FishEye-SDK
Library Edition
V1.02.20

API Reference Guide

Table of Contents

1 OVERVIEW

Introduction

The SDK can help with application for transforming fisheye image to calibration image.

Start Up with FISHEYE library

Following is a scenario of an application.

Open the Interface

HANDLE myFISHEYE = FishEye_OpenInterface();

Then the application can use the handle to using the SDK function.

Prepare structures

There are some structures need to be prepared after using the interface.

EN_FISHEYE_MODE : for choose Fisheye Transformable Type

ST_FISHEYE_PARAMETER: Set up Fisheye Transformation Parameter

ST_WALL_D: Wall De-warping Parameter

ST_WALL_P : Wall Panorama Parameter

ST_CEILING_D: Ceiling De-warping Parameter

ST_CEILING_DP: Ceiling Double Panorama Parameter

ST_FISHEYE_IMGINFO: Input Image Information

ST_FISHEYE_OUT: Output Result information

ST_FISHEYE_COORD : Get Coordinate from De-warping

FishEye_Initial

```
int CAMERA = MODEL_NO_SPECIFIC;
bool AutoCenter = true;

If(myFISHEYE)
{
	FishEye_Initial(FISHEYE, CAMERA, AutoCenter);
}
```

FishEye_Transform

```
EN_FISHEYE_MODE	FisheyeMode = enWall_Dewarping;
ST_FISHEYE_IMGINFO	ImgInfo;
ST_FISHEYE_PARAMETER	Par;
ST_FISHEYE_OUT		FisheyeOut;

If(myFishEye)
{
	FishEye_Transform(myFISHEYE, FisheyeMode, ImgInfo, Par , FisheyeOut);
}
```

FishEye_GetCoordinate

```
ST_FISHEYE_IMGINFO	ImgInfo;
ST_FISHEYE_COORD	Coord;
ST_FISHEYE_COORD_MODE		Mode
ST_FISHEYE_PARAMETER	Par;
ST_FISHEYE_OUT		FisheyeOut;

If(myFishEye)
{
	FishEye_GetCoordinate(myFishEye, Mode, ImgInfo, Coord, Par, FisheyeOut);
}
```

FishEye_GetCircle

```
int CenterXout
int CenterYout
int RadiusOut
int ImgWidthOut
int ImgHeightOut

If(myFishEye)
{
    FishEye_GetCircle(myFishEye, CenterXout, CenterYout, RadiusOut, ImgWidthOut,
    ImgHeightOut);
}
```

FishEye_UserSetCircle

```
int UserCenterX
int UserCenterY
int UserRadius
int ImgWidth
int ImgHeight

If(myFishEye)
{
    FishEye_UserSetCircle(myFishEye, UserCenterX, UserCenterY, UserRadius,
    ImgWidth, ImgHeight);
}
```

FishEye_Release

```
If(myFishEye)
{
    FishEye_Release(FISHEYE);
}
```

FishEye_CloseInterface

```
If(myFishEye)
{
	FishEye_CloseInterface(FISHEYE);
}
```

API Architectures

Step1:

#include "FishEyeSDK.h"

#include "FishEyeBase.h"

Step2:

Choose a lens module

int Camera = MODEL_NO_SPECIFIC;

Create an Image information object

ST_FISHEYE_IMGINFO ImgInfo;

Step3:

Create fisheye object

HANDLE FISHEYE = FishEye_OpenInterface();

Initialize fisheye object

FishEye_Initial(FISHEYE, Camera, AutoCenter);

Step4:

Set parameter of chosen transformable fisheye mode

ST_FISHEYE_PARAMETER Par;

Set transformable Fisheye Mode

EN_FISHEYE_MODE FishEyeMode = en(Mode);

Get Circle information

FishEye_GetCircle(myFishEye, CenterXout, CenterYout, RadiusOut, ImgWidthOut, ImgHeightOut);

Set User defined Circle

FishEye_UserSetCircle(myFishEye, UserCenterX, UserCenterY, UserRadius, ImgWidth, ImgHeight);

Get Transform coordinate

FishEye_GetCoordinate(FISHEYE, Mode, ImgInfo, Coord, Par, Fisheye_out);

Fisheye Transform

FishEye_Transform(FISHEYE, FishEyeMode, ImgInfo, Par, Fisheye_out);

Step5:

Release

FishEye_Release(FISHEYE);

Setp6:

Delete fisheye object

FishEye_CloseInterface(FISHEYE);

What’s New in this release

Ver1.02.20

2 Data Structure

EN_FISHEYE_LENS

Here are definitions of the fisheye camera models for APIs. With specific camera models, the APIs perform more précised dewarp calculation for panorama and ePTZ views. You could use MODEL_NO_SPECIFIC to have generic dewarp calculation for all types of fisheye cameras.

```
#define MODEL_FCS_3094          3
#define MODEL_FCS_3092          5
#define MODEL_FCS_3093          7
#define MODEL_NO_SPECIFIC       999
```

EN_FISHEYE_MODE

```
enum EN_FISHEYE_MODE
{
    enWall_Dewarping        = 0,
    enWall_Panorama         = 1,
    enCeiling_Dewarping     = 2,
    enCeiling_Panorama      = 3,
    enCeiling_DoublePanorama  = 4,
};
```

Members

Fisheye Mode	Description
enWall_Dewarping	Wall De-warping Mode
enWall_Panorama	Wall Panorama Mode
enCeiling_Dewarping	Ceiling De-warping Mode
enCeiling_Panorama	Ceiling Panorama Mode
enCeiling_DoublePanorama	Ceiling Double Panorama Mode

EN_FISHEYE_COORD_MODE

```
enum EN_FISHEYE_COORD_MODE
{
    enWall_Panorama2PTZ        = 0,
    enCeiling_Panorama2PTZ     = 1,
    enCeiling_Fisheye2PTZ      = 2,
    enwall_Fisheye2PTZ         = 3,
    enwall_Dewarping2PTZ       =4,
    enCeiling_Dewarping2PTZ    =5,
};
```

Members

Fisheye Mode	Description
enWall_Panorama2PTZ	Wall Panorama Coordinate to PTZ Coordinate
enCeiling_Panorama2PTZ	Ceiling Panorama Coordinate to PTZ Coordinate
enCeiling_Fisheye2PTZ	Fisheye coordinate to ceiling PTZ
enwall_Fisheye2PTZ	Fisheye coordinate to wall PTZ
enwall_Dewarping2PTZ	Wall Dewarping coordinate to wall PTZ
enCeiling_Dewarping2PTZ	Ceiling Dewarping coordinate to ceiling PTZ

ST_FISHEYE_IMGINFO

```
struct ST_FISHEYE_IMGINFO
{
    unsigned char* lpSrc;
    unsigned char* lpDst;
    int ImgWidth;
    int ImgHeight;
    int ImgWidthOut;
    int ImgHeightOut;
    int BitPerPixel;
    bool YUV;
};
```

Members

Parameter	Range	Default Value	Description
lpSrc	N/A	!= NULL	Input Image Buffer
lpDst	N/A	!= NULL	Output Image Buffer

ImgWidth	320~4096	N/A	Input Image Width
ImgHeight	240~4096	N/A	Input Image Height
ImgWidthOut	32~1920	N/A	Output Image Width
ImgHeightOut	24~1080	N/A	Output Image Height
BitPerPixel	24/32	24	Bits Per Pixel
YUV	0/1	N/A	0:RGB image ; 1:YUV image

Note: if YUV is used, Input/Output ImgWidth/Height must be even number

ST_FISHEYE_COORD

```
struct ST_FISHEYE_COORD
{
    int InputRectangleCoordX;
    int InputRectangleCoordY;
    int TransformTilt;
    int TransformPan;
    int OriginalCoordX;
    int OriginalCoordY;
};
```

Members

Parameter	Range	Default Value	Description
InputRectangleCoordX	-RectangleWidth/2 ~ RectangleWidth/2	0	X Coordinate from Rectangle Image
InputRectangleCoordY	-RectangleHeight/2 ~ RectangleHeight/2	0	Y Coordinate from Rectangle Image
TransformTilt	Tilt range	0	De-warping coordinate from transformation of Panorama/original image
TransformPan	Pan range	0	De-warping coordinate from transformation of Panorama/original image
OriginalCoordX	-ImgWidth/2~ImgWidth/2	0	Fisheye Original Image Coordinate X
OriginalCoordY	-ImgHeight/2~ImgHeight/2	0	Fisheye Original Image Coordinate Y

ST_WALL_D

```
struct ST_WALL_D
{
    int Pan;
    int Tilt;
    int Zoom;
};
```

Members

Parameter	Range	Default Value	Description
Pan	-16384~16384	0	Absolute pan angle replace [-90, 90] with [-16384, 16384]
Tilt	-16384~16384	0	Absolute tilt angle replace [-90, 90] with [-16384, 16384]
Zoom	32~1024	256	Absolute Zoom scale 256 = 1scale, 128=2scale, 64=4scale, 32=8scale

ST_WALL_P

```
struct ST_WALL_P
{
    double ViewX;
    double ViewY;
    int Tilt;
};
```

Members

Parameter	Range	Default Value	Description
ViewX	0.3~1.0	1.0	View Range of X-axis
ViewY	0.3~1.0	1.0	View Range of Y-axis
Tilt	-8192~8192	0	Absolute tilt angle replace [-45, 45] with [-8192, 8192]

ST_CEILING_D

```
struct ST_CEILING_D
{
    int Pan;
    int Tilt;
    int Zoom;
};
```

Members

Parameter	Range	Default Value	Description
Pan	-65536~65536	0	Absolute pan angle replace [-360, 360] with [-65536, 65536]
Tilt	0~16384	0	Absolute tilt angle replace [0, 90] with [0, 16384]
Zoom	32~1024	256	Absolute Zoom scale 256 = 1scale, 128=2scale, 64=4scale, 32=8scale

ST_CEILING_P

```
struct ST_CEILING_P
{
    double View;
    int Pan;
};
```

Parameter	Range	Default Value	Description
View	0.5~1.0	1.0	View Range of Y-axis
Pan	-65536~65536	0	Absolute pan angle replace [-360, 360] with [-65536, 65536]

ST_CEILING_DP

```
struct ST_CEILING_DP
{
	int Move_XP;
	double View;
	int ExtendCenterView;
};
```

Parameter	Range	Default Value	Description
Move_XP	-65536~65536	0	X Coordinate from Double Panorama Image
View	0.5~1.0	1.0	View Range of Y-axis
ExtendCenterView	0~90	0	Extension for view range at center

ST_GROUND_D

```
struct ST_GROUND_D
{
	int Pan;
	int Tilt;
	int Zoom;
};
```

Members

Parameter	Range	Default Value	Description
Pan	-65536~65536	0	Absolute pan angle replace [-360, 360] with [-65536, 65536]
Tilt	0~16384	0	Absolute tilt angle replace [0, 90] with [0, 16384]
Zoom	32~1024	256	Absolute Zoom scale 256 = 1scale, 128=2scale, 64=4scale, 32=8scale

ST_ GROUND_P

```
struct ST_ GROUND_P
{
    double View;
    int Pan;
};
```

Parameter	Range	Default Value	Description
View	0.5~1.0	1.0	View Range of Y-axis
Pan	-65536~65536	0	Absolute pan angle replace [-360, 360] with [-65536, 65536]

ST_ GROUND_DP

```
struct ST_ GROUND_DP
{
    int Move_XP;
    double View;
    int ExtendCenterView;
};
```

Parameter	Range	Default Value	Description
Move_XP	-65536~65536	0	X Coordinate from Double Panorama Image
View	0.5~1.0	1.0	View Range of Y-axis
ExtendCenterView	0~90	0	Extension for view range at center

ST_RECTANGLE

```
Struct ST_RECTANGLE
{
    Int RectNumber;
```

```
    Int RectStartX;
    Int RectStartY;
    Int RectWidth;
    Int RectHeight;
}
```

Parameter	Range	Default Value	Description
RectNumber	0~63	0	The number of rectangle image
RectStartX	0~image_outX0-1	0	Start position x-coordiante of rectangle image
RectStartY	0~image_outY-1	0	Start position x-coordiante of rectangle image
RectWidth	0~image_outX	0	Rectangle width
RectHeight	0~image_outY	0	Rectangle height

ST_FISHEYE_OUT

```
Struct ST_FISHEYE_OUT
{
    Int Real_Pan;
    Int Real_Tilt;
    Int Real_Zoom;
    Int* Boundary_Coordinate_X
    Int* Boundary_Coordinate_Y;
    int Boundary_Count;
};
```

Parameter	Range	Default Value	Description
Real_Pan	Wall Range	N/A	It almost the same with input pan except in the image boundary. It show the real angle view
Real_Tilt	Ceiling/Wall Range	N/A	It almost the same with input tilt except in the image boundary. It show the real angle view
Real_Zoom	32~2048	N/A	It almost the same with input

			zoom except in the image boundary. It show the real angle view
Boundary_Coordinate_X	0~-ImgWidth	!NULL	X-axis Coordinate in original fisheye image
Boundary_Coordinate_Y	0~ImgHeight	!NULL	Y-axis Coordinate in original fisheye image
Boundary_Count	4~ (ImgWidthOut*2+ ImgHeightOut*2-4)	4	De-warping Boundary Count in original image

3 API Reference

FishEye_OpenInterface

FishEye_CloseInterface

Description

FishEye_OpenInterface and FishEye_CloseInterface are used for open and close SDK's interface. User call HANDLE h = FishEye_Openterface(); to get the analysis object handle. Then user can use the handle to deal with the Fisheye analysis.
When the user wants to end the process, just call FishEye_CloseInterface(h); to delete the object;

Syntax

HANDLE FishEye_OpenInterface();
Bool FishEye_CloseInterface(HANDLE h);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()

Returns

Valid handle returned if success otherwise NULL.

FishEye_Initial

Description

Initial the parameters of the fisheye dependent on lens module.

Syntax

bool FishEye_Initial(HANDLE h, int camera, bool AutoCenter);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()
camera	int	Camera module by #define
AutoCenter	bool	To Choose AutoCenter or Default Center Coordinate

Returns

If the function succeeds, then initial the parameters.

If the function fails, fail to initial parameters.

FishEye_Transform

Description

Transform a fisheye to a chosen transformable mode.

Syntax

bool FishEye_Transform(HANDLE h, EN_FISHEYE_MODE Mode, ST_FISHEYE_IMGINFO ImgInfo, ST_FISHEYE_PARAMETER Par, ST_FISHEYE_OUT Fisheye_out);

Parameters

Name	***Type***	***Description***
h	HANDLE	The handle returned by FishEye_OpenInterface()
Mode	EN_FISHEYE_MODE	Transformable Mode
ImgInfo	ST_FISHEYE_IMGINFO	All Image information include input and output
Par	ST_FISHEYE_PARAMETER	Fisheye transformable mode parameter
Fisheye_out	ST_FISHEYE_OUT	Output Result

Returns

If the function succeeds, then the image has been transformed.

If the function fails, fail to transform the image.

FishEye_Release

Description

Release internal memory

Syntax

bool FishEye_Release(HANDLE h);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()

Returns

If the function succeeds, then the memory has been free

If the function fails, fail to free the memory.

FishEye_GetCoordinate

Description

Transforming Panorama coordinate to De-warping coordinate.

Transforming original fisheye coordinate to De-warping coordinate.

Syntax

bool FishEye_CoordinateTrans(HANDLE h, EN_FISHEYE_MODE Mode, ST_FISHEYE_IMGINFO ImgInfo, ST_FISHEYE_COORD& Coord, ST_FISHEYE_PARAMETER Par, ST_FISHEYE_OUT Fisheye_out);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()
Mode	EN_FISHEYE_COORD_MODE	Transformable Mode
ImgWidth	Int	Input source image width
ImgHeight	Int	Input source image height
Coord	ST_FISHEYE_COORD&	The transformative coordinate.
Par	ST_FISHEYE_PARAMETER	Fisheye transformable mode parameter

Returns

If the function succeeds, then get the transformation coordinate.

If the function fails, fail to get the transformation coordinate.

FishEye_GetCircle

Description

Get current center coordinate and radius.

Syntax

bool FishEye_GetCircle(HANDLE h, int& CenterXout, int& CenterYout, int& RadiusOut, int&ImgWidthOut, int& ImgHeightOut);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()
CenterXout	Int	Output x coordinate of current center
CenterXout	Int	Output y coordinate of current center
RadiusOut	Int	Output current radius
ImgWidthOut	Int	Output reference image width
ImgHeightOut	Int	Output reference image height

Returns

If the function succeeds, then get the center coordinate and radius with the reference image resolution.

If the function fails, fail to get the center coordinate and radius.

FishEye_UserSetCircle

Description

Manually set center coordinate and radius.

Syntax

bool FishEye_UserSetCircle(HANDLE h, int UserCenterX, int UserCenterY, int UserRadius, int ImgWidth, int ImgHeight);

Parameters

Name	*Type*	*Description*
h	HANDLE	The handle returned by FishEye_OpenInterface()
UserCenterX	Int	Input x coordinate of center
UserCenterY	Int	Input y coordinate of center
UserRadius	Int	Input radius
ImgWidth	Int	Width of source image
ImgHeight	Int	Height of source image

Returns

If the function succeeds, then center coordinate and radius is set as input.

If the function fails, fail to set the center coordinate and radius.

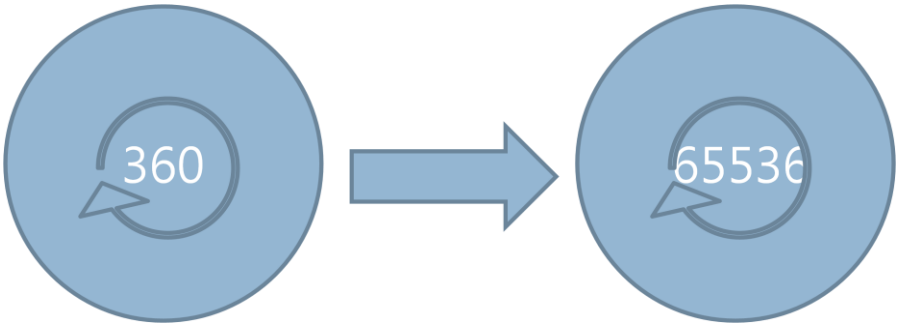
Special Case

If **UserCenterX** and **UserCenterY** are both set = -1, disable using user defined center.

4 Fisheye Coordinate

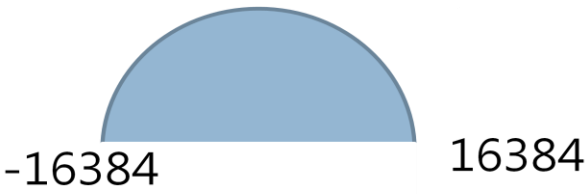
Angle explanation

Traditional angle explanation is [0, 360].In this SDK, we replace [0, 360] with [0, 65536].



Wall Mode

Pan(Horizontal):



Tilt(Vertical):

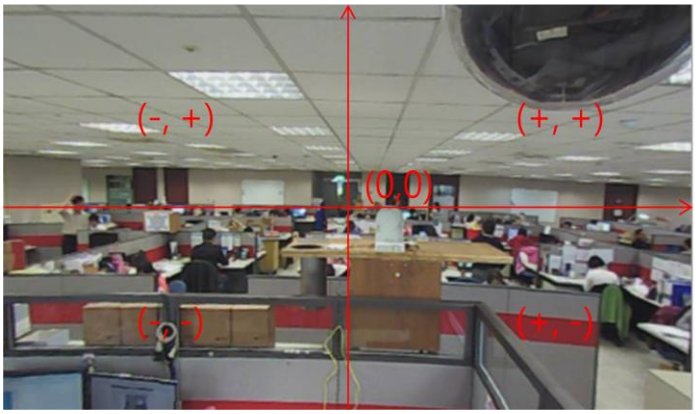
Wall De-Warping

De-warp mode is absolute coordinate, Center Position default is (0,0).

For example:

Set Coordinate Parameter (Pan, Tilt, Zoom) = (0, 0, 256)

Fig1.

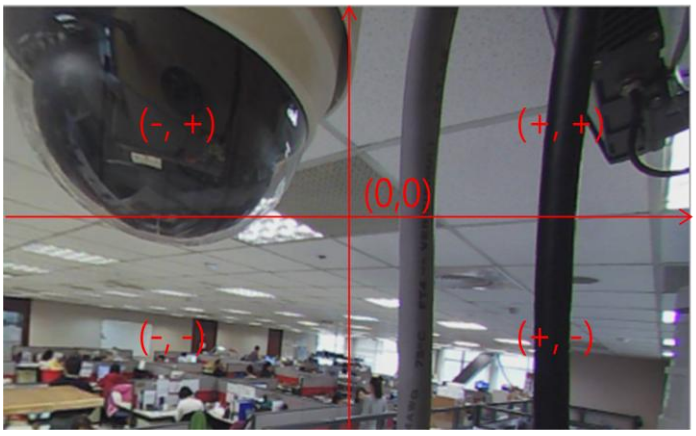


Set Coordinate Parameter (Pan, Tilt, Zoom) = (6000, 3000, 256) from Fig1.

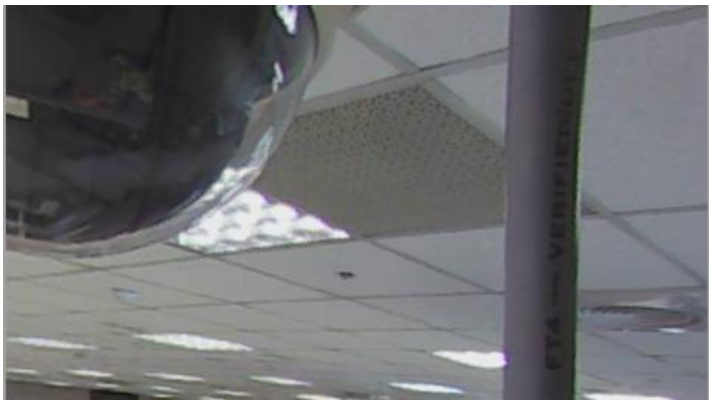
Fig2.

Set Coordinate Parameter (Pan, Tilt, Zoom) = (12000, 6000, 256) from Fig2.

Fig3.



Set Coordinate Parameter (Pan, Tilt, Zoom) = (12000, 6000, 128) from Fig3.



Wall Panorama

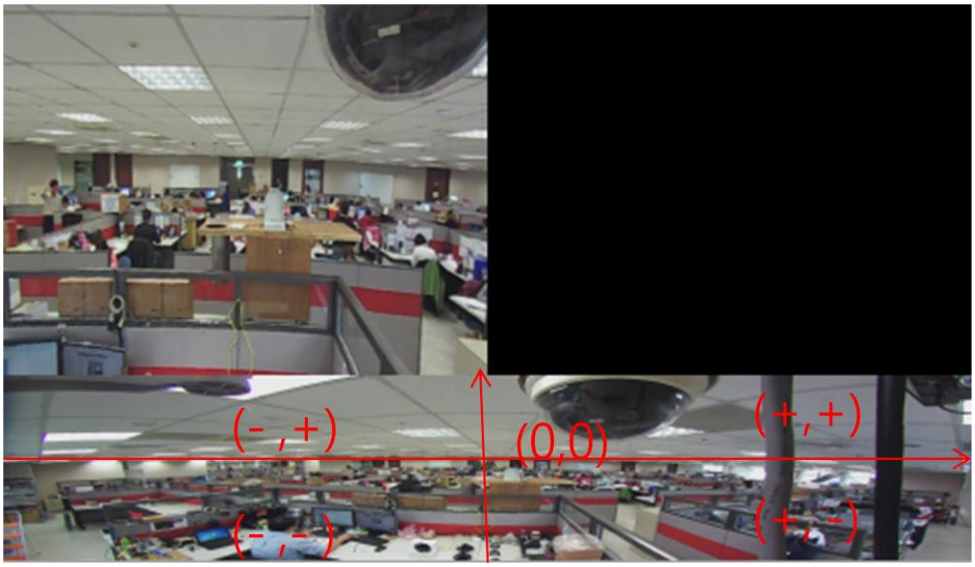
Parameter ViewX = 1.0, Parameter ViewY = 1.0

Parameter ViewX = 1.0, Parameter ViewY = 0.5

Wall De-Warping and Panorama

Panorama coordinate is the width and height of output image
x-axis is (-width/2, width/2)
y-axis is (-height/2, height/2)

for example of below image, it's a 1920x1080 image.
Panorama width = 1920, height = 360

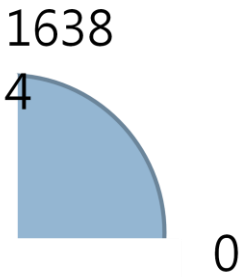


If we choose the coordinate from panorama, then we can transform coordinate from panorama to De-warping by using function" FishEye_GetCoordinate ".And the view of De-warping approach to chosen coordinate in panorama.

Ceiling/Ground Mode

Pan(Horizontal):

Tilt(Vertical):



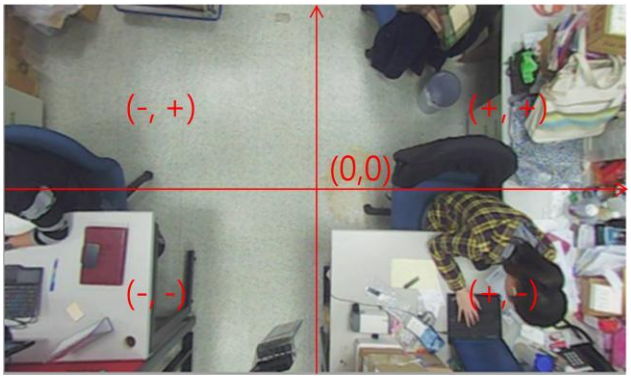
Ceiling/Ground De-warping

De-warp mode is absolute coordinate, Center Position default is (0,0).

For example:

Set Coordinate Parameter (Pan, Tilt, Zoom) = (0, 0, 256)

Fig1.

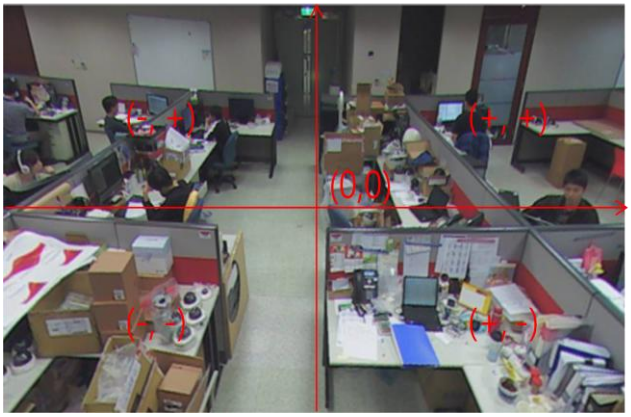


Set Coordinate Parameter (Pan, Tilt, Zoom) = (0, 6000, 256) from Fig1.

Fig2.

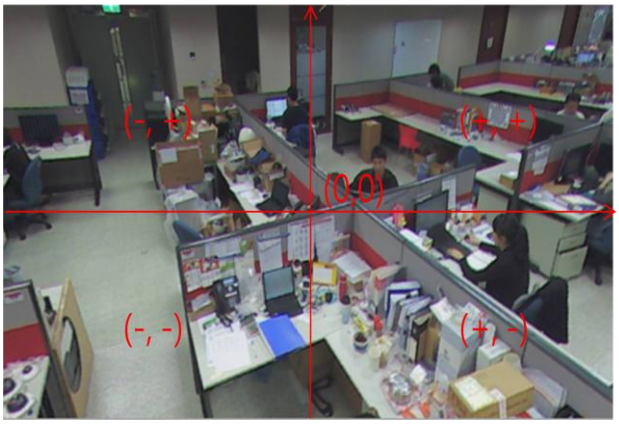
Set Coordinate Parameter (Pan, Tilt, Zoom) = (0, 12000, 256) from Fig2.

Fig3.



Set Coordinate Parameter (Pan, Tilt, Zoom) = (5000, 12000, 256) from Fig3.

Fig4.



Set Coordinate Parameter (Pan, Tilt, Zoom) = (5000, 12000, 128) from Fig4

Ceiling Panorama

Ground Panorama

Ceiling/Ground Double Panorama

Double Panorama mode is absolute coordinate, Center Position default is (0)
The x-axis range is (-65536, 65536)
For example:
Set Coordinate Parameter (Move_XP1, Move_XP2) = (0,32768)
Fig1.

Set Coordinate Parameter (Move_XP1, Move_XP2) = (6000,6000+32768) from Fig1.

Fig2.

Set Extend Parameter (ExtendCenterView1, ExtendCenterView2) = (90, 90) from Fig1.

Fig3.

Ceiling/Ground De-warping and Panorama

Panorama coordinate is the width and height of output image
x-axis is (-width/2, width/2)
y-axis is (-height/2, height/2)
for example of below image, it's a 1920x1080 image.
Panorama width = 1920, height = 360

If we choose the coordinate from panorama, then we can transform coordinate from panorama to De-warping by using function" FishEye_GetCoordinate ".And the view of De-warping approach to chosen coordinate in panorama.

Get original Fisheye Coordinate

The relationship between double panorama with original fisheye image

The relationship between Ceiling De-warping with original fisheye image

FishEye Coordinate to De-warping

Fisheye Coordinate

If we choose the coordinate from fisheye, then we can transform coordinate from fisheye to De-warping by using function" FishEye_GetCoordinate ".And the view of De-warping approach to chosen coordinate in fisheye.

5 Sample Code

Single Wall De-Warping

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par = {NULL};
ST_FISHEYE_OUT FisheyeOut = {NULL};
Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);

……
Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;

…….

HANDLE FISHEYE = FishEye_OpenInterface();
FishEye_Initial(FISHEYE, MODEL_NO_SPECIFIC, AutoCenter);


Par.Wall_D.Pan = 10000;
Par.Wall_D.Tilt = 5000;
```

```
Par.Wall_D.Zoom = 256;
Par.Rectangle.RectStartX = 0;
Par.Rectangle.RectStartY = 0;
Par.Rectangle.RectWidth = Img.ImgWidthOut;
Par.Rectangle.RectHeight = Img.ImgHeightOut ;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par, FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Multi-Wall De-Warping

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

……

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par[4] = {NULL};
ST_FISHEYE_OUT Fisheye_out = {NULL};

Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);

……

Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;

//first De-warping
Par[0].Wall_D.Pan = 10000;
Par[0].Wall_D.Tilt = 5000;
Par[0].Wall_D.Zoom = 256;
Par[0].Rectangle.RectStartX = 0;
Par[0].Rectangle.RectStartY = 0;
Par[0].Rectangle.RectWidth = Img.ImgWidthOut /2;
Par[0].Rectangle.RectHeight = Img.ImgHeightOut/2;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[0], FisheyeOut);


//second De-warping
```

```
Par[1].Wall_D.Pan = 10000;
Par[1].Wall_D.Tilt = 5000;
Par[1].Wall_D.Zoom = 256;
Par[1].Rectangle.RectStartX = Img.ImgWidthOut /2;
Par[1].Rectangle.RectStartY = 0;
Par[1].Rectangle.RectWidth = Img.ImgWidthOut /2;
Par[1].Rectangle.RectHeight = Img.ImgHeightOut/2;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[1], FisheyeOut);

//third De-warping
Par[2].Wall_D.Pan = 10000;
Par[2].Wall_D.Tilt = 5000;
Par[2].Wall_D.Zoom = 256;
Par[2].Rectangle.RectStartX = 0;
Par[2].Rectangle.RectStartY = Img.ImgHeightOut/2;
Par[2].Rectangle.RectWidth = Img.ImgWidthOut /2;
Par[2].Rectangle.RectHeight = Img.ImgHeightOut/2;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[2], FisheyeOut);

//fourth De-warping
Par[3].Wall_D.Pan = 10000;
Par[3].Wall_D.Tilt = 5000;
Par[3].Wall_D.Zoom = 256;
Par[3].Rectangle.RectStartX = Img.ImgWidthOut /2;
Par[3].Rectangle.RectStartY = Img.ImgHeightOut/2;
Par[3].Rectangle.RectWidth = Img.ImgWidthOut /2;
Par[3].Rectangle.RectHeight = Img.ImgHeightOut/2;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[3], FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Single Wall De-warping and Panorama

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par[2] = {NULL};
ST_FISHEYE_COORD Coord ={NULL}
ST_FISHEYE_OUT Fisheye_out = {NULL};

Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);
……

Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;


Par[0].Wall_P.ViewX = 1;
Par[0].Wall_P.ViewY = 1;
Par[0].Wall_P.Tilt = 1024;
Par[0].Rectangle.RectStartX = 0;
Par[0].Rectangle.RectStartY = Img.ImgHeightOut/2;
Par[0].Rectangle.RectWidth = Img.ImgWidthOut;
Par[0].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[0].Rectangle.RectNumber = 0;
FishEye_Transform(FISHEYE, enWall_Panorama, Img, Par[0], FisheyeOut);

Coord. InputRectangleCoordX = -497;
```

```
Coord. InputRectangleCoordY= 110;
FishEye_GetCoordinate(FISHEYE, enWall_Panorama2PTZ, ImgWidth , ImgHeight,
Coord, Par[0],);

Par[1].Wall_D.Pan = Coord.TransformPan;
Par[1].Wall_D.Tilt = Coord.TransformTilt;
Par[1].Wall_D.Zoom = 256;
Par[1].Rectangle.RectStartX = 0;
Par[1].Rectangle.RectStartY = 0;
Par[1].Rectangle.RectWidth = Img.ImgWidthOut/2;
Par[1].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[1].Rectangle.RectNumber = 1;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[1], FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Ceiling Double Panorama (Same for Ground)

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par[2] = {NULL};
Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);
ST_FISHEYE_OUT Fisheye_out = {NULL};

……

Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;

Par[0].Ceiling_DP.View = 1;
Par[0].Ceiling_DP.Move_XP = 0;
Par[0].Ceiling_DP.ExtendCenterView = 0;
Par[0].Rectangle.RectStartX = 0;
Par[0].Rectangle.RectStartY = 0;
Par[0].Rectangle.RectWidth = Img.ImgWidthOut;
Par[0].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[0].Rectangle.RectNumber = 0;
FishEye_Transform(FISHEYE, enCeiling_DoublePanorama, Img, Par[0], FisheyeOut);

Par[1].Ceiling_DP.View = 1;
Par[1].Ceiling_DP.Move_XP = 32768;
Par[1].Ceiling_DP.ExtendCenterView = 0;
```

```
Par[1].Rectangle.RectStartX = 0;
Par[1].Rectangle.RectStartY = Img.ImgHeightOut/2;
Par[1].Rectangle.RectWidth = Img.ImgWidthOut;
Par[1].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[1].Rectangle.RectNumber = 1;
FishEye_Transform(FISHEYE, enCeiling_DoublePanorama, Img, Par[1], FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Get Original Fisheye Coordinate from De-warping

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par = {NULL};
ST_FISHEYE_OUT FisheyeOut = {NULL};
Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);

……
Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;

int Array_X[100];
int Array_Y[100];
FisheyeOut.Boundary_Count = 100;
FisheyeOut.Boundary_Coordinate_X = Array_X;
FisheyeOut.Boundary_Coordinate_Y = Array_Y;
…….

HANDLE FISHEYE = FishEye_OpenInterface();
FishEye_Initial(FISHEYE, MODEL_NO_SPECIFIC, AutoCenter);


Par.Wall_D.Pan = 10000;
Par.Wall_D.Tilt = 5000;
Par.Wall_D.Zoom = 256;
```

```
Par.Rectangle.RectStartX = 0;
Par.Rectangle.RectStartY = 0;
Par.Rectangle.RectWidth = Img.ImgWidthOut;
Par.Rectangle.RectHeight = Img.ImgHeightOut ;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par, FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Get De-warping Coordinate from Original Fisheye

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par[2] = {NULL};
ST_FISHEYE_COORD Coord ={NULL}
ST_FISHEYE_OUT Fisheye_out = {NULL};

Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);
……

Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];
Img.YUV = false;


Par[0].Wall_P.ViewX = 1;
Par[0].Wall_P.ViewY = 1;
Par[0].Rectangle.RectStartX = 0;
Par[0].Rectangle.RectStartY = Img.ImgHeightOut/2;
Par[0].Rectangle.RectWidth = Img.ImgWidthOut;
Par[0].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[0].Rectangle.RectNumber = 0;
FishEye_Transform(FISHEYE, enWall_Panorama, Img, Par[0], FisheyeOut);

Coord.InputRectangleCoordX = 220;
Coord.InputRectangleCoordY= 100;
```

```
FishEye_GetCoordinate(FISHEYE, enWall_Fisheye2PTZ, ImgWidth, ImgHeight,
Coord, Par[0]);

Par[1].Wall_D.Pan = Coord.TransformPan;
Par[1].Wall_D.Tilt = Coord.TransformTilt;
Par[1].Wall_D.Zoom = 256;
Par[1].Rectangle.RectStartX = 0;
Par[1].Rectangle.RectStartY = 0;
Par[1].Rectangle.RectWidth = Img.ImgWidthOut/2;
Par[1].Rectangle.RectHeight = Img.ImgHeightOut/2;
Par[1].Rectangle.RectNumber = 1;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par[1], FisheyeOut);

FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

Get and Set Circle

```
#include "FishEyeSDK.h"
#include "FishEyeBase.h"

…….

ST_FISHEYE_IMGINFO Img = {NULL};
ST_FISHEYE_PARAMETER Par = {NULL};
ST_FISHEYE_OUT FisheyeOut = {NULL};
Int ImgWidth = GetFromSrc(Width);
Int ImgHeight = GetFromSrc(Height);
LPBYTE lpSrc = GetFromSrc(Buffer);

…….
Bool AutoCenter = true;
Img.lpSrc =lpSrc;
Img.ImgWidth =ImgWidth;
Img.ImgHeight =ImgHeight;
Img.ImgWidthOut = 1920;
Img.ImgHeightOut = 1080;
Img.lpDst = new BYTE[Img.ImgWidthOut * Img.ImgHeightOut *BitPerPixel];

…….

HANDLE FISHEYE = FishEye_OpenInterface();
FishEye_Initial(FISHEYE, MODEL_NO_SPECIFIC, AutoCenter);


Par.Wall_D.Pan = 10000;
Par.Wall_D.Tilt = 5000;
Par.Wall_D.Zoom = 256;
Par.Rectangle.RectStartX = 0;
Par.Rectangle.RectStartY = 0;
Par.Rectangle.RectWidth = Img.ImgWidthOut;
Par.Rectangle.RectHeight = Img.ImgHeightOut ;
FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par, FisheyeOut);

//Get Current Circle Information
int CenterXout, CenterYout, RadiusOut, ImgWidthOut, ImgHeightOut;
```

```
FishEye_GetCircle(FISHEYE, CenterXout, CenterYout, RadiusOut, ImgWidthOut,
ImgHeightOut);

//Set User Defined Circle
int UserCenterX = 600;
int UserCenterY = 400;
int UserRadius    = 500;
FishEye_UserSetCircle(FISHEYE, UserCenterX, UserCenterY, UserRadius, ImgWidth,
ImgHeight);

FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par, FisheyeOut);

//Disable Using User Defined Circle
UserCenterX = -1;
UserCenterY = -1;
FishEye_UserSetCircle(FISHEYE, UserCenterX, UserCenterY, UserRadius, ImgWidth,
ImgHeight);

FishEye_Transform(FISHEYE, enWall_Dewarping, Img, Par, FisheyeOut);


FishEye_Release(FISHEYE);
FishEye_CloseInterface(FISHEYE);

Delete []FisheyeOut.lpDst;
```

6 Linux Makefile Sample

```
LIB := ../Lib/ #This is the directory I put “FishEyeSDK.h” and “FishEyeBase.h”
EXE := FishEye_API

Dynamic: FishEye_API.cpp
	gcc -O2 $< -o $(EXE) -I$(LIB) ../FisheyeSDK/libFisheyeSDK.so -lm -lstdc++
Static: FishEye_API.cpp
	gcc -O2 $< -o $(EXE) -I$(LIB) ../FisheyeSDK/libFisheyeSDK.a    -lm -lstdc++

clean:
	rm -rf FishEye_API
```