AVT Image Transform

Programmer's Manual

V1.2
2013-Jun-25

Allied Vision Technologies GmbH
Taschenweg 2a
D-07646 Stadtroda / Germany

ALLIED Vision Technologies

Legal Notice

Trademarks

Unless stated otherwise, all trademarks appearing in this document of Allied Vision Technologies are brands protected by law.

Warranty

The information provided by Allied Vision Technologies is supplied without any guarantees or warranty whatsoever, be it specific or implicit. Also excluded are all implicit warranties concerning the negotiability, the suitability for specific applications or the non-breaking of laws and patents. Even if we assume that the information supplied to us is accurate, errors and inaccuracy may still occur.

Copyright

All texts, pictures and graphics are protected by copyright and other laws protecting intellectual property. It is not permitted to copy or modify them for trade use or transfer, nor may they be used on websites.

Allied Vision Technologies GmbH 06/2013

Managing Director: Mr. Frank Grube
Tax ID: DE 184383113

Headquarters:

Taschenweg 2a
D-07646 Stadtroda, Germany
Tel.: +49 (0)36428 6770
Fax: +49 (0)36428 677-28
e-mail: info@alliedvisiontec.com

Contents

1 Contacting Allied Vision Technologies

Note

- **Technical Information**
  http://www.alliedvisiontec.com
- **Support**
  support@alliedvisiontec.com

Allied Vision Technologies GmbH (Headquarters)
Taschenweg 2a
07646 Stadtroda, Germany
Tel.: +49 36428-677-0
Fax.: +49 36428-677-28
Email: info@alliedvisiontec.com

Allied Vision Technologies Canada Inc.
101-3750 North Fraser Way
Burnaby, BC, V5J 5E9, Canada
Tel: +1 604-875-8855
Fax: +1 604-875-8856
Email: info@alliedvisiontec.com

Allied Vision Technologies Inc.
38 Washington Street
Newburyport, MA 01950, USA
Toll Free number +1 877-USA-1394
Tel.: +1 978-225-2030
Fax: +1 978-225-2029
Email: info@alliedvisiontec.com

Allied Vision Technologies Asia Pte. Ltd.
82 Playfair Road
#07-02 D'Lithium
Singapore 368001
Tel. +65 6634-9027
Fax:+65 6634-9029
Email: info@alliedvisiontec.com

Allied Vision Technologies (Shanghai) Co., Ltd.
2-2109 Hongwell International Plaza
1602# ZhongShanXi Road
Shanghai 200235, China
Tel: +86 (21) 64861133
Fax: +86 (21) 54233670
Email: info@alliedvisiontec.com

2 Introduction

2.1 Document history

Version	Date	Changes
1.0	22. 03. 2013-03-22	Initial version
1.2	2013-Jun-18	Small corrections, layout changes

2.2 Conventions used in this manual

To give this manual an easily understood layout and to emphasize important information, the following typographical styles and symbols are used:

2.2.1 Styles

Style	Function	Example
Bold	Programs, inputs or highlighting important things	**bold**
Courier	Code listings etc.	`Input`
Upper case	Constants	CONSTANT
Italics	Modes, fields	*Mode*
Parentheses and/or blue	Links	(Link)

2.2.2 Symbols

Note

This symbol highlights important information.

Caution

This symbol highlights important instructions. You have to follow these instructions to avoid malfunctions.

WWW

This symbol highlights URLs for further information. The URL itself is shown in blue.
Example: http://www.alliedvisiontec.com

3 General

The AVT Image Transform API enables to transform images received via AVT VIMBA C/C++ API into common display image formats. The API consists of the VmbImageTransform function and a collection of parameter initialization helper functions that help set image and options parameters.

4 Methods

4.1 VmbGetVersion()

GetVersion inquires the library version.

```
VmbError_t VmbGetVersion ( VmbUint32Ptr_t   Value );
```

	Type	Name	Description
out	VmbUint32Ptr_t	Value	Contains the library version (Major,Minor,Sub,Build)

4.2 VmbGetErrorInfo()

GetErrorInfo translates VIMBA error codes into a human-readable string.

```
VmbError_t VmbGetErrorInfo( VmbError_t     ErrorCode,
                            VmbANSIChar_t *InfoString,
                            VmbUint32_t    MaxInfoLength );
```

	Type	Name	Description
in	VmbError_t	ErrorCode	The error code to get a readable string for
out	VmbANSIChar_t*	InfoString	Pointer to a zero terminated string that will contain the error information on return
in	VmbUint32_t	MaxInfoLength	The length of the InfoString buffer

4.3 VmbGetApiInfoString()

Get information about the currently loaded VIMBA ImageTransform API.

```
VmbError_t VmbGetApiInfoString( VmbAPIInfo_t   InfoType,
                                VmbANSIChar_t *Info,
                                VmbUint32_t    MaxInfoLength);
```

	Type	Name	Description
in	VmbAPIInfo_t	InfoType	Type of information to return
out	VmbANSIChar_t*	Info	Pointer to a zero terminated string that will contain the information on return
in	VmbUint32_t	MaxInfoLength	The length of the Info buffer

InfoType:

Value	Description
VmbAPIInfoAll	Returns all information about the API
VmbAPIInfoPlatform	Returns information about the platform the API was built for (x86 or x64)
VmbAPIInfoBuild	Returns info about the API built (debug or release).
VmbApiInfoTechnology	Returns info about the supported technologies the API was built for (OpenMP or OpenCL).

4.4 SetImageInfo()

Set image info member values in VmbImage from pixel format or string.

```
VmbError_t VmbSetImageInfoFromPixelFormat( VmbPixelFormat_t PixelFormat,
                                           VmbUint32_t      Width,
                                           VmbUint32_t      Height,
                                           VmbImage        *Image);

VmbError_t VmbSetImageInfoFromString(const VmbANSIChar_t *ImageFormat,
                                           VmbUint32_t    StringLength,
                                           VmbUint32_t    Width,
                                           VmbUint32_t    Height,
                                           VmbImage      *Image);
```

	Type	Name	Description
in	VmbPixelFormat_t	PixelFormat	PixelFormat describes the pixel format used by the image data member. VmbPixelFormat_t can be obtained from VIMBA C/Cpp APIs frame or from the "PixelFormat" feature. For displaying images, it is suggested to use VmbSetImageInfoFromString or to look up a matching VmbPixelFormat
in	VmbANSIChar_t*	ImageFormat	Image format as a (const) case insensitive string that is either a PixelFormat (Vmb is optional) or a pixel struct name
in	VmbUint32_t	StringLength	The length of the pixel format string
in	VmbUint32_t	Width	Width of the image in pixels
in	VmbUint32_t	Height	Height of the image in pixels
in/out	VmbImage*	Image	Pointer to VIMBA image to set the info to

Example:

```
std::string name("Mono8_REC601");
VmbSetImageInfoFromString( name.c_str(),
                           static_cast<VmbUint32_t>(name.size()) ,
                           64 ,
                           64 ,
                           &image );
```

4.5 VmbImageTransform()

Images are transformed using the VmbImageTransform function. The transformation is defined by the images used and the desired transformation. If a transformation is not supported, VmbErrorBadParameter is returned.

```
VmbError_t VmbImageTransform( const VmbImage          *Source,
                              VmbImage                *Destination,
                              const VmbTransformInfo  *Parameter,
                              VmbUint32_t              ParameterCount);
```

	Type	Name	Description
in	`VmbImage*`	Source	Image to transform
out	`VmbImage*`	Destination	Destination image
in	`VmbTransformInfo*`	Parameter	Optional transform parameters
in	`VmbUint32_t`	ParameterCount	Number of transform parameters

5 Structs

5.1 VmbImage

VmbImage encapsulates image data for the transformation function.

```
typedef struct VmbImage
{
    VmbUint32_t     Size;
    void           *Data;
    VmbImageInfo    ImageInfo;
}VmbImage;
```

Type	Name	Description
VmbUint32_t	Size	Size of the structure
void*	Data	Pointer to the payload received from AVT VIMBA C/Cpp API or to the display image data
VmbImageInfo	ImageInfo	Internal information data used for mapping the images to the correct transformation, ImageInfo data can be set with VmbSetImageInfo helper functions from VmbPixelFormat_t or format string

Example:

```
VmbImage vimba_source;
vimba_source.Size = sizeof(vimba_source);
```

5.2 VmbImageInfo

VmbImageInfo contains image information needed for the transformation function.

```
typedef struct VmbImageInfo
{
    VmbUint32_t     Width;
    VmbUint32_t     Height;
    VmbInt32_t      Stride;
    VmbPixelInfo    PixelInfo;
};
```

Type	Name	Description
VmbUint32_t	Width	The width of the image in pixels. For macro pixel formats like YUV, it is the width in sub pixels
VmbUint32_t	Height	The height of the image in pixels
VmbInt32_t	Stride	The offset from the current line to the next line, a value not equal to Width is currently not supported
VmbPixelInfo	PixelInfo	Information about the pixel format

5.3 VmbPixelInfo

VmbPixelInfo describes the pixel format of an image.

```
typedef struct VmbPixelInfo
{
    VmbUint32_t         BitsPerPixel;
    VmbUint32_t         BitsUsed;
    VmbAlignment_t      Alignment;
    VmbEndianness_t     Endianness;
    VmbPixelLayout_t    PixelLayout;
    VmbBayerPattern_t   BayerPattern;
    VmbColorSpace_t     Reserved;
};
```

Type	Name	Description
VmbUint32_t	BitsPerPixel	Number of bits for one image pixel, or sub pixel in macro pixel formats
VmbUint32_t	BitsUsed	Number of bits used per pixel, e.g. RGB12 has 48 bits per pixel and 36 bits used
VmbAlignment_t	Alignment	For image formats where BitsPerPixel is not equal to BitsUsed, the alignment specifies the bit layout of the pixel
VmbEndianness_t	Endianness	Specifies the endianness of pixels that are larger than one byte
VmbPixelLayout_t	PixelLayout	Describes the layout of the pixel component, e.g., RGB or BGR layout
VmbBayerPattern_t	BayerPattern	For raw image data, this field specifies the color filter array layout of the sensor

5.4 VmbTransformInfo

Optional transformation parameters are used to invoke special and additional functionality, which is applied while interpreting the source image (debayering function) or transforming the images (gamma and color correction).

```
VmbError_t VmbSetDebayerMode( VmbDebayerMode_t  DebayerMode,
                              VmbTransformInfo *TransformInfo);

VmbError_t VmbSetColorCorrectionMatrix3x3( const VmbFloat_t *Matrix,
                                           VmbTransformInfo *TransformInfo);

VmbError_t VmbSetGammaCorrection( VmbFloat_t        Gamma,
                                  VmbTransformInfo *TransformInfo);
```

	Type	Name	Description
in	`VmbDebayerMode_t`	DebayerMode	The mode used for debayering the source raw image, default mode is 2x2 debayering. Debayering modes only work for image widths and heights divisible by two.
in	`VmbFloat_t*`	Matrix	Color correction matrix
in	`VmbFloat_t`	Gamma	Gamma correction, currently not supported
in/out	`VmbTransformInfo_t*`	TransformInfo	Optional parameter that contains information about special transform functionality.

DebayerMode:
Please note that debayering is only applicable to image formats with both an even width and an even height.

Value	Description
`VmbDebayerMode2x2`	2x2 with green averaging
`VmbDebayerMode3x3`	3x3 with equal green weighting per line
`VmbDebayerModeLCAA`	Debayering with local color anti-aliasing
`VmbDebayerModeLCAAV`	Debayering with local color anti-aliasing and vertical sub-sampling
`VmbDebayerModeYUV422`	Debayering with YUV422 vertical sub-sampling

Matrix:
A 3x3 row order float matrix for color correcting the image.

$$\begin{pmatrix} rr & rg & rb \\ gr & gg & gb \\ br & bg & bb \end{pmatrix}$$

```
VmbTransformInfo    info[2];
VmbFloat_t          mat[9];
VmbSetDebayerMode(VmbDebayerMode2x2, &info[0]  );
VmbSetColorCorrectionMatrix3x3(mat, &info[1] );
```