



The APEX-CV Base Library

UG-10328-01-05

Copyright

Copyright © 2015 CogniVue Corporation ("CogniVue") All rights reserved.

This document contains information which is proprietary to CogniVue and may be used for non-commercial purposes within your organization in support of CogniVue's products. No other use or transmission of all or any part of this document is permitted without written permission from CogniVue, and must include all copyright and other proprietary notices. Use or transmission of all or any part of this document in violation of any applicable Canadian or other legislation is hereby expressly prohibited.

User obtains no rights in the information or in any product, process, technology or trademark which it includes or describes, and is expressly prohibited from modifying the information or creating derivative works without the express written consent of CogniVue.

Disclaimer

CogniVue assumes no responsibility for the accuracy or completeness of the information presented which is subject to change without notice. In no event will CogniVue be liable for any direct, indirect, special, incidental or consequential damages, including lost profits, lost business or lost data, resulting from the use of or reliance upon the information, whether or not CogniVue has been advised of the possibility of such damages.

Mention of non-CogniVue products or services is for information purposes only and constitutes neither an endorsement nor a recommendation.

Uncontrolled Copy

The master of this document is stored on CogniVue's document management system Docushare / Confluence. Viewing of the master electronically ensures access to the current issue. Any hardcopies are considered uncontrolled copies.

Version	Details of Change	Author	Date
01	Initial Revision	Doxygen	July 4, 2014
02	Update to add Median, Convolve, Sobel, Prewitt filters, RGB2Y and RGB2YUV color conversion, Histogram and Integral Image	C. Garrard	August 1, 2014
03	Update to add Bilinear/Linear Interpolation, Accumulate, and Accumulate Squared	A. Saechao, J. Cairns	October 1, 2014
04	Update to supported box, dilate, and sobel data types	A. Saechao	October 31, 2014
05	APEX-CV base update: abs, clz	G. Billig	January 29, 2015

Contents

1	APEX-CV Base Library	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	4
3.1	apex::Arithmetic Class Reference	4
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.2.1	Arithmetic	5
3.1.2.2	~Arithmetic	5
3.1.3	Member Function Documentation	5
3.1.3.1	abs	5
3.1.3.2	absdiff	6
3.1.3.3	accumulate	6
3.1.3.4	accumulateSquared	6
3.1.3.5	accumulateWeighted	7
3.1.3.6	add	7
3.1.3.7	bitwiseAND	8
3.1.3.8	bitwiseNOT	8
3.1.3.9	bitwiseOR	9
3.1.3.10	bitwiseXOR	9
3.1.3.11	clz	10
3.1.3.12	magnitude	10
3.1.3.13	subtract	11
3.1.3.14	threshold	11
3.2	apex::ColorConverter Class Reference	12
3.2.1	Detailed Description	12
3.2.2	Member Enumeration Documentation	12

3.2.2.1	ConversionType	12
3.2.3	Constructor & Destructor Documentation	12
3.2.3.1	ColorConverter	12
3.2.3.2	~ColorConverter	13
3.2.4	Member Function Documentation	13
3.2.4.1	convert	13
3.3	apex::Histogram Class Reference	13
3.3.1	Detailed Description	13
3.3.2	Constructor & Destructor Documentation	14
3.3.2.1	Histogram	14
3.3.2.2	~Histogram	14
3.3.3	Member Function Documentation	14
3.3.3.1	exec	14
3.4	apex::ImageFilter Class Reference	14
3.4.1	Detailed Description	15
3.4.2	Member Enumeration Documentation	15
3.4.2.1	PrewittType	15
3.4.2.2	SobelType	15
3.4.3	Constructor & Destructor Documentation	16
3.4.3.1	ImageFilter	16
3.4.3.2	~ImageFilter	16
3.4.4	Member Function Documentation	16
3.4.4.1	bilateralFilter	16
3.4.4.2	boxFilter	16
3.4.4.3	convolveFilter	17
3.4.4.4	dilateFilter	17
3.4.4.5	erodeFilter	18
3.4.4.6	gaussianFilter	18
3.4.4.7	medianFilter	19
3.4.4.8	prewittFilter	19
3.4.4.9	sobelFilter	20
3.5	apex::IntegrallImage Class Reference	21
3.5.1	Detailed Description	21
3.5.2	Constructor & Destructor Documentation	21
3.5.2.1	IntegrallImage	21
3.5.2.2	~IntegrallImage	21
3.5.3	Member Function Documentation	21

3.5.3.1	exec	21
3.6	apex::Interpolation Class Reference	22
3.6.1	Detailed Description	22
3.6.2	Constructor & Destructor Documentation	22
3.6.2.1	Interpolation	22
3.6.2.2	~Interpolation	22
3.6.3	Member Function Documentation	23
3.6.3.1	bilinearGrayscale	23
3.6.3.2	linearGrayscale	23
Bibliography		25
Index		26

Chapter 1

APEX-CV Base Library

The APEX-CV Base library provides basic functionality for developers to design their own imaging-based applications while taking advantage of CogniVue's massively parallel APEX architecture. Currently various arithmetic operations, color conversions and image filters are provided as well as image calculations such as histogram and integral image as listed below. The maximum resolution image supported is 512 pixel width with 32 CUs and 1024 with 64 CUs.

- Arithmetic Operations:
 - Absolute difference
 - Accumulate
 - Accumulate squared
 - Accumulate weighted
 - Addition
 - Bitwise AND, NOT, OR, XOR
 - Magnitude
 - Subtraction
 - Thresholding
- Interpolation:
 - Bilinear Grayscale
 - Linear Grayscale
- Color Conversions:
 - RGB565 to RGB888
 - RGB888 to RGB565
 - RGB888 to Y
 - RGB888 to YUV
- Image Filters:
 - Bilateral filter
 - Box filter
 - Convolve filter
 - Dilate filter

- Erode filter
 - Gaussian filter
 - Median filter
 - Prewitt filter
 - Sobel filter
- Histogram
- Integral Image

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

apex::Arithmetic	
Arithmetic class containing various arithmetic functions	4
apex::ColorConverter	
Color converter class containing support for converting image color types	12
apex::Histogram	
Histogram class containing histogram support	13
apex::ImageFilter	
Image filter class containing various image filters	14
apex::IntegralImage	
Integral Image class containing integral image support	21
apex::Interpolation	
Image interpolation class containing various interpolation methods	22

Chapter 3

Class Documentation

3.1 apex::Arithmetic Class Reference

Arithmetic class containing various arithmetic functions.

```
#include <ApexCV_base.hpp>
```

Public Member Functions

- **Arithmetic** ()
Default constructor.
- **~Arithmetic** ()
Destructor.
- int **abs** (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)
Absolute value function.
- int **absdiff** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Absolute difference function.
- int **accumulate** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Accumulate function.
- int **accumulateSquared** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst, unsigned char scale)
Accumulate Squared function.
- int **accumulateWeighted** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst, unsigned char alpha)
Accumulate Weighted function.
- int **add** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Addition function.
- int **bitwiseAND** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Bitwise AND function.
- int **bitwiseNOT** (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)
Bitwise NOT function.
- int **bitwiseOR** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Bitwise OR function.

- int **bitwiseXOR** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Bitwise XOR function.
- int **clz** (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)
Count leading zeros function.
- int **magnitude** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Magnitude function.
- int **subtract** (const ICP_ContigDataDesc &src0, const ICP_ContigDataDesc &src1, ICP_ContigDataDesc &dst)
Subtraction function.
- int **threshold** (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, unsigned int threshold)
Threshold function.

3.1.1 Detailed Description

Arithmetic class containing various arithmetic functions.

This class is an interface for using arithmetic functions on the host.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 apex::Arithmetic::Arithmetic ()

Default constructor.

3.1.2.2 apex::Arithmetic::~~Arithmetic ()

Destructor.

3.1.3 Member Function Documentation

3.1.3.1 int apex::Arithmetic::abs (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst)

Absolute value function.

Calculates the absolute value of *src0* buffer and stores the result in *dst* buffer.

Supported datatypes are:

- signed 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.2 `int apex::Arithmetic::absdiff (const ICP_ContigDataDesc & src0, const ICP_ContigDataDesc & src1, ICP_ContigDataDesc & dst)`

Absolute difference function.

Calculates the absolute difference between *src0* and *src1* buffers and stores the result in *dst* buffer.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.3 `int apex::Arithmetic::accumulate (const ICP_ContigDataDesc & src0, const ICP_ContigDataDesc & src1, ICP_ContigDataDesc & dst)`

Accumulate function.

Calculates the addition of the *src0* and *src1* buffers and stores the result in *dst* buffer.

Supported datatypes are:

- unsigned 8 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.4 `int apex::Arithmetic::accumulateSquared (const ICP_ContigDataDesc & src0, const ICP_ContigDataDesc & src1, ICP_ContigDataDesc & dst, unsigned char scale)`

Accumulate Squared function.

Calculates the addition of the *src0* and *src1* (squared) buffers and stores the result in *dst* buffer.

Supported datatypes are:

- unsigned 8 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>scale</i>	Shift amount. Value of [0, 15]

3.1.3.5 int apex::Arithmetic::accumulateWeighted (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*, unsigned char *alpha*)

Accumulate Weighted function.

Accumulate weighted uses the equation: $dst = src_0 * (1 - \frac{alpha}{256}) + src_1 * \frac{alpha}{256}$.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>alpha</i>	Weight amount. 8 bit fixed point value of [0, 1)

3.1.3.6 int apex::Arithmetic::add (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*)

Addition function.

Adds *src0* and *src1* buffers and stores the result in *dst* buffer. In case of overflow the result is saturated.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- signed 16 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.7 int apex::Arithmetic::bitwiseAND (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*)

Bitwise AND function.

Bitwise ANDs *src0* and *src1* buffers and stores the result in *dst* buffer. See (http://en.wikipedia.org/wiki/Bitwise_operation) for more information on bitwise operations.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 16 bit to unsigned 16 bit
- unsigned 32 bit to unsigned 32 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.8 int apex::Arithmetic::bitwiseNOT (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Bitwise NOT function.

Bitwise NOTs *src* buffer and stores the result in *dst* buffer. See (http://en.wikipedia.org/wiki/Bitwise_operation) for more information on bitwise operations.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.9 int apex::Arithmetic::bitwiseOR (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*)

Bitwise OR function.

Bitwise ORs *src0* and *src1* buffers and stores the result in *dst* buffer. See (http://en.wikipedia.org/wiki/Bitwise_operation) for more information on bitwise operations.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 16 bit to unsigned 16 bit
- unsigned 32 bit to unsigned 32 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.10 int apex::Arithmetic::bitwiseXOR (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*)

Bitwise XOR function.

Bitwise XORs *src0* and *src1* buffers and stores the result in *dst* buffer. See (http://en.wikipedia.org/wiki/Bitwise_operation) for more information on bitwise operations.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 16 bit to unsigned 16 bit
- unsigned 32 bit to unsigned 32 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.11 int apex::Arithmetic::clz (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Count leading zeros function.

Counts the leading zeros of *src0* buffer and stores the result in *dst* buffer.

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- signed 8 bit to unsigned 8 bit
- unsigned 16 bit to unsigned 8 bit
- signed 16 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.12 int apex::Arithmetic::magnitude (const ICP_ContigDataDesc & *src0*, const ICP_ContigDataDesc & *src1*, ICP_ContigDataDesc & *dst*)

Magnitude function.

Calculates the magnitude between two sources following the format $dst = \sqrt{src_0^2 + src_1^2}$. The result is floored to the nearest integer. Note if you want to calculate the gradient magnitude you must calculate the image gradients and pass them in to this function.

Supported datatypes are:

- signed 16 bit to unsigned 16 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.13 `int apex::Arithmetic::subtract (const ICP_ContigDataDesc & src0, const ICP_ContigDataDesc & src1, ICP_ContigDataDesc & dst)`

Subtraction function.

Subtracts *src0* from *src1* buffers and stores the result in *dst*. In case of over/underflow the result is saturated.

Supported datatypes are:

- unsigned 8 bit to signed 16 bit
- signed 16 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src0</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
<i>src1</i>	Source 1 memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.1.3.14 `int apex::Arithmetic::threshold (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst, unsigned int threshold)`

Threshold function.

Returns an 8 bit boolean image based on the equation: $dst = \begin{cases} 255 & \text{if } src > threshold \\ 0 & \text{otherwise} \end{cases}$

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 16 bit to unsigned 8 bit
- unsigned 32 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source 0 memory buffer. See ICP_ContigDataDesc [1]
------------	--

<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>threshold</i>	Threshold value.

3.2 apex::ColorConverter Class Reference

Color converter class containing support for converting image color types.

```
#include <ApexCV_base.hpp>
```

Public Types

- enum [ConversionType](#) { [RGB565_TO_RGB888](#), [RGB888_TO_RGB565](#), [RGB888_TO_Y](#), [RGB888_TO_YUV](#) }
List of conversion types.

Public Member Functions

- [ColorConverter](#) ()
Default constructor.
- [~ColorConverter](#) ()
Destructor.
- int [convert](#) (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, [ConversionType](#) ct, int R2YFactor=0, int G2YFactor=0, int B2YFactor=0)
Convert function.

3.2.1 Detailed Description

Color converter class containing support for converting image color types.

This class is an interface for using color conversion functions on the host.

3.2.2 Member Enumeration Documentation

3.2.2.1 enum apex::ColorConverter::ConversionType

List of conversion types.

Enumerator

- RGB565_TO_RGB888*** 16 bit RGB565 to 32 bit representation of RGB888X
RGB888_TO_RGB565 32 bit representation of RGB888X to 16 bit RGB565
RGB888_TO_Y 4-tuple 8 bit R, G, B, X to 8 bit Y
RGB888_TO_YUV 4-tuple 8 bit R, G, B, X to 4-tuple 8 bit Y, U, V, X

3.2.3 Constructor & Destructor Documentation

3.2.3.1 apex::ColorConverter::ColorConverter ()

Default constructor.

3.2.3.2 apex::ColorConverter::~~ColorConverter ()

Destructor.

3.2.4 Member Function Documentation

3.2.4.1 int apex::ColorConverter::convert (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst, ConversionType ct, int R2YFactor = 0, int G2YFactor = 0, int B2YFactor = 0)

Convert function.

Converts an image from one type to another based on [ConversionType](#).

R2YFactor, G2YFactor and B2YFactor are Q0.8 fixed point values used with RGB888_TO_Y following the formula:

$$Y = \left\lfloor \frac{R2YFactor}{256} * R + \frac{G2YFactor}{256} * G + \frac{B2YFactor}{256} * B + 0.5 \right\rfloor$$

For example, conversion following Recommendation ITU-R BT.601-7 (<http://www.itu.int/rec/R-REC-B-T.601-7-201103-I/en>) would use factor values of 77(0.299), 150(0.587) and 29(0.114).

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>ct</i>	Color conversion type. See ConversionType
<i>R2YFactor</i>	Conversion factor for red used with RGB888_TO_Y
<i>G2YFactor</i>	Conversion factor for green used with RGB888_TO_Y
<i>B2YFactor</i>	Conversion factor for blue used with RGB888_TO_Y

3.3 apex::Histogram Class Reference

[Histogram](#) class containing histogram support.

```
#include <ApexCV_base.hpp>
```

Public Member Functions

- [Histogram](#) ()
Default constructor.
- [~Histogram](#) ()
Destructor.
- int [exec](#) (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)
Histogram execution.

3.3.1 Detailed Description

[Histogram](#) class containing histogram support.

This class is an interface for using the histogram kernel on the host.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 apex::Histogram::Histogram ()

Default constructor.

3.3.2.2 apex::Histogram::~~Histogram ()

Destructor.

3.3.3 Member Function Documentation

3.3.3.1 int apex::Histogram::exec (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst)

[Histogram](#) execution.

Calculates an image histogram (http://en.wikipedia.org/wiki/Image_histogram) from the *src* buffer and returns the result in *dst* buffer. The resulting histogram has 256 bins each representing the respective number of pixel values found in the *src* image.

Supported datatypes are:

- unsigned 8 bit to unsigned 32 bit bins

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.4 apex::ImageFilter Class Reference

Image filter class containing various image filters.

```
#include <ApexCV_base.hpp>
```

Public Types

- enum [SobelType](#) { [SOBEL_X](#), [SOBEL_Y](#), [SOBEL_BOTH](#) }
List of sobel filter types.
- enum [PrewittType](#) { [PREWITT_X](#), [PREWITT_Y](#) }
List of prewitt filter types.

Public Member Functions

- [ImageFilter](#) ()

- Default constructor.*
- `~ImageFilter ()`
Destructor.
- `int bilateralFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, int sigmaColor, int sigmaSpace)`
Bilateral image filter.
- `int boxFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)`
Box image filter.
- `int dilateFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)`
Dilate image filter.
- `int erodeFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)`
Erode image filter.
- `int medianFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, int windowSize)`
Median filter.
- `int gaussianFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)`
Gaussian image filter.
- `int sobelFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, int windowSize, SobelType st=SOBEL_BOTH)`
Sobel filter.
- `int convolveFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, const signed char *filterCoeff, int windowSize=3, int filterScale=0)`
Convolve filter.
- `int prewittFilter (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, PrewittType pt=PREWITT_X)`
Prewitt filter.

3.4.1 Detailed Description

Image filter class containing various image filters.

This class is an interface for using image filters on the host. Filter windows that exceed the boundaries of the image use replicated pixels for the padding.

3.4.2 Member Enumeration Documentation

3.4.2.1 enum apex::ImageFilter::PrewittType

List of prewitt filter types.

Enumerator

PREWITT_X Apply prewitt filter in x direction

PREWITT_Y Apply prewitt filter in y direction

3.4.2.2 enum apex::ImageFilter::SobelType

List of sobel filter types.

Enumerator

SOBEL_X Apply sobel filter in x direction

SOBEL_Y Apply sobel filter in y direction

SOBEL_BOTH Apply sobel filter in both x and y direction then sum their absolute values

3.4.3 Constructor & Destructor Documentation

3.4.3.1 apex::ImageFilter::ImageFilter ()

Default constructor.

3.4.3.2 apex::ImageFilter::~~ImageFilter ()

Destructor.

3.4.4 Member Function Documentation

3.4.4.1 int apex::ImageFilter::bilateralFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*, int *sigmaColor*, int *sigmaSpace*)

Bilateral image filter.

Applies a bilateral filter to *src*. *sigmaColor* represents the weight of color/intensity difference and *sigmaSpace* represents the weight of spacial difference. See: [2] for more information.

Supported window size is:

- 5 x 5

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>sigmaColor</i>	Sigma value for color space.
<i>sigmaSpace</i>	Sigma value for distance space.

3.4.4.2 int apex::ImageFilter::boxFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Box image filter.

Applies a box filter to *src* buffer. Each *dst* buffer pixel is calculated as follows: $dst = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * src$

Supported window size is:

- 3 x 3

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- signed 16 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1].
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.4.4.3 int apex::ImageFilter::convolveFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*, const signed char * *filterCoeff*, int *windowSize* = 3, int *filterScale* = 0)

Convolve filter.

Applies a user defined filter with coefficients *filterCoeff* to *src* buffer and returns the result in *dst* buffer. *filterScale* can be used to divide each filtered pixel by a value of $2^{filterScale}$. If the resulting pixel would overflow the *dst* datatype its value is saturated.

Supported window sizes are:

- 3 x 3
- 5 x 5

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 8 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>filterCoeff</i>	Array containing the filter coefficients to be applied.
<i>windowSize</i>	Defines a filter window with dimensions <i>windowSize</i> x <i>windowSize</i> . Default is 3. See supported window sizes.
<i>filterScale</i>	Optional: used to scale the resulting filtered pixel by $2^{filterScale}$. Supported range of [0, 16].

3.4.4.4 int apex::ImageFilter::dilateFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Dilate image filter.

Applies a dilate filter to *src* buffer. Each *dst* buffer pixel is the maximum pixel value contained in the window of the respective *src* buffer pixel.

Supported window size is:

- 3 x 3

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- signed 16 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.4.4.5 int apex::ImageFilter::erodeFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Erode image filter.

Applies an erode filter to *src* buffer. Each *dst* buffer pixel is the minimum pixel value contained in the window of the respective *src* buffer pixel.

Supported window size is:

- 3 x 3

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.4.4.6 int apex::ImageFilter::gaussianFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*)

Gaussian image filter.

Applies a gaussian filter to *src* buffer. Each *dst* buffer pixel is calculated as follows: $dst = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * src$

Supported window size is:

- 3 x 3

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1].
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.4.4.7 int apex::ImageFilter::medianFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*, int *windowSize*)

Median filter.

Applies a median filter to *src* buffer. Each *dst* buffer pixel is the median pixel value contained in the window of the respective *src* buffer pixel.

Supported window sizes are:

- 3 x 3
- 5 x 5

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>windowSize</i>	Defines a filter window with dimensions <i>windowSize</i> x <i>windowSize</i> . Default is 3. See supported window sizes.

3.4.4.8 int apex::ImageFilter::prewittFilter (const ICP_ContigDataDesc & *src*, ICP_ContigDataDesc & *dst*, PrewittType *pt* = PREWITT_X)

Prewitt filter.

Applies a prewitt filter based on [PrewittType](#) to *src* buffer and returns the result in *dst* buffer. Similar to the [sobelFilter](#), prewitt uses the following filter coefficients:

$$filter_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, filter_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

Supported formats are:

- x direction
- y direction

Supported window size is:

- 3 x 3

Supported datatypes are:

- unsigned 8 bit to signed 16 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>pt</i>	Specify which prewitt filter type to use. Default is PREWITT_X. See PrewittType .

3.4.4.9 int apex::ImageFilter::sobelFilter (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst, int windowSize, SobelType st = SOBEL_BOTH)

Sobel filter.

Applies a sobel filter based on [SobelType](#) to *src* buffer and returns the result in *dst* buffer. SOBEL_BOTH calculates both SOBEL_X and SOBEL_Y images and then sums their absolute values following the formula: $dst(i, j) = |SOBEL_X(i, j)| + |SOBEL_Y(i, j)|$

The 3x3 filter coefficients for x and y directions are:

$$filter_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, filter_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

The 5x5 filter coefficients for x and y directions are:

$$filter_x = \begin{bmatrix} -1 & -2 & 0 & +2 & +1 \\ -4 & -8 & 0 & +8 & +4 \\ -6 & -12 & 0 & +12 & +6 \\ -4 & -8 & 0 & +8 & +4 \\ -1 & -2 & 0 & +2 & +1 \end{bmatrix}, filter_y = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ +2 & +8 & +12 & +8 & +2 \\ +1 & +4 & +6 & +4 & +1 \end{bmatrix}$$

Supported formats are:

- x direction
- y direction
- both directions

Supported window sizes are:

- 3 x 3
- 5 x 5

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit
- unsigned 8 bit to signed 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]
<i>windowSize</i>	Defines a filter window with dimensions <i>windowSize</i> x <i>windowSize</i> . Default is 3. See supported window sizes.
<i>st</i>	Specify which sobel filter type to use. Default is SOBEL_BOTH. See SobelType .

3.5 apex::IntegrallImage Class Reference

Integral Image class containing integral image support.

```
#include <ApexCV_base.hpp>
```

Public Member Functions

- [IntegrallImage](#) ()
Default constructor.
- [~IntegrallImage](#) ()
Destructor.
- int [exec](#) (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst)
Integral Image execution.

3.5.1 Detailed Description

Integral Image class containing integral image support.

This class is an interface for using the integral image kernel on the host.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 apex::IntegrallImage::IntegrallImage ()

Default constructor.

3.5.2.2 apex::IntegrallImage::~~IntegrallImage ()

Destructor.

3.5.3 Member Function Documentation

3.5.3.1 int apex::IntegrallImage::exec (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst)

Integral Image execution.

Calculates an integral image (http://en.wikipedia.org/wiki/Summed_area_table) from the *src* buffer and returns the result in *dst* buffer. Each *dst* pixel represents the sum of all *src* pixels left of and above the *dst* pixel including the *src* pixel

Supported datatypes are:

- unsigned 8 bit to unsigned 32 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer. See ICP_ContigDataDesc [1]
<i>dst</i>	Destination memory buffer. See ICP_ContigDataDesc [1]

3.6 apex::Interpolation Class Reference

Image interpolation class containing various interpolation methods.

```
#include <ApexCV_base.hpp>
```

Public Member Functions

- [Interpolation](#) ()
Default constructor.
- [~Interpolation](#) ()
Destructor.
- int [linearGrayscale](#) (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, ICP_ContigDataDesc &deltaX)
Linear Grayscale interpolation.
- int [bilinearGrayscale](#) (const ICP_ContigDataDesc &src, ICP_ContigDataDesc &dst, ICP_ContigDataDesc &delta)
Bilinear Grayscale interpolation.

3.6.1 Detailed Description

Image interpolation class containing various interpolation methods.

This class is an interface for using image interpolations on the host. [Interpolation](#) windows that exceed the boundaries of the image use replicated pixels for the padding.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 apex::Interpolation::Interpolation ()

Default constructor.

3.6.2.2 apex::Interpolation::~~Interpolation ()

Destructor.

3.6.3 Member Function Documentation

3.6.3.1 `int apex::Interpolation::bilinearGrayscale (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst, ICP_ContigDataDesc & delta)`

Bilinear Grayscale interpolation.

Applies bilinear interpolation to *src*. *delta* represents the distance the resultant pixel was from the left adjacent pixels and from the top adjacent pixels <x, y>, normalized between 0-255.

Supported window size is:

- 2 x 2

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer.
<i>dst</i>	Destination memory buffer.
<i>delta</i>	Delta XY values for interpolation.

3.6.3.2 `int apex::Interpolation::linearGrayscale (const ICP_ContigDataDesc & src, ICP_ContigDataDesc & dst, ICP_ContigDataDesc & deltaX)`

Linear Grayscale interpolation.

Applies linear interpolation to *src*. *deltaX* represents the distance the resultant pixel was from the left adjacent pixel, normalized between 0-255.

Supported window size is:

- 2 x 1

Supported image size is:

- Width: multiple of 2
- Height: any

Supported datatypes are:

- unsigned 8 bit to unsigned 8 bit

Returns

Error code (zero on success).

Parameters

<i>src</i>	Source memory buffer.
<i>dst</i>	Destination memory buffer.
<i>deltaX</i>	Delta X values for interpolation.

Bibliography

- [1] Acf.chm. Technical report, CogniVue, 2014. [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#), [22](#)
- [2] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998. [16](#)

Index

- ~Arithmetic
 - apex::Arithmetic, 5
- ~ColorConverter
 - apex::ColorConverter, 12
- ~Histogram
 - apex::Histogram, 14
- ~ImageFilter
 - apex::ImageFilter, 16
- ~IntegrallImage
 - apex::IntegrallImage, 21
- ~Interpolation
 - apex::Interpolation, 22
- abs
 - apex::Arithmetic, 5
- absdiff
 - apex::Arithmetic, 6
- accumulate
 - apex::Arithmetic, 6
- accumulateSquared
 - apex::Arithmetic, 6
- accumulateWeighted
 - apex::Arithmetic, 7
- add
 - apex::Arithmetic, 7
- apex::ColorConverter
 - RGB565_TO_RGB888, 12
 - RGB888_TO_RGB565, 12
 - RGB888_TO_Y, 12
 - RGB888_TO_YUV, 12
- apex::ImageFilter
 - PREWITT_X, 15
 - PREWITT_Y, 15
 - SOBEL_BOTH, 15
 - SOBEL_X, 15
 - SOBEL_Y, 15
- apex::Arithmetic, 4
 - ~Arithmetic, 5
 - abs, 5
 - absdiff, 6
 - accumulate, 6
 - accumulateSquared, 6
 - accumulateWeighted, 7
 - add, 7
 - Arithmetic, 5
 - bitwiseAND, 8
 - bitwiseNOT, 8
 - bitwiseOR, 9
 - bitwiseXOR, 9
 - clz, 10
 - magnitude, 10
 - subtract, 11
 - threshold, 11
 - apex::ColorConverter, 12
 - ~ColorConverter, 12
 - ColorConverter, 12
 - ConversionType, 12
 - convert, 13
 - apex::Histogram, 13
 - ~Histogram, 14
 - exec, 14
 - Histogram, 14
 - apex::ImageFilter, 14
 - ~ImageFilter, 16
 - bilateralFilter, 16
 - boxFilter, 16
 - convolveFilter, 17
 - dilateFilter, 17
 - erodeFilter, 18
 - gaussianFilter, 18
 - ImageFilter, 16
 - medianFilter, 19
 - prewittFilter, 19
 - PrewittType, 15
 - sobelFilter, 20
 - SobelType, 15
 - apex::IntegrallImage, 21
 - ~IntegrallImage, 21
 - exec, 21
 - IntegrallImage, 21
 - apex::Interpolation, 22
 - ~Interpolation, 22
 - bilinearGrayscale, 23
 - Interpolation, 22
 - linearGrayscale, 23
 - Arithmetic
 - apex::Arithmetic, 5
 - bilateralFilter
 - apex::ImageFilter, 16
 - bilinearGrayscale
 - apex::Interpolation, 23

bitwiseAND
 apex::Arithmetic, 8
bitwiseNOT
 apex::Arithmetic, 8
bitwiseOR
 apex::Arithmetic, 9
bitwiseXOR
 apex::Arithmetic, 9
boxFilter
 apex::ImageFilter, 16

clz
 apex::Arithmetic, 10
ColorConverter
 apex::ColorConverter, 12
ConversionType
 apex::ColorConverter, 12
convert
 apex::ColorConverter, 13
convolveFilter
 apex::ImageFilter, 17

dilateFilter
 apex::ImageFilter, 17

erodeFilter
 apex::ImageFilter, 18
exec
 apex::Histogram, 14
 apex::IntegrallImage, 21

gaussianFilter
 apex::ImageFilter, 18

Histogram
 apex::Histogram, 14

ImageFilter
 apex::ImageFilter, 16
IntegrallImage
 apex::IntegrallImage, 21
Interpolation
 apex::Interpolation, 22

linearGrayscale
 apex::Interpolation, 23

magnitude
 apex::Arithmetic, 10
medianFilter
 apex::ImageFilter, 19

PREWITT_X
 apex::ImageFilter, 15
PREWITT_Y
 apex::ImageFilter, 15

prewittFilter
 apex::ImageFilter, 19
PrewittType
 apex::ImageFilter, 15

RGB565_TO_RGB888
 apex::ColorConverter, 12
RGB888_TO_RGB565
 apex::ColorConverter, 12
RGB888_TO_Y
 apex::ColorConverter, 12
RGB888_TO_YUV
 apex::ColorConverter, 12

SOBEL_BOTH
 apex::ImageFilter, 15
SOBEL_X
 apex::ImageFilter, 15
SOBEL_Y
 apex::ImageFilter, 15
sobelFilter
 apex::ImageFilter, 20
SobelType
 apex::ImageFilter, 15
subtract
 apex::Arithmetic, 11

threshold
 apex::Arithmetic, 11