

# ImmerVision Enables 2.0

## ANNEX RectangleSurface

### Programmer's Guide

Document version: 1.2

Document release date: 2019-11-20

Information, including illustrations, data and drawings, contained in this document is strictly confidential and proprietary to ImmerVision and benefits from extended and combined intellectual property rights protection including copyrights, industrial designs, patents, know how, trade secrets and trademarks. It may not be used, modified, copied, published, disclosed, distributed, displayed or exhibited in either electronic or printed formats without the express written authorization of an officer of ImmerVision. Nothing in this document shall be construed as granting any license, express or implied, by any means whatever, to any intellectual property rights in products or technology of ImmerVision or in any information contained in this document. This document is provided "as is" with no warranty of merchantability or fitness and is subject to change without notice. ImmerVision, the ImmerVision logo and other ImmerVision marks are owned by ImmerVision. All other trademarks are the property of their respective owners.

# Contents

---

CONTENTS .....	2
<b>DESCRIPTION .....</b>	<b>3</b>
INTRODUCTION .....	3
SPECIAL CASE1: NADIR PATCHING.....	4
<i>TestCase</i> .....	4
SPECIAL CASE2: BACK PATCHING .....	6
ADDERECTANGLESURFACE.....	7
GETRECTANGLESURFACEPOSITION .....	8
GETRECTANGLESURFACEPOSITION .....	9
REMOVERECTANGLESURFACE .....	10
SETRECTANGLESURFACEINPUTBUFFER .....	11
SETRECTANGLESURFACEPOSITION .....	12
SETRECTANGLESURFACEPOSITION .....	13
SHOWRECTANGLESURFACE .....	14
ANNEX 1 .....	15

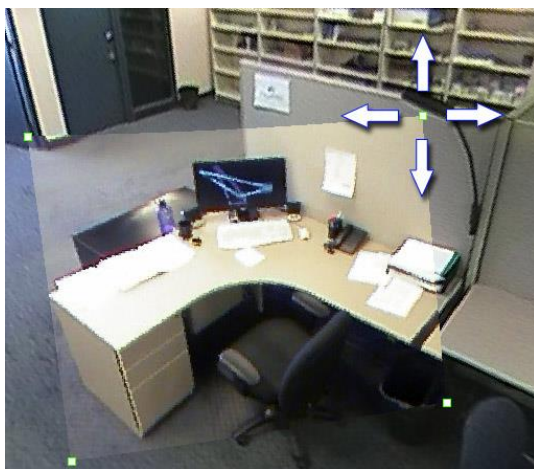
# DESCRIPTION

---

## Introduction

---

Rectangle surface represents an image or video surface you can map into the panoramic image. The points defining the corner of the rectangle can be placed precisely in your environment.



This could be used if you incrust high-res camera images into a panoramic environment.

Sample7 and Sample Surface Calibration demonstrate this functionality.

## Special case1: Nadir Patching

360° panomorph videos can show the hand or the device itself. It's mainly visible at Nadir position.

In this special case, we will explain how to patch the Nadir with any logo or graphic file.

### TestCase

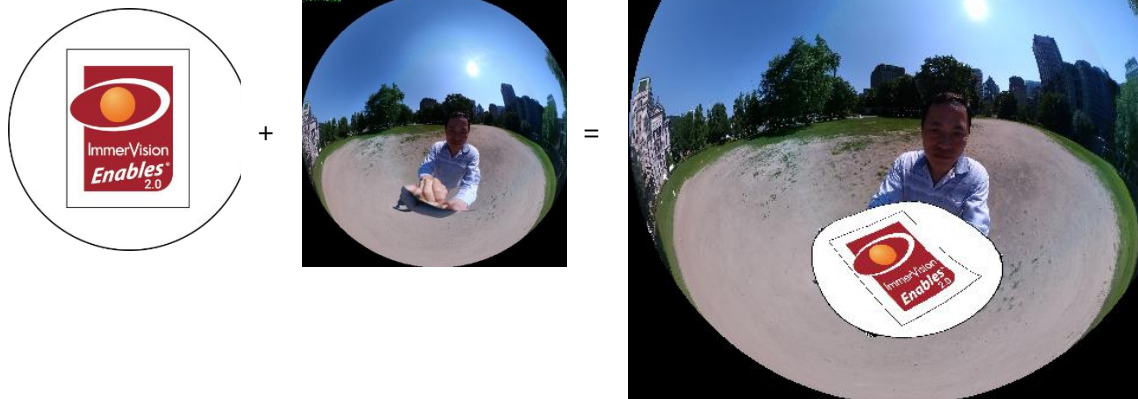
Here is a 360° video shot that illustrate the Nadir patching:



The raw 360° dewarping result will be:



The patch consists in drawing an image on top on the panorama at Nadir Position, where the device is located such as:



The way to add the image is to set the position of the 4 corners of the logo at the right position.

Considering tilt =  $-90^\circ$  at Nadir position and tilt =  $90^\circ$  at Zenith position, then we could position all our image corners at **tilt =  $-50^\circ$** .

To position pan, being included between 0 and  $360^\circ$ , we just have to set pan every  $90^\circ$  to cover the whole panorama. So **pan=0, 90, 180, 270**.

Our code will simply be:

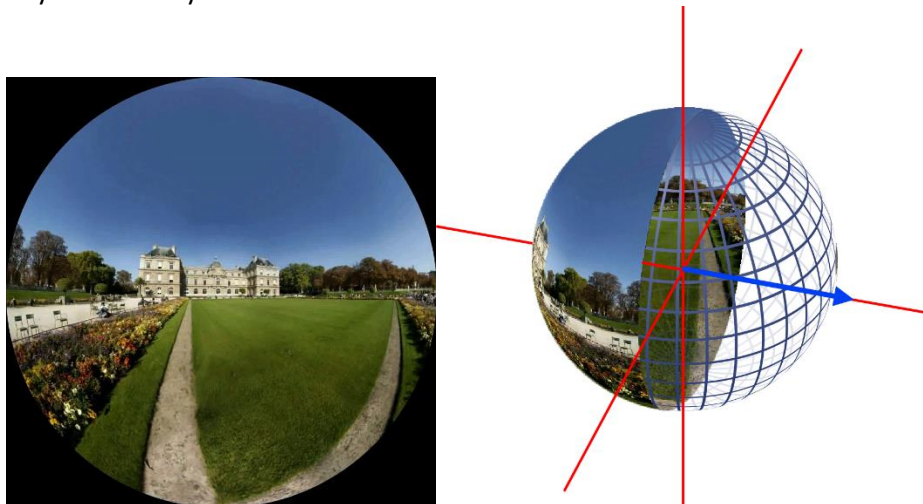
```
int index = 1;
IVE_Camera->AddRectangleSurface(&inputBufferLOGO, &index);
IVE_Camera->SetRectangleSurfacePosition(index, 0, -50, 90, -50, 180, -50, 270, -50);
```

## Special case2: Back Patching

---

180° image represents half of the whole 360° environment.

Using stabilized navigation type, you will be able to navigate in the 360° environment without any constraints. Looking at the opposite orientation than the source image, you may not see any content anymore.



Looking backward will not display any image.

In this case you may want to display an image (telling to look the other side) or your logo.

## AddRectangleSurface

---

```
unsigned long AddRectangleSurface(  
    IMV_Buffer /*in*/ *inputBuffer,  
    int /*out*/ *RectangleSurfaceIndex);
```

### Description

This function adds a rectangle surface in the panoramic scene. This surface will be mapped with the video data contained in the inputBuffer parameter.

This function generates an integer value that uniquely identifies this surface.

### Parameters

inputBuffer	Input buffer used to fill the surface.
rectangleSurfaceIndex	If the function succeeds, it returns the index of the surface.

### Return codes

IMV\_Defs::E\_ERR\_NOTINITIALIZED

The library is not currently initialized.

IMV\_Defs::E\_ERR\_NOTALLOWED

The library cannot add more rectangleSurface. The number of maximum surface has been reached.

IMV\_Defs::E\_ERR\_OUTOFMEMORY

Fatal Error: the computer does not have enough memory to continue running the library and/or other programs. Close the library immediately.

IMV\_Defs::E\_ERR\_OK

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) and video-out buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the IMV\_Buffer structure requirement.

- The library allows you to add 256 Surfaces of this type. An error will be returned if you reach this number.

- The color format (RGB, YUV...) of the video data contained in the buffer is considered to be the same format than the main input buffer.

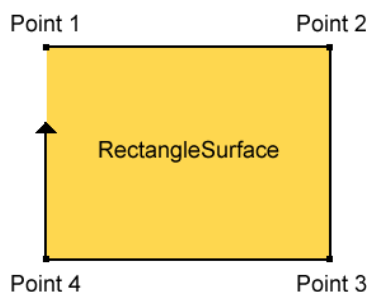
## GetRectangleSurfacePosition

```
unsigned long GetRectangleSurfacePosition(  
    int rectangleSurfaceIndex,  
    float /*out*/ *x1, float /*out*/ *y1, float /*out*/ *z1,  
    float /*out*/ *x2, float /*out*/ *y2, float /*out*/ *z2,  
    float /*out*/ *x3, float /*out*/ *y3, float /*out*/ *z3,  
    float /*out*/ *x4, float /*out*/ *y4, float /*out*/ *z4);
```

### Description

This function gets the position of a rectangleSurface.

The position is characterized by four 3D points. Each point represents a corner of the rectangleSurface.



### Parameters

rectangleSurfaceIndex	Index of the surface to move.
x1, y1, z1	Coordinates of the point 1 in the camera coordinate system*
x2, y2, z2	Coordinates of the point 2 in the camera coordinate system*
x3, y3, z3	Coordinates of the point 3 in the camera coordinate system*
x4, y4, z4	Coordinates of the point 4 in the camera coordinate system*

\* see 'annex 1' to have a representation of the camera coordinate system

### Return codes

IMV\_Defs::E\_ERR\_NOTINITIALIZED

The library is not currently initialized.

IMV\_Defs::E\_ERR\_INDEXINVALID

The input index is invalid. The RectangleSurfaceIndex is not assigned to any rectangleSurface.

IMV\_Defs::E\_ERR\_OK

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the IMV\_Buffer structure requirement.



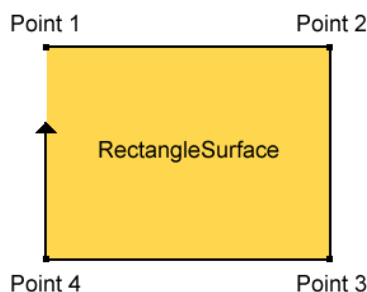
## GetRectangleSurfacePosition

```
unsigned long GetRectangleSurfacePosition(
    int rectangleSurfaceIndex,
    float /*out*/ *pan1, float /*out*/ *tilt1,
    float /*out*/ *pan2, float /*out*/ *tilt2,
    float /*out*/ *pan3, float /*out*/ *tilt3,
    float /*out*/ *pan4, float /*out*/ *tilt4);
```

### Description

This function sets the position of a rectangleSurface.

The position is characterized by 4 point angles. Each point represents a corner of the rectangleSurface.



### Parameters

rectangleSurfaceIndex	Index of the surface to move.
pan1, tilt1	Coordinates of the point 1 in the camera coordinate system*
pan2, tilt2	Coordinates of the point 2 in the camera coordinate system*
pan3, tilt3	Coordinates of the point 3 in the camera coordinate system*
pan4, tilt4	Coordinates of the point 4 in the camera coordinate system*

\* see 'annex 1' to have a representation of the camera coordinate system

### Return codes

IMV\_Defs::E\_ERR\_NOTINITIALIZED

The library is not currently initialized.

IMV\_Defs::E\_ERR\_INDEXINVALID

The input index is invalid. The RectangleSurfaceIndex is not assigned to any rectangleSurface.

IMV\_Defs::E\_ERR\_OK

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the IMV\_Buffer structure requirement.

## RemoveRectangleSurface

---

```
unsigned long RemoveRectangleSurface(int RectangleSurfaceIndex);
```

### Description

This function removes a rectangleSurface from the panoramic environment.

### Parameters

RectangleSurfaceIndex	Index of the surface to remove
-----------------------	--------------------------------

### Return codes

IMV\_Defs::E\_ERR\_NOTINITIALIZED

The library is not currently initialized.

IMV\_Defs::E\_ERR\_INDEXINVALID

The input index is invalid. The RectangleSurfaceIndex is not assigned to any rectangleSurface.

IMV\_Defs::E\_ERR\_OK

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the IMV\_Buffer structure requirement.

## SetRectangleSurfaceInputBuffer

---

```
unsigned long SetRectangleSurfaceInputBuffer(  
    int rectangleSurfaceIndex,  
    IMV_Buffer /*in*/ *inputBuffer);
```

### Description

This function is to replace the current input buffer by the one passed in parameter to the function. The change will take effect during the next call up of the Update() function.

### Parameters

<code>rectangleSurfaceIndex</code>	Index of the surface to replace.
<code>inputBuffer</code>	New input buffer used by the library.

### Return codes

`IMV_Defs::E_ERR_NOTINITIALIZED`

The library is not currently initialized.

`IMV_Defs::E_ERR_INDEXINVALID`

The input index is invalid. The `RectangleSurfaceIndex` is not assigned to any `rectangleSurface`.

`IMV_Defs::E_ERR_OUTOFMEMORY`

Fatal Error: the computer does not have enough memory to continue running the library and/or other programs. Close the library immediately.

`IMV_Defs::E_ERR_OK`

The function succeeded.

### Remarks

- Video-in (panomorph video and `rectangleSurface` video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the `IMV_Buffer` structure requirement.

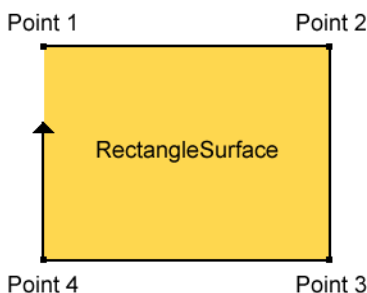
## SetRectangleSurfacePosition

```
unsigned long SetRectangleSurfacePosition(  
    int rectangleSurfaceIndex,  
    float x1, float y1, float z1,  
    float x2, float y2, float z2,  
    float x3, float y3, float z3,  
    float x4, float y4, float z4);
```

### Description

This function sets the position of a rectangleSurface. This position is characterized by four 3D points. Each point represents a corner of the rectangleSurface.

To correctly visualize the surface you need to set the points in a clockwise order.



### Parameters

<code>rectangleSurfaceIndex</code>	Index of the surface to move.
<code>x1, y1, z1</code>	Coordinates of the point 1 in the camera coordinate system*
<code>x2, y2, z2</code>	Coordinates of the point 2 in the camera coordinate system*
<code>x3, y3, z3</code>	Coordinates of the point 3 in the camera coordinate system*
<code>x4, y4, z4</code>	Coordinates of the point 4 in the camera coordinate system*

\* see 'annex 1' to have a representation of the camera coordinate system

### Return codes

`IMV_Defs::E_ERR_NOTINITIALIZED`

The library is not currently initialized.

`IMV_Defs::E_ERR_INDEXINVALID`

The input index is invalid. The `RectangleSurfaceIndex` is not assigned to any `rectangleSurface`.

`IMV_Defs::E_ERR_PARAMINVALID`

The input parameters are invalid. The point coordinates are not set in a clockwise order. The surface may not be displayed.

`IMV_Defs::E_ERR_OK`

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the `IMV_Buffer` structure requirement.

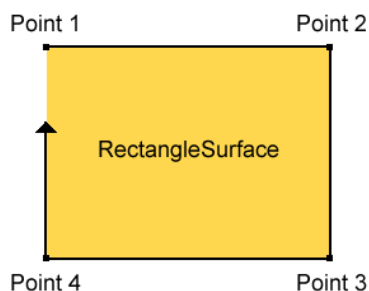
## SetRectangleSurfacePosition

```
unsigned long SetRectangleSurfacePosition(
    int rectangleSurfaceIndex,
    float pan1, float tilt1,
    float pan2, float tilt2,
    float pan3, float tilt3,
    float pan4, float tilt4);
```

### Description

This function sets the position of a rectangleSurface. This position is characterized by 4 point angles. Each point represents a corner of the rectangleSurface.

In order to correctly visualize the surface you need to set points in a clockwise order.



### Parameters

<a href="#">rectangleSurfaceIndex</a>	Index of the surface to move.
<a href="#">pan1, tilt1</a>	Coordinates of the point 1 in the camera coordinate system*
<a href="#">pan2, tilt2</a>	Coordinates of the point 2 in the camera coordinate system*
<a href="#">pan3, tilt3</a>	Coordinates of the point 3 in the camera coordinate system*
<a href="#">pan4, tilt4</a>	Coordinates of the point 4 in the camera coordinate system*

\* see 'annex 1' to have a representation of the camera coordinate system

### Return codes

[IMV\\_Defs::E\\_ERR\\_NOTINITIALIZED](#)

The library is not currently initialized.

[IMV\\_Defs::E\\_ERR\\_INDEXINVALID](#)

The input index is invalid. The RectangleSurfaceIndex is not assigned to any rectangleSurface.

[IMV\\_Defs::E\\_ERR\\_PARAMINVALID](#)

The input parameters are invalid. The point coordinates are not set in a clockwise order. The surface may not be displayed.

[IMV\\_Defs::E\\_ERR\\_OK](#)

The function succeeded.

### Remarks

- Video-in (panomorph video and rectangleSurface video data) buffers must be initialized (!= Null) and allocated (size = width\*height\*Color\_depth). Their data must respect the IMV\_Buffer structure requirement.

## ShowRectangleSurface

---

```
unsigned long ShowRectangleSurface(    int rectangleSurfaceIndex,  
                                     bool visible);
```

### Description

This function sets the rectangleSurface visibility.

### Parameters

<code>rectangleSurfaceIndex</code>	Index of the surface to modify.
<code>visible</code>	Surface visibility. The default value is true. Set it to false to avoid the surface display.

### Return codes

`IMV_Defs::E_ERR_NOTINITIALIZED`

The library is not currently initialized.

`IMV_Defs::E_ERR_INDEXINVALID`

The input index is invalid. The RectangleSurfaceIndex is not assigned to any rectangleSurface.

`IMV_Defs::E_ERR_OK`

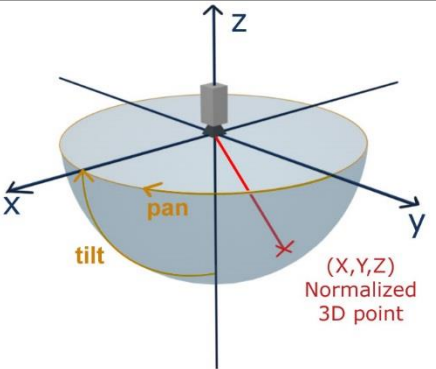
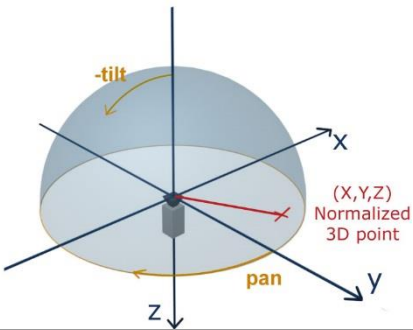
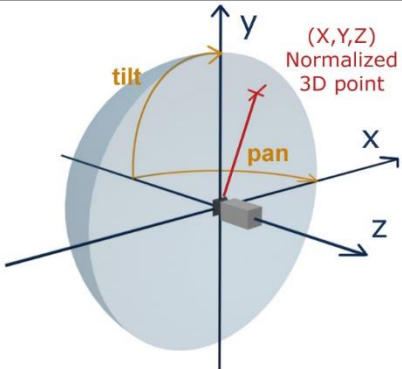
The function succeeded.

### Remarks

-

## Annex 1

### The camera coordinates system

	Ceiling mode
	Ground mode
	Wall mode

### Remarks

(pan,tilt) orientations are adapted to the 360° navigation referential.  
(x,y,z) coordinate system stays the same.