheye\_OneFrame is called, InVPicture, OutVPicture, FOVCenter, FOVRadius and OutROI fields must be set with valid values.
3. Yaw, Pitch, Roll, Scale and Scroll fields are not available in clip panorama projection.
4. The Width and Height fields of InVPicture must be 1920 and 1080.
5. Users can obtain Full HD streaming from VIVOTEK fisheye camera by setting Media->Video->FOV as 1080P Full HD.

# 4. API Reference

This chapter contains the API function and elements provided by fisheyesw.

VIVOTEK CONFIDENTIAL  
2014.12.10

## 4.1. Enumeration

This section depicts the fisheyesw enumerations.

- FEDEWARPTYPE
- FEMOUNTTYPE
- FELENSTYPE
- FEOPTIONFLAG
- FEVPIXELFORMAT

VIVOTEK CONFIDENTIAL  
2014.12.10

## FEDEWARPTYPE

This enumeration specifies the dewarp type.

```
typedef enum _FisheyeDewarpType{
    FE_DEWARP_RECTILINEAR,
    FE_DEWARP_FULLVIEWPANORAMA,
    FE_DEWARP_DUALVIEWPANORAMA,
    FE_DEWARP_CLIPVIEWPANORAMA
} FEDEWARPTYPE;
```

### Values

#### **FE\_DEWARP\_RECTILINEAR**

Specifies the dewarp type is rectilinear projection.

#### **FE\_DEWARP\_FULLVIEWPANORAMA**

Specifies the dewarp type is panorama projection. In ceiling and floor mount, the output is a single-view content with 360 degrees in horizontal. In wall mount, the output is a single-view content with 180 degrees in horizontal.

#### **FE\_DEWARP\_DUALVIEWPANORAMA**

Specifies the dewarp type is panorama projection. In ceiling and floor mount, the output is a dual-view content with 180 degrees per view in horizontal. In wall mount, the output is the same as FE\_DEWARP\_FULLVIEWPANORAMA.

#### **FE\_DEWARP\_CLIPVIEWPANORAMA**

Specifies the dewarp type is clipped panorama projection. The output is a single-view content with 180 degrees in horizontal.

### Remarks

The FE\_DEWARP\_CLIPVIEWPANORAMA is a specific dewarp type for the 1080P Full HD streaming of VIVOTEK fisheye camera. FE\_DEWARP\_CLIPVIEWPANORAMA is similar to FE\_DEWARP\_FULLVIEWPANORAMA in wall mount. However, it is designed only for the image which comes from Full HD streaming. Users should provide the Full HD video streaming to get correct dewarped result. The ePTZ control will be disabled in FE\_DEWARP\_CLIPVIEWPANORAMA.

### Requirements

fisheyesw.h

## FEMOUNTTYPE

This enumeration specifies the mount type of the installed fisheye camera.

```
typedef enum _FisheyeMountType{
    FE_MOUNT_WALL,
    FE_MOUNT_CEILING,
    FE_MOUNT_FLOOR,
} FEMOUNTTYPE;
```

### Values

#### FE\_MOUNT\_WALL

Specifies the mount type is wall-mounted. The horizontal and vertical field of view (FOV) are 180 degrees.

#### FE\_MOUNT\_CEILING

Specifies the mount type is ceiling-mounted. The horizontal FOV 360 degrees and vertical FOV is 90 degrees.

#### FE\_MOUNT\_FLOOR

Specifies the mount type is floor-mounted. The horizontal FOV 360 degrees and vertical FOV is 90 degrees.

### Remarks

### Requirements

fisheyesw.h

## FELENSTYPE

This enumeration indicates the VIVOTEK fisheye lens type.

```
typedef enum _FisheyeLensType{
    FE_LENSTYPE1,
    FE_LENSTYPE2,
    FE_LENSTYPE3,
    FE_LENSTYPE4,
    FE_LENSTYPE5,
    FE_LENSTYPE6
} FELENSTYPE;
```

### Values

#### FE\_LENSTYPE1

Specifies the FE8171 fisheye camera.

#### FE\_LENSTYPE2

Specifies the FE8172 or SF8172 fisheye camera.

#### FE\_LENSTYPE3

Specifies the FE8173 fisheye camera.

#### FE\_LENSTYPE4

Specifies the FE8174, SF8174, FE8181 fisheye camera.

#### FE\_LENSTYPE5

Specifies the CU8171 fisheye camera.

#### FE\_LENSTYPE6

Specifies the FE8180 fisheye camera.

### Remarks

This field is a optional option. However, each VIVOTEK fisheye camera is related to one of the lens type. Users can set FELENSTYPE to obtain more geometry characteristics view of objects

### Requirements

fisheyesw.h

## FEOPTIONFLAG

This enumeration is a set of bit flags that specify options in the fisheyesw module.

```
typedef enum _FisheyeOptionFlag{
    FE_OPTION_INIMAGEHEADER = (1 << 0),
    FE_OPTION_INIMAGEBUFFER = (1 << 1),
    FE_OPTION_OUTIMAGEHEADER = (1 << 2),
    FE_OPTION_OUTIMAGEBUFFER = (1 << 3),
    FE_OPTION_FOVCENTER = (1 << 4),
    FE_OPTION_FOVRADIUS = (1 << 5),
    FE_OPTION_MOUNTTYPE = (1 << 6),
    FE_OPTION_DEWARPTYPE = (1 << 7),
    FE_OPTION_YAW = (1 << 8),
    FE_OPTION_PITCH = (1 << 9),
    FE_OPTION_ROLL = (1 << 10),
    FE_OPTION_SCALE = (1 << 11),
    FE_OPTION_SCROLL = (1 << 12),
    FE_OPTION_OUTROI = (1 << 13),
    FE_OPTION_LENSTYPE = (1 << 14)
} FEOPTIONFLAG;
```

### Values

#### **FE\_OPTION\_INIMAGEHEADER**

Specifies the InVPicture structure's header fields.

#### **FE\_OPTION\_INIMAGEBUFFER**

Specifies the InVPicture structure's buffer fields.

#### **FE\_OPTION\_OUTIMAGEHEADER**

Specifies the OutVPicture structure's header fields.

#### **FE\_OPTION\_OUTIMAGEBUFFER**

Specifies the OutVPicture structure's buffer fields.

#### **FE\_OPTION\_FOVCENTER**

Specifies the center in the field of view(FOV).

#### **FE\_OPTION\_FOVRADIUS**

Specifies the radius of the field of view(FOV).

#### **FE\_OPTION\_MOUNTTYPE**

Specifies the mount type.

#### **FE\_OPTION\_DEWARPTYPE**

Specifies the dewarp type.

#### **FE\_OPTION\_YAW**

Specifies the yaw angle.

#### **FE\_OPTION\_PITCH**

Specifies the pitch angle.

**FE\_OPTION\_ROLL**

Specifies the roll angle.

**FE\_OPTION\_SCALE**

Specifies the scaling factor.

**FE\_OPTION\_SCROLL**

Specifies the scroll angle.

**FE\_OPTION\_OUTROI**

Specifies the output region-of-interest(ROI).

**FE\_OPTION\_LENSTYPE**

Specifies the lens type.

**Remarks**

You can combine the bit flags using bitwise-OR.

**Requirements**

fisheyesw.h

**See Also**

FEOPTION

## FEVPIXELFORMAT

This enumeration specifies the color format for each pixel in the picture.

```
typedef enum _FisheyVPixelFormat{
    FE_PIXELFORMAT_YUV420P,
    FE_PIXELFORMAT_RGB32,
} FEVPICTUREFORMAT;
```

### Values

#### FE\_PIXELFORMAT\_YUV420P

Specifies the format is a YUV4:2:0 planar format. Y, U and V components are grouped together.

#### FE\_PIXELFORMAT\_RGB32

Specifies the format is 32bit per pixel; each 8 bits are used in red, green and blue components. The rest 8bits are not used.

### Remarks

### Requirements

fisheyesw.h

## 4.2. Data Structure

This section depicts the fisheyesw structure.

- FEOPTION
- FEPOINT
- FERECT
- FEVPICTURE

VIVOTEK CONFIDENTIAL  
2014.12.10

## FEOPTION

This structure specifies options of fisheyesw module.

```
typedef struct _FisheyeOption{
    DWORD Flags;
    FEVPICTURE InVPicture;
    FEVPICTURE OutVPicture;
    FEPOINT FOVCenter;
    unsigned int FOVRadius;
    FEMOUNTTYPE MountType;
    FEDEWARPTYPE DewarpType;
    float Yaw;
    float Pitch;
    float Roll;
    unsigned int Scale;
    float Scroll;
    FERECT OutRoi;
    FELENSTYPE LensType;
} FEOPTION;
```

### Members

#### Flags

Specifies the option flags. You can combine the bit flags which are defined in FEOPTIONFLAG using bitwise-OR.

#### InVPicture

Specifies fields to describe the input picture.

#### OutVPicture

Specifies fields to describe the output picture.

#### FOVCenter

Specifies the pixel coordinate of the center in the field of view(FOV) of the input picture.

#### FOVRadius

Specifies the radius of FOV in the input picture.

#### MountType

Specifies the mount type of fisheye camera. The values are defined in FEMOUNTTYPE.

#### DewarpType

Specifies the mount type of fisheye camera. The values are defined in FEDEWARPTYPE. If FE\_DEWARP\_DUALVIEWPANORAMA is used, it only takes effect in FE\_MOUNT\_CEILING and FE\_MOUNT\_FLOOR.

#### Yaw

Specifies the yaw angle(in degree) around the y-axis. It is only available in FE\_MOUNT\_WALL.

#### Pitch

Specifies the pitch angle(in degree) around x-axis.

### **Roll**

Specifies the pitch angle(in degree) around z-axis, which is straight toward the screen. It is available in FE\_MOUNT\_CEILING and FE\_MOUNT\_FLOOR.

### **Scale**

Specifies the scaling factor. The default value is 100, which means that 1x zoom and no scaling. You can use a number which is larger than 100 to perform scaling. For example, passing 120 will produce a 1.2x zoom-in effect. The minimum and maximum values are 70 and 1200. Scale is available in FE\_DEWARP\_RECTILINEAR.

### **Scroll**

Specifies the scroll offset (in degree) in the panorama projection. The default value is 0, which means no offset. You can adjust this parameter to change the start position of panorama projection.

### **OutRoi**

Specifies the destination rectangle in output picture.

### **LensType**

Specifies the lens type of the fisheye camera.

## **Remarks**

Users need to handle the input and output buffer pointers. Before updating the buffer pointers in InVPicture and OutVPicture, you need to allocate buffers with sufficient size. When leaving the program, free the above buffers by yourselves. The pixel format must be the same in InVPicture and OutVPicture

The FOV means the circular region of the input picture which is captured by VIVOTEK fisheye lens. When the width and height of input picture are changed, you need to update the FOV's center and radius.

Yaw, Pitch, Roll and Scale are used for ePTZ control in rectilinear projection. Scroll is used for sliding control in panorama projection. The minimum and maximum values of these parameters are controlled by fisheyesw module.

If you want to display several dewarp results in one view, you can use OutRoi to control each output coordinate.

## **Requirements**

fisheyesw.h

## FEPOINT

This structure specifies a point in 2-D coordinate.

```
typedef struct _FisheyPoint{  
                                int X;  
                                int Y;  
} FEPOINT;
```

### Members

#### X

Specifies the x-coordinate of the point.

#### Y

Specifies the y-coordinate of the point.

### Remarks

### Requirements

fisheyesw.h

## FERECT

This structure specifies a rectangle in 2-D coordinate.

```
typedef struct _FisheyRect{
    int Left;
    int Top;
    int Right;
    int Bottom;
} FERECT;
```

### Members

#### Left

Specifies the x-coordinate of the upper-left point of the rectangle.

#### Top

Specifies the y-coordinate of the upper-left point of the rectangle.

#### Right

Specifies the x-coordinate of the lower-right point of the rectangle.

#### Bottom

Specifies the y-coordinate of the lower-right point of the rectangle.

### Remarks

### Requirements

fisheyesw.h

## FEVPICTURE

This structure specifies fields to describe a picture.

```
typedef struct _FisheyeVPicture{
    unsigned int Width;
    unsigned int Height;
    unsigned int Stride;
    FEVPIXELFORMAT Format;
    BYTE* Buffer;
} FEVPICTURE;
```

### Members

#### Width

Specifies the width (in pixels) of the picture.

#### Height

Specifies the height (in pixels) of the picture..

#### Stride

Specifies the number of bytes in one row of the picture.

#### Format

Specifies the picture format, which is one of values defined in FEVPIXELFORMAT.

#### Buffer

Specifies the pointer of data bits.

### Remarks

### Requirements

fisheyesw.h

### 4.3. API Definition

This section depicts the fisheyesw APIs.

- Fisheye\_Initial
- Fisheye\_Release
- Fisheye\_SetOption
- Fisheye\_OneFrame
- Fisheye\_PositionToYawPitchRoll

VIVOTEK CONFIDENTIAL  
2014.12.10

## Fisheye\_Initial

Call this function to initialize the fisheyesw module.

### Syntax

```
SCODE Fisheye_Initial {  
                                     HANDLE *phObject,  
                                     DWORD dwVersion  
};
```

### Parameters

#### phObject

[out] Pointer to the handle of fisheye object.

#### dwVersion

[in] The version of fisheyesw library. USE LIBFISHEYE)VERSION as the parameter.

### Return Values

#### FISHEYE\_S\_OK

Initialize the fisheyesw library successfully.

#### FISHEYE\_E\_INVALID\_ARG

The pointer to handle of fisheye object is NULL.

#### FISHEYE\_E\_OUT\_OF\_MEMORY

The library can not acquire enough memory space.

#### FISHEYE\_E\_FAIL

Fail to initialize the fisheyesw library.

### Remarks

### Requirements

fisheyesw.h fisheyesw\_errordef.h

### See Also

## Fisheye\_Release

Call this function to release the fisheyesw module.

### Syntax

```
SCODE Fisheye_Release {  
                                HANDLE *phObject,  
};
```

### Parameters

#### phObject

[in] Pointer to the handle of fisheye object.

### Return Values

#### FISHEYE\_S\_OK

Initialize the fisheyesw library successfully.

#### FISHEYE\_E\_INVALID\_ARG

The pointer to handle of fisheye object is NULL.

#### FISHEYE\_E\_INVALID\_HANDLE

The handle of fisheye object is NULL.

### Remarks

### Requirements

fisheyesw.h fisheyesw\_errordef.h

### See Also

## Fisheye\_SetOption

Call this function to set option fields.

### Syntax

```
SCODE Fisheye_SetOption {  
                                HANDLE hObject,  
                                FEOPTION *pOption  
                                };
```

### Parameters

#### hObject

[in] The handle of fisheye object.

#### pOption

[in/out] Pointer to FEOPTION which contains the option fields.

### Return Values

#### FISHEYE\_S\_OK

Set option fields successfully.

#### FISHEYE\_E\_INVALID\_HANDLE

The handle of fisheye object is NULL.

#### FISHEYE\_E\_NOT\_SUPPORT\_PTZ

PTZ parameters (which are Yaw, Pitch, Roll, Scale and Scroll) are not supported in current dewarp mode (e.g. FE\_DEWARP\_CLIPVIEWPANORAMA).

#### FISHEYE\_E\_INVALID\_ARG

The pointer to FEOPTION is NULL. One or more option fields are invalid, see Remarks section.

#### FISHEYE\_E\_FAIL

Fail to set option fields, see Remarks section.

### Remarks

If invalid mount type or dewarp type is set, the library returns FISHEYE\_E\_INVALID\_ARG.

If invalid picture's color format or NULL data buffer is set, the library returns FISHEYE\_E\_FAIL.

### Requirements

fisheyesw.h, fisheyesw\_errordef.h

### See Also

## Fisheye\_OneFrame

Call this function to dewarp the fisheye picture.

### Syntax

```
SCODE Fisheye_OneFrame {  
                                HANDLE hObject,  
};
```

### Parameters

#### hObject

[in] The handle of fisheye object.

### Return Values

#### FISHEYE\_S\_OK

Dewarp successfully.

#### FISHEYE\_E\_INVALID\_HANDLE

The handle of fisheye object is NULL.

#### FISHEYE\_E\_NOT\_INITIALIZED\_OPTION

One or more essential option fields are not provided. See Remarks section.

#### FISHEYE\_E\_WATERMARK\_CHECK\_FAIL

Fail to identify proprietary VIVOTEK watermark information. See 2.1 Enable VIVOTEK Fisheye Camera Watermark.

#### FISHEYE\_E\_FAIL

Fail to dewarp.

### Remarks

The library need to calculate internal parameters using the information in essential option fields. These options are InVPicture, OutVPicture, FOVCenter, FOVRadius and OutRoi. Before first Fisheye\_OneFrame is called, these fields must have been set with valid values.

### Requirements

fisheyesw.h, fisheyesw\_errordef.h

### See Also

## Fisheye\_PositionToYawPitchRoll

Call this function to get the Yaw, Pitch and Roll which are mapped from fisheye picture.

### Syntax

```
void Fisheye_PositionToYawPitchRoll(  
    FEPOINT FOVCenter,  
    long FOVRadius,  
    FEMOUNTTYPE MountType,  
    int X,  
    int Y,  
    float *pYaw,  
    float *pPitch,  
    float *pRoll  
);
```

### Parameters

#### FOVCenter

[in] The center of fisheye circular region.

#### FOVRadius

[in] The radius of fisheye circular region.

#### MountType

[in] Specifies the mount type of fisheye camera. You can use one of the values defined in FEMOUNTTYPE.

#### X

[in] The x-coordinate in fisheye image.

#### Y

[in] The y-coordinate in fisheye image.

#### pYaw

[out] The yaw angle. This parameter is valid only when the MountType is FE\_MOUNT\_WALL.

#### pPitch

[out] The pitch angle.

#### pRoll

[out] The roll angle. This parameter is valid only when the MountType is FE\_MOUNT\_CEILING and FE\_MOUNT\_FLOOR.

### Return Values

N/A

## Remarks

This function is available for Windows, Mac and Linux platforms. This function is a standalone utility to retrieve the relative Yaw, Pitch and Roll according to the position(X,Y) of fisheye image. You need to provide the values of FOVCenter, FOVRadius, X and Y based on the target width and height. Yaw, pitch and roll are used for ePTZ control in rectilinear projection. The minimum and maximum values of these parameters are controlled by fisheyesw module. You can directly set the values of yaw, pitch and roll which are get from Fisheye\_PositionToYawPitchRoll.

## Requirements

fisheyesw.h

## See Also

Example: ( 768x768 fisheye image )

```
FEPOINT center = {384, 384};
```

```
int radius = 384;
```

```
FEMOUNTTYPE type = FE_MOUNT_CEILING;
```

```
int X = 499, Y = 277;
```

```
FEOPTION option = {0};
```

```
Fisheye_PositionToYawPitchRoll(center, radius, type, X, Y,  
&option.Yaw, &option.Pitch, &option.Roll);
```

```
option.Flags = (FE_OPTION_ROLL | FE_OPTION_PITCH);
```

```
Fisheye_SetOption(hObject, pOption);
```

```
Fisheye_OneFrame(hObject);
```