

FastCV SDK Release Notes

Version 1.7.1

Last Updated: 06/05/2019

1 Introduction

The *FastCV* SDK is a collection of algorithms implemented for ARM and optimized for Qualcomm's Snapdragon processor.

1.1 Platforms

The libraries currently supported are:

1. *Android* 32 bit and 64 bit library
2. IA-32 (x86) *Win32* and *MS Visual C++* 2010, 2012, and 2013.
3. IA-32 (x86) *Win64* and *MS Visual C++* 2012, and 2013.
4. Linux Embedded 32 bit ('softfp' and 'hard') and 64 bit library

1.2 Installation

The library and header files can be incorporated into a project in a variety of ways. No path is written into the software, so no specific installation path is required.

1.3 Operation Mode

To get the advantages of algorithms implemented for Qualcomm's Snapdragon processor, there are APIs should be called as part of initialization and de-initialization process.

1.3.1 Initialization process

As part of initialization process, below API

```
FASTCV_API int fcvSetOperationMode( fcvOperationMode mode )
```

should be called. An option should be selected based on the application goal.

Below are the available *fcvOperationMode* options

Operation mode	Description
FASTCV_OP_LOW_POWER	The QDSP implementation will be used unless the QDSP speed is 3 times slower than CPU speed.
FASTCV_OP_PERFORMANCE	The fastest implementation will be used.
FASTCV_OP_CPU_OFFLOAD	The QDSP implementation will be used when it's available, otherwise it will find for GPU and CPU implementation.
FASTCV_OP_CPU_PERFORMANCE	The CPU fastest implementation will be used.

1.3.2 De-initialization process

As part of de-initialization process, below API

```
FASTCV_API void fcvCleanUp( void );
```

should be called.

2 Changes

2.1 Release Numbering Scheme

Releases are generally motivated for the following reasons:

- Changes to previously released APIs
- Addition of new functions
- Performance improvements and/or bug fixes – also known as implementation modifications

Each motivation has a varying degree of impact on the user of the library.

The general release numbering scheme captures this variety of motivations. Given release ID: A.B.C

An increase in “A” indicates that a previously released API has changed, so a developer may encounter compilation issues which require modification of their code in order to adhere to the modified API. Qualcomm will make every effort to minimize these changes. Additionally, new functions and implementation modifications may be present.

An increase in “B” indicates that new functions have been added to the library, so additional functionality is available, however existing APIs have not changed. Additionally, implementation modifications may be present.

An increase in “C” indicates that implementation modifications only have been made.

2.2 Release 1.0.2

Initial external release.

2.3 Release 1.0.3

2.3.1 Existing API Modifications

Section 2.1 indicates that based on this release number, this release contains only implementation modifications. While that is the motivation of this release, there is a single API change which was deemed insignificant enough to add in this release without designating this as a major revision because it does not affect current callers.

fcvSumOfSquaredDiffs36x4s8:

Previous API:

```
FASTCV_API void
fcvSumOfSquaredDiffs36x4s8
(
    const int8_t* __restrict  a,
    float        invLenA,
    const int8_t* __restrict  b0,
    const int8_t* __restrict  b1,
    const int8_t* __restrict  b2,
    const int8_t* __restrict  b3,
    float* __restrict        invLenB,
    float* __restrict        distance );
s
```

New API:

```
FASTCV_API void
fcvSumOfSquaredDiffs36x4s8( const int8_t* __restrict a,
                             float          invLenA,
                             const int8_t* __restrict b0,
                             const int8_t* __restrict b1,
                             const int8_t* __restrict b2,
                             const int8_t* __restrict b3,
                             const float* __restrict invLenB,
                             float* __restrict  distances );
```

Reason:

This API was modified to indicate the “const-ness” of *invLenB.

2.3.2 New API Additions

There are no API additions in this release.

2.3.3 Bug Fixes

Jitter in FastCVSample – On some devices, there was jitter seen while using the FastCVSample app

FastCV KLT optical flow crashes for features close to borders.

Sample App fails on some devices if camera parameters aren't supported.

Running FastCVSample for long durations causes it to crash.

2.3.4 Performance Improvements

Performance improvements are generally categorized as either applying to either FastCV or FastCV for Snapdragon. FastCV level performance improvements will be experienced by users of the library who do not execute on Qualcomm devices, while FastCV for Snapdragon level performance improvements will be experienced by users of the library who do execute on qualcomm devices.

2.3.4.1 FastCV

There are no improvements added to this release.

2.3.4.2 FastCV for Snapdragon

All fcvHammingDistance functions have been optimized.

All fcvImageGradientSobel functions for the "width is multiple of 8" case have been optimized

All fcvBitCount functions have been optimized.

The fcvCornerFast9Scoreu8 has been optimized.

All fcvClusterEuclidean functions have been optimized.

The fcvCornerFast9u8 has been further optimized.

The fcvCornerFast9InMasku8 has been further optimized.

2.3.5 Other Changes

Sample code which demonstrates the loading and preparation of a JPEG image has been provided.

2.4 Release 1.0.4

2.4.1 Existing API Modifications

There are no API Modifications in this release. However, there have been modifications to the comments portion of the API header file.

2.4.2 New API Additions

There are no API additions in this release.

2.4.3 Bug Fixes

FastCVSample fails to run properly on Android ICS.

ImageGradient functions fail to use stride value.

HarrisCorner function memory access failure.

Access faults detected while executing fcvNCCPatchOnCircle8x8u8.

For FastCV for Snapdragon: fastcvEdgeSobel function does not restore ARM registers

2.4.4 Performance Improvements

2.4.4.1 FastCV

Performance Improvement for LKOpticalFlow

2.4.4.2 FastCV for Snapdragon

Performance Improvement for LKOpticalFlow, fcvImageDiffu8, fcvPyramidSobelGradient, and fcvScaleDownBy2Gaussian5x5u8.

2.4.5 Other Changes

None

2.5 Release 1.1.0

2.5.1 Existing API Modifications

There have been no API Modifications which would affect compatability with applications which used previous releases of the fastCV library. However, there were a significant number of functions which were determined to require function signature changes. In order to maintain backward compatability and continue to improve the fastcv library, the

following strategy is being applied to functions which were determined to require function signature change:

First, the existing API with its preexisting function signature is maintained. This allows the library to maintain backwards compatibility. Next, a new function is added to the library which shares the same name as the original function except that “_v2” (version #2) is appended to the name. The version 2 function is further modified to change the function signature as needed. The developer is encouraged to migrate to the “_v2” function to make use of the modified API, however, this migration is not mandatory for the current release. In a future release, release version 2.0.0, the original function with its original function signature will be removed and the _v2 function will be renamed to the original function name along with its function signature.

An example function progression would be:

Release 1.0.4:

```
void A (void);
```

Release 1.1.0:

```
void A (void);
```

```
void A_v2 (int x); // New function signature added
```

Release 2.0.0

```
void A (int x);
```

The following table identifies the functions which have function signature changes.

Function Name	Signature Change	Reason For Change
fcvFilterMedian3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterGaussian3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterGaussian5x5u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterGaussian11x11u8_v2	Add srcStride and dstStride	Stride Support
fcvColorRGB888toYCrCb8_v2	Add srcStride and dstStride	Stride Support
fcvFilterSobel3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterCanny3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvImageDiffu8_v2	Add srcStride and dstStride	Stride Support
fcvImageGradientInterleaveds16_v2	Add gradStride	Stride Support
fcvImageGradientInterleavedf32_v2	Add gradStride	Stride Support
fcvImageGradientPlanars16_v2	Add srcStride and dxyStride	Stride Support
fcvImageGradientPlanarf32_v2	Add srcStride and dxyStride	Stride Support
fcvFilterCorr3x3s8_v2	Add srcStride and dstStride	Stride Support
fcvFilterCorrSep9x9s16_v2	Add srcStride and dstStride	Stride Support
fcvFilterCorrSep11x11s16_v2	Add srcStride and dstStride	Stride Support
fcvFilterCorrSep13x13s16_v2	Add srcStride and dstStride	Stride Support
fcvFilterCorrSep15x15s16_v2	Add srcStride and dstStride	Stride Support
fcvFilterCorrSep17x17s16_v2	Add srcStride and dstStride	Stride Support
fcvIntegrateImageu8_v2	Add srcStride and dstStride	Stride Support
fcvIntegratePatchu8_v2	Add srcStride	Stride Support
fcvIntegratePatch12x12u8_v2	Add srcStride	Stride Support
fcvIntegratePatch18x18u8_v2	Add srcStride	Stride Support
fcvNCCPatchOnCircle8x8u8_v2	Add filterLowVariance	Add minimum variance support
fcvNCCPatchOnSquare8x8u8_v2	Add filterLowVariance	Add minimum variance support
fcvSumOfAbsoluteDiffs8x8u8_v2	Add srcStride and dstStride	Stride Support
fcvScaleDownBy2u8_v2	Add srcStride and dstStride	Stride Support
fcvScaleDownBy2Gaussian5x5u8_v2	Add srcStride and dstStride	Stride Support
fcvScaleDownBy4u8_v2	Add srcStride and dstStride	Stride Support
fcvScaleDownu8_v2	Add srcStride and dstStride	Stride Support
fcvScaleUpBy2Gaussian5x5u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterThresholdu8_v2	Add srcStride and dstStride	Stride Support
fcvFilterDilate3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvFilterErode3x3u8_v2	Add srcStride and dstStride	Stride Support
fcvTransformAffine8x8u8_v2	Add srcStride and patchStride	Stride Support
fcvWarpPerspectiveu8_v2	Add srcStride and dstStride	Stride Support
fcv3ChannelWarpPerspectiveu8_v2	Add srcStride and dstStride	Stride Support
fcvFilterGaussian5x5s16_v2	Add srcStride and dstStride	Stride Support
fcvFilterGaussian5x5s32_v2	Add srcStride and dstStride	Stride Support
fcvTransformAffineu8_v2	Add srcStride and patchStride	Stride Support
fcvImageGradientSobelPlanars16_v2	Add srcStride and dxyStride	Stride Support
fcvImageGradientSobelInterleaveds16_v2	Add srcStride and gradStride	Stride Support
fcvImageGradientSobelInterleavedf32_v2	Add srcStride and gradStride	Stride Support
fcvImageGradientSobelPlanars8_v2	Add srcStride and dxyStride	Stride Support
fcvImageGradientSobelPlanarf32_v2	Add srcStride and dxyStride	Stride Support
fcvImageGradientSobelPlanarf32f32_v2	Add dxyStride. Also changed srcStride semantics to indicate byte (previous function interpreted srcStride to indicate pixel stride).	Stride Support

A general note on the semantics of stride. In previous releases, the interpretation of stride was ambiguous. Starting in the 1.1.0 release, an effort was made to remove that ambiguity by ensuring that all usage of stride are consistently interpreted as the number of bytes to stride. As most of the functions in 1.0.4 library operate on images where pixels are expressed as bytes, this ambiguity was acceptable (as either interpretation of stride as meaning either bytes or pixels resulted in the same operation). As new functions have been added into 1.1.0, however, many functions have several bytes per pixel, so this disambiguation was necessary.

2.5.2 New API Additions

The 1.1.0 release of the fastCV library contains a substantial set of new functions.

The following table identifies the new functions which are added to the fastcv library. Please see individual function documentation for further information.

fcvImageDiff32	fcvScaleDownMNU8
fcvImageDiffu8f32	fcvScaleDownMNInterleavedu8
fcvImageDiffu8s8	fcvKMeansTreeSearch36x10s8
fcvMserInit	fcvLinearSearchPrepare8x36s8
fcvMserRelease	fcvLinearSearch8x36s8
fcvMseru8	fcvFindContoursExternalu8
fcvMserExtu8	fcvFindContoursListu8
fcvLocalHarrisMaxu8	fcvFindContoursCcompu8
fcvGeom3PointPoseEstimatef32	fcvFindContoursTreeu8
fcvScaleDown3To2u8	fcvFindContoursAllocate
fcvScaleDownNNU8	fcvFindContoursDelete
fcvSort8Scoresf32	fcvSolvef32
fcvTrackBMOpticalFlow16x16u8	fcvGetPerspectiveTransformf32
fcvBitwiseOru8	fcvSetElementsu8
fcvBitwiseOrs32	fcvSetElementss32
fcvColorRGB888ToGrayu8	fcvSetElementsf32
fcvTiltedIntegralu8s32	fcvSetElementsc4u8
fcvConvValid16	fcvSetElementsc4s32
fcvBoundingRectangle	fcvSetElementsc4f32
fcvUpsampleVerticalu8	fcvSetElementsc3u8
fcvUpsampleHorizontalu8	fcvSetElementsc3s32
fcvUpsample2DU8	fcvSetElementsc3f32
fcvUpsampleVerticalInterleavedu8	fcvAdaptiveThresholdGaussian3x3u8
fcvUpsampleHorizontalInterleavedu8	fcvAdaptiveThresholdGaussian5x5u8
fcvUpsample2DInterleavedu8	fcvAdaptiveThresholdGaussian11x11u8
fcvColorYCbCr444PlanarToRGB565u8	fcvAdaptiveThresholdMean3x3u8
fcvColorYCbCr444PlanarToRGB888u8	fcvAdaptiveThresholdMean5x5u8
fcvColorYCbCr444PlanarToRGBA8888u8	fcvAdaptiveThresholdMean11x11u8
fcvColorYCbCr422PlanarToRGB565u8	fcvBoxFilter3x3u8
fcvColorYCbCr422PlanarToRGB888u8	fcvBoxFilter5x5u8
fcvColorYCbCr422PlanarToRGBA8888u8	fcvBoxFilter11x11u8
fcvColorYCbCr420PlanarToRGB565u8	fcvBilateralFilter5x5u8
fcvColorYCbCr420PlanarToRGB888u8	fcvBilateralFilter7x7u8
fcvColorYCbCr420PlanarToRGBA8888u8	fcvBilateralFilter9x9u8
fcvColorYCbCr444PseudoPlanarToRGB565u8	fcvSegmentFGMasku8
fcvColorYCbCr444PseudoPlanarToRGB888u8	fcvAbsDiffu8
fcvColorYCbCr444PseudoPlanarToRGBA8888u8	fcvAbsDiffs32
fcvColorYCbCr422PseudoPlanarToRGB565u8	fcvAbsDiffs32
fcvColorYCbCr422PseudoPlanarToRGB888u8	fcvAbsDiffVu8

fcvColorYCbCr422PseudoPlanarToRGBA8888u8	fcvAbsDiffVs32
fcvColorYCbCr420PseudoPlanarToRGB565u8	fcvAbsDiffVf32
fcvColorYCbCr420PseudoPlanarToRGB888u8	fcvAbsDiffVc4u8
fcvColorYCbCr420PseudoPlanarToRGBA8888u8	fcvAbsDiffVs32c4
fcvEdgeWeightings16	fcvAbsDiffVc4f32
fcvDeinterleaveu8	fcvAbsDiffVc3u8
fcvInterleaveu8	fcvAbsDiffVc3s32
fcvDWT Harr Transposeu8	fcvAbsDiffVc3f32
fcvDWT53TabTransposes16	fcvKDTreeCreate36s8f32
fcvIDWT53TabTransposes16	fcvKDTreeDestroy36s8f32
fcvIDWT Harr Transposes16	fcvKDTreeQuery36s8f32

2.5.3 Bug Fixes

fcvGeomHomographyEvaluatef32 was performing an improper comparison

fcvIntegratePatchu8 was spanning out of bounds.

2.5.4 Performance Improvements

2.5.4.1 FastCV

There are no significant performance improvements in the existing 1.0.4 functions. Cannot characterize the improvements for the newly added 1.1.0 functions as there is nothing to which to compare the functions' improvements.

2.5.4.2 FastCV for Snapdragon

Many of the Snapdragon supported functions that existed in the 1.0.4 have been further accelerated as have the majority of the newly added 1.1.0 functions.

2.5.5 Other Changes

The fastcv library now ships with a selection of Qualcomm enabled Android devices. When an application which uses fastcv is executing on such a device, the device implementation of a given fastcv function is executed rather than the statically linked implementation.

Should a newly added function be called that is not supported on such a device, then the code will revert to the statically linked implementation.

The intention of this mechanism is to allow full utilization of Qualcomm hardware resources which are often unable to be utilized as a static library.

2.6 Release 1.1.1

2.6.1 Existing API Modifications

There are no API Modifications in this release. However, the API documentation has been improved for many of the APIs.

2.6.2 New API Additions

There no API Additions in this release.

2.6.3 Bug Fixes

There are several bug fixes that are included in this release. Some of the functions that have been improved in this release are:

fcvIntegratePatchu8
fcvScaleUpBy2Gaussian5x5u8
fcvFilterCorr3x3s8

All these functions could have performance degradation due to the bug fixes.

Also in this release, some necessary parameter checks were added to most of the APIs. These were required to disallow adverse use-cases that could prevent the library from operating correctly. As a result of these checks, some of the functions that have really short execution times would see a dip in their performance. For functions that have longer execution times, this affect is virtually zero.

The following functions could see some dip in their performance because of these parameter checks.

fcvBitCntu8 (all variants)
fcvHammingDistanceu8 (all variants)
fcvDotProductu8 (most variants)
fcvDotProductNormu8 (most variants)

2.6.4 Performance Improvements

Several functions' performance have been further improved in this release. They include functions in both *FastCV for ARM* and *FastCV for Snapdragon* implementations.

2.6.4.1 *FastCV for ARM*

Performance of these functions in *FastCV for ARM* implementation were further improved in this release.

fcvBoundingRectangle
fcvFilterCanny3x3u8
fcvFilterCanny3x3u8_v2
fcvLinearSearchPrepare8x36s8
fcvSolvef32
fcvSetElementsf32
fcvSetElementsc3s32
fcvSetElementsc3f32
fcvSetElementsc4f32
fcvSegmentFGMasku8

2.6.4.2 *FastCV for Snapdragon*

Performance of these functions in *FastCV for Snapdragon* implementation were further improved in this release.

fcvBoundingRectangle
fcvMseru8
fcvMserExtu8
fcvIDWTHaarTransposes16
fcvSolvef32
fcvFilterCanny3x3u8
fcvFilterCanny3x3u8_v2
fcvScaleDownBy2Gaussian5x5u8_v2
fcvColorYCbCr444PlanarToRGB565u8
fcvIntegrateImageu8
fcvIntegrateImageu8_v2
fcvTrackLKOpticalFlowu8
fcvLinearSearchPrepare8x36s8
fcvAdaptiveThresholdGaussian3x3u8
fcvAdaptiveThresholdGaussian5x5u8
fcvAdaptiveThresholdMean3x3u8
fcvAdaptiveThresholdMean5x5u8
fcvAdaptiveThresholdMean11x11u8
fcvBoxFilter3x3u8
fcvBoxFilter5x5u8
fcvBoxFilter11x11u8
fcvAbsDiffu8

2.6.5 Other changes

In this release, a brand new FastCV Demo App is being introduced. This app has been redesigned to include more demos of FastCV functions than before. This new version contains demos for the following functional areas:

- Fast Corner (from the original demo app)
- Affine
- Correlational Filters
- Scaling
- Image Diff
- Canny Edge Filter
- Image Filters (Sobel, Gaussian, Erode, Dialate, Median, Threshold)
- Image Gradient
- Rotate
- Warp

With the introduction of this new demo app, the original Fast-Corner demo app will be marked for deprecation. That app, though is currently available in this release, will cease to exist in the next release of the library.

2.7 Release 1.2.0

2.7.1 Existing API Modifications

There are no API Modifications in this release. However, `_v<2,3,4...>` versions of the following APIs have been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest `_v<2,3,4...>` version of the specific API.

<code>fcvCornerFast9Scoreu8_v2</code>
<code>fcvCornerFast9InMaskScoreu8_v2</code>
<code>fcvImageGradientSobelPlanars16_v3</code>

2.7.2 New API Additions

The 1.2.0 release of the fastCV library contains a substantial set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (`fastcv.h`) for further information.

<code>fcvIntegrateImageYCbCr420PseudoPlanaru8</code>	<code>fcvCornerFast10u8</code>
<code>fcvFindForegroundIntegrateImageYCbCr420u32</code>	<code>fcvCornerFast10Scoreu8</code>
<code>fcvTransformAffineClippedu8</code>	<code>fcvCornerFast10InMasku8</code>
<code>fcvJacobianSE2f32</code>	<code>fcvCornerFast10InMaskScoreu8</code>
<code>fcvColorRGB565ToYCbCr444Planaru8</code>	<code>fcvFeatureExtractInMask</code>
<code>fcvColorRGB565ToYCbCr422Planaru8</code>	<code>fcvDWTHaaru8</code>
<code>fcvColorRGB565ToYCbCr420Planaru8</code>	<code>fcvDCTu8</code>
<code>fcvColorRGB888ToYCbCr444Planaru8</code>	<code>fcvIDWTHaars16</code>
<code>fcvColorRGB888ToYCbCr422Planaru8</code>	<code>fcvIDCTs16</code>
<code>fcvColorRGB888ToYCbCr420Planaru8</code>	<code>fcvDWT53Tabs16</code>
<code>fcvColorRGBA8888ToYCbCr444Planaru8</code>	<code>fcvIDWT53Tabs16</code>
<code>fcvColorRGBA8888ToYCbCr422Planaru8</code>	<code>fcvAverages32</code>
<code>fcvColorRGBA8888ToYCbCr420Planaru8</code>	<code>fcvAverageu8</code>
<code>fcvColorRGB565ToYCbCr444PseudoPlanaru8</code>	<code>fcvMeanShiftu8</code>
<code>fcvColorRGB565ToYCbCr422PseudoPlanaru8</code>	<code>fcvMeanShifts32</code>
<code>fcvColorRGB565ToYCbCr420PseudoPlanaru8</code>	<code>fcvMeanShiftf32</code>
<code>fcvColorRGB888ToYCbCr444PseudoPlanaru8</code>	<code>fcvCamShiftu8</code>
<code>fcvColorRGB888ToYCbCr422PseudoPlanaru8</code>	<code>fcvCamShifts32</code>
<code>fcvColorRGB888ToYCbCr420PseudoPlanaru8</code>	<code>fcvCamShiftf32</code>
<code>fcvColorRGBA8888ToYCbCr444PseudoPlanaru8</code>	<code>fcvFloodfillSimpleu8</code>
<code>fcvColorRGBA8888ToYCbCr422PseudoPlanaru8</code>	<code>fcvUpdateMotionHistoryu8s32</code>
<code>fcvColorRGBA8888ToYCbCr420PseudoPlanaru8</code>	<code>fcvSVDf32</code>
<code>fcvColorRGB565ToRGB888u8</code>	<code>fcvSolveCholeskyf32</code>
<code>fcvColorRGB565ToRGBA8888u8</code>	<code>fcvFillConvexPolyu8</code>
<code>fcvColorRGB565ToBGR565u8</code>	<code>fcvPointPolygonTest</code>
<code>fcvColorRGB565ToBGR888u8</code>	<code>fcvFindConvexHull</code>
<code>fcvColorRGB565ToBGRA8888u8</code>	<code>fcvBGCodeBookDiffu8</code>
<code>fcvColorRGB888ToRGBA8888u8</code>	<code>fcvBGCodeBookUpdateu8</code>
<code>fcvColorRGB888ToBGR565u8</code>	<code>fcvDrawContouru8</code>
<code>fcvColorRGB888ToBGR888u8</code>	<code>fcvDrawContourInterleavedu8</code>
<code>fcvColorRGB888ToBGRA8888u8</code>	<code>fcvDrawContourPlanaru8</code>
<code>fcvColorRGBA8888ToRGB565u8</code>	<code>fcvHoughCircleu8</code>
<code>fcvColorRGBA8888ToRGB888u8</code>	<code>fcvGeomDistortPoint2x1f32</code>
<code>fcvColorRGBA8888ToBGR565u8</code>	<code>fcvGeomUndistortPoint2x1f32</code>
<code>fcvColorRGBA8888ToBGR888u8</code>	<code>fcvGeomProjectPoint3x1f32</code>
<code>fcvColorRGBA8888ToBGRA8888u8</code>	<code>fcvGeomDistortPoint2xNf32</code>
<code>fcvColorRGBA8888ToLABu8</code>	<code>fcvColorYCbCr444PseudoPlanarToYCbCr420PseudoPlanaru8</code>

fcvColorYCbCr444PlanarToYCbCr422Planaru8	fcvColorYCbCr422PseudoPlanarToYCbCr444PseudoPlanaru8
fcvColorYCbCr444PlanarToYCbCr420Planaru8	fcvColorYCbCr422PseudoPlanarToYCbCr420PseudoPlanaru8
fcvColorYCbCr422PlanarToYCbCr444Planaru8	fcvColorYCbCr420PseudoPlanarToYCbCr444PseudoPlanaru8
fcvColorYCbCr422PlanarToYCbCr420Planaru8	fcvColorYCbCr420PseudoPlanarToYCbCr422PseudoPlanaru8
fcvColorYCbCr420PlanarToYCbCr444Planaru8	fcvColorYCbCr444PseudoPlanarToYCbCr444Planaru8
fcvColorYCbCr420PlanarToYCbCr422Planaru8	fcvColorYCbCr444PseudoPlanarToYCbCr422Planaru8
fcvColorYCbCr444PlanarToYCbCr444PseudoPlanaru8	fcvColorYCbCr444PseudoPlanarToYCbCr420Planaru8
fcvColorYCbCr444PlanarToYCbCr422PseudoPlanaru8	fcvColorYCbCr422PseudoPlanarToYCbCr444Planaru8
fcvColorYCbCr444PlanarToYCbCr420PseudoPlanaru8	fcvColorYCbCr422PseudoPlanarToYCbCr422Planaru8
fcvColorYCbCr422PlanarToYCbCr444PseudoPlanaru8	fcvColorYCbCr422PseudoPlanarToYCbCr420Planaru8
fcvColorYCbCr422PlanarToYCbCr422PseudoPlanaru8	fcvColorYCbCr420PseudoPlanarToYCbCr444Planaru8
fcvColorYCbCr422PlanarToYCbCr420PseudoPlanaru8	fcvColorYCbCr420PseudoPlanarToYCbCr422Planaru8
fcvColorYCbCr420PlanarToYCbCr444PseudoPlanaru8	fcvColorYCbCr420PseudoPlanarToYCbCr420Planaru8
fcvColorYCbCr420PlanarToYCbCr422PseudoPlanaru8	fcvGeomUndistortPoint2xNf32
fcvColorYCbCr420PlanarToYCbCr420PseudoPlanaru8	fcvGeomProjectPoint3xNf32
fcvColorYCbCr444PseudoPlanarToYCbCr422PseudoPlanaru8	fcvRemapRGBA8888Nnu8
	fcvRemapRGBA8888BLu8

2.7.3 Bug Fixes

Some of the bug fixes for this release are –

1. Fix bit exactness issue at the first pixel for DCT API
2. Clean up fcvDescriptorSampledMeanAndVar36f32 and fcvImageIntensityStats
3. Add stride support to fcvScaleDownu8C
4. Fix status at LK optical flow
5. Fixed dstStride related bug for fcvImageDiffu8f32
6. Fixed fcvPyramidDelete – removed implicit ‘delete()’ of ‘pyr’ pointer parameter

2.7.4 Performance Improvements

2.7.4.1 FastCV

Cannot characterize the improvements for the newly added 1.2.0 functions as there is no reference to which to compare the functions’ improvements. Following functions have undergone performance improvements when compared to 1.1.1 release

fcv3ChannelWarpPerspectiveu8	fcv3ChannelWarpPerspectiveu8_v2
fcvFilterCorrSep11x11s16	fcvFilterCorrSep13x13s16
fcvFilterCorrSep17x17s16	fcvScaleDownu8
fcvScaleDownu8_v2	fcvMserExt
fcvTrackBMOpticalFlow16x16u8	fcvMser

2.7.4.2 FastCV for Snapdragon

Most of the newly added 1.2.0 functions have been accelerated in Qualcomm Snapdragon platform. Following functions have undergone performance improvements when compared to 1.1.1 release.

fcvScaleDownu8_v2	fcvScaleDownu8
fcvMserExt	fcvMser

2.7.5 Other Changes

Consistency has been maintained in keeping alignment checks (with respect to address, width and stride) for all new functions added or modified.

2.8 Release 1.2.1

2.8.1 Existing API Modifications

There is no API Modifications in this release. However, the API documentation has been improved for many of the APIs.

2.8.2 New API Additions

There is no API Additions in this release.

2.8.3 Bug Fixes

There are several bug fixes that are included in this release.

2.8.3.1 *FastCV for ARM*

Some of the bug fixes in *FastCV for ARM* implementations are

1. Fix precision issue at `fcvGetPerspectiveTransformf32`
2. Fix precision issue at `fcvGeomHomographyFitf32`
3. Remove white border at `fcvFilterDilate3x3` and `fcvFilterDilate3x3_v2`
4. Remove white border at `fcvFilterErode3x3` and `fcvFilterErode3x3_v2`
5. Fix out of bound memory access at `fcvScaleDownBy2Gaussian5x5u8_v2`
6. Fix incorrect result at `fcvFloodfillSimpleu8`

These functions may show limited performance degradation due to the bug fixes.

2.8.3.2 *FastCV for Snapdragon*

Some of the bug fixes in *FastCV for Snapdragon* implementations are

1. Fix precision issue at `fcvGetPerspectiveTransformf32`
2. Fix precision issue at `fcvGeomHomographyFitf32`
3. Remove white border at `fcvFilterDilate3x3` and `fcvFilterDilate3x3_v2`
4. Remove white border at `fcvFilterErode3x3` and `fcvFilterErode3x3_v2`
5. Fix out of bound memory access at `fcvScaleDownBy2Gaussian5x5u8` and `fcvScaleDownBy2Gaussian5x5u8_v2`
6. Fix incorrect result at `fcvFloodfillSimpleu8`
7. Fix bit exactness at `fcvHoughCircleu8`
8. Fix corruption at `fcvImageGradientSobelPlanars16_v3`
9. Fix corruption at `fcvFilterCanny3x3u8` and `fcvFilterCanny3x3u8_v2`
10. Fix memory corruption at `fcvCornerHarrisu8` and variant

These functions may show limited performance degradation due to the bug fixes.

2.8.4 Performance Improvements

Several functions' performance have been further improved in this release. They include functions in both *FastCV for ARM* and *FastCV for Snapdragon* implementations.

2.8.4.1 *FastCV for ARM*

Performance of these functions in *FastCV for ARM* implementation were further improved in this release.

fcvMseru8
fcvMserExtu8

2.8.4.2 *FastCV for Snapdragon*

Performance of these functions in *FastCV for Snapdragon* implementation were further improved in this release.

fcvMseru8
fcvMserExtu8
fcvSVDf32

2.8.5 Other changes

Version checking has been added to make sure fastcv.h and fastcv.inl are used from the same fastcv version.

2.9 Release 1.2.2

2.9.1 Existing API Modifications

There is no API Modification in this release.

2.9.2 New API Additions

There is no API Addition in this release.

2.9.3 Compiler Change

Different compiler is used to build Fastcv1.2.2 library. Prior to FastCV1.2.2, the library is built using GCC compiler.

FastCV 1.2.2 is built using LLVM 3.2.5 64 bit toolchain compiler to improve majority API's performance. However, a few of APIs' performances are degraded.

2.9.4 Bug Fixes

There are several bug fixes that are included in this release.

This release fixed remote library mechanism.

Prior to FastCV1.2.2, if the device has dynamic library (libfastcvopt.so) at the system, the *FastCV for ARM* or *FastCV for Snapdragon* implementation will be taken from dynamic library, regardless the application was compiled using static library.

2.9.4.1 *FastCV for ARM*

Some of the bug fixes in *FastCV for ARM* implementations are

1. Fix precision issue at `fcvImageIntensityStats`
2. Fix at `fcvClusterEuclideanf32` and `fcvClusterEuclideanNormedf32`
3. Fix rounding at `fcvScaleDownMNU8` and `fcvScaleDownMNInterleaveu8`
4. Fix crash issue at `fcvTransfromAffineClippedu8`
5. Fix stride issue at `fcvFilterGaussian5x5s16` and `fcvFilterGaussian5x5s32`

These functions may show limited performance degradation due to the bug fixes.

2.9.4.2 *FastCV for Snapdragon*

Some of the bug fixes in *FastCV for Snapdragon* implementations are

1. Add special handling at `fcvSVDf32` that is being used by `fcvGetPerspectiveTransformf32`
2. Fix bug at `fcvTrackLKOpticalFlowu8` and `fcvTrackLKOpticalFlowu8`
3. Fix border at `fcvFilterCorrSepNxNs16` variances
4. Fix crash at `fcvTransformAffine8x8u8` for multi thread use.
5. Fix precision issue at `fcvImageIntensityStats`
6. Fix at `fcvClusterEuclideanf32` and `fcvClusterEuclideanNormedf32`
7. Fix bug at `fcvFloodFillSimpleu8`
8. Fix crash issue at `fcvTransfromAffineClippedu8`
9. Fix stride issue at `fcvFilterGaussian5x5s16` and `fcvFilterGaussian5x5s32`
10. Fix out of bound memory access at `fcvFilterDilate3x3u8` and `fcvFilterErode3x3u8`
11. Fix out of bound memory access at `fcvWarpPerspectiveu8`, `fcv3ChannelWarpPerspectiveu8` and variances
12. Fix out of bound memory access at `fcvScaleUpBy2Gaussian5x5u8_v2` and `fcvScaleDownBy2Gaussian5x5f32`
13. Fix out of bound memory access at `fcvFilterMedian3x3u8_v2`

These functions may show limited performance degradation due to the bug fixes.

2.9.5 Performance Improvements

In general, majority functions' performances have been further improved in this release. They include functions in both *FastCV for ARM* and *FastCV for Snapdragon* implementations.

2.9.5.1 FastCV for ARM

Performance of these functions in *FastCV for ARM* implementation were further improved in this release.

fcvAbsDiffs32	fcvFillConvexPolyu8
fcvAbsDiffVc3u8	fcvFilterCorr3x3s8
fcvAbsDiffVc4u8	fcvFilterCorr3x3s8_v2
fcvAbsDiffVf32	fcvFilterDilate3x3
fcvAbsDiffVs32	fcvFilterDilate3x3_v2
fcvAbsDiffVu8	fcvFilterErode3x3
fcvAdaptiveThresholdGaussian11x11u8	fcvFilterErode3x3_v2
fcvAdaptiveThresholdMean11x11u8	fcvFilterGaussian11x11u8
fcvAverageu8	fcvFilterGaussian11x11u8_v2
fcvColorRGB565ToYCbCr420Planaru8	fcvFilterGaussian5x5s16
fcvColorRGB565ToYCbCr420PseudoPlanaru8	fcvFilterGaussian5x5s16
fcvColorRGB565ToYCbCr422Planaru8	fcvFilterGaussian5x5s16_v2
fcvColorRGB565ToYCbCr422PseudoPlanaru8	fcvFilterGaussian5x5s16_v2
fcvColorRGB888ToYCbCr420Planaru8	fcvFilterSobel3x3u8
fcvColorRGB888ToYCbCr420PseudoPlanaru8	fcvFilterSobel3x3u8_v2
fcvColorRGB888ToYCbCr422Planaru8	fcvFindContoursExternalu8
fcvColorRGB888ToYCbCr422PseudoPlanaru8	fcvGeomAffinefitf32
fcvColorRGB888ToYCbCr444Planaru8	fcvImageDifff32
fcvColorRGB888ToYCbCr444PseudoPlanaru8	fcvImageDiffu8
fcvColorRGBA8888ToLABu8	fcvImageDiffu8_v2
fcvColorRGBA8888ToYCbCr420Planaru8	fcvImageDiffu8s8
fcvColorRGBA8888ToYCbCr420PseudoPlanaru8	fcvImageGradientSobelPlanars16
fcvColorRGBA8888ToYCbCr422Planaru8	fcvImageGradientSobelPlanars16_v2
fcvColorRGBA8888ToYCbCr422PseudoPlanaru8	fcvImageGradientSobelPlanars16_v3
fcvColorRGBA8888ToYCbCr444Planaru8	fcvMeanShiftf32
fcvColorRGBA8888ToYCbCr444PseudoPlanaru8	fcvScaleDownMNInterleaveu8
fcvColorYCrCbH1V2toRGB888u8	fcvScaleDownMNU8
fcvColorYCrCbH2V1toRGB888u8	fcvScaleDownu8
fcvColorYCrCbH2V2toRGB888u8	fcvScaleDownu8_v2
fcvConAdaTrackf32	fcvScaleUpPolyInterleaveu8
fcvDescriptor17x17u8To36s8	fcvSetElementsf32
fcvDescriptorSampledMeanAndVar36f32	fcvSetElementsu8
fcvDotProduct11x12u8	fcvSumOfAbsoluteDiffs16x16u8
fcvEdgeWeightings16	fcvTrackBMOpticalFlow16x16u8

2.9.5.2 FastCV for Snapdragon

Performance of these functions in *FastCV for Snapdragon* implementation were further improved in this release.

fcvAbsDiffVc3f32	fcvFilterCorrSep9x9s16
fcvAbsDiffVc3s32	fcvFilterCorrSep9x9s16_v2
fcvAbsDiffVc3u8	fcvFilterGaussian11x11u8
fcvAbsDiffVf32	fcvFilterGaussian11x11u8_v2
fcvAbsDiffVs32	fcvFilterGaussian5x5u8
fcvAdaptiveThresholdGaussian11x11u8	fcvFilterGaussian5x5u8_v2
fcvAdaptiveThresholdGaussian5x5u8	fcvGeomPoseEvaluatef32
fcvClusterEuclideanf32	fcvImageGradientInterleavedf32
fcvClusterEuclideanNormed36f32	fcvImageGradientInterleavedf32_v2
fcvColorRGB888toYCrCb8u8	fcvImageIntensityStats
fcvColorRGB888toYCrCb8_u2	fcvPyramidCreateu8
fcvColorYCbCr420PlanarToRGB565u8	fcvRemapRGBA8888NNu8
fcvColorYCbCr420PseudoPlanarToRGB565u8	fcvScaleDownBy2Gaussian5x5u8
fcvColorYCbCr422PlanarToRGB565u8	fcvScaleDownBy2Gaussian5x5u8_v2
fcvColorYCbCr422PseudoPlanarToRGB565u8	fcvScaleDownBy2u8
fcvColorYCbCr444PseudoPlanarToRGB565u8	fcvScaleDownBy2u8_v2
fcvColorYCrCb420PseudoPlanarToRGB888u8	fcvScaleDownBy4u8_v2
fcvColorYUV420toRGB565u8	fcvScaleUpBy2Gaussian5x5u8
fcvColorYUV420toRGB888u8	fcvScaleUpBy2Gaussian5x5u8_v2
fcvDescriptorSampledMeanAndVar36f32	fcvSetElementss32 with mask
fcvDotProduct11x12u8	fcvSumOfAbsoluteDiffs16x16u8
fcvFilterCorrSep13x13s16	fcvSumOfAbsoluteDiffs8x8u8
fcvFilterCorrSep13x13s16_v2	fcvSumOfAbsoluteDiffs8x8u8_v2
fcvFilterCorrSep15x15s16	fcvSVDf32
fcvFilterCorrSep15x15s16_v2	fcvTrackBMOpticalFlow16x16u8
fcvFilterCorrSep17x17s16	fcvTrackLKOpticalFlowu8
fcvFilterCorrSep17x17s16_v2	fcvVecNormalize36s8f32

2.9.6 Other changes

FastCV 1.2.2 has the capability to select the QDSP or GPU implementation when using QDSP and GPU supported Qualcomm device.

FastCV 1.2.2 will automatic select the best implementation for each API based on the operation mode chosen.

Operation mode	Description
FASTCV_OP_LOW_POWER	The QDSP implementation will be used unless the QDSP speed is 3 times slower than CPU speed.
FASTCV_OP_PERFORMANCE	The fastest implementation will be used.
FASTCV_OP_CPU_OFFLOAD	The QDSP implementation will be used when it's available, otherwise it will find for GPU and CPU implementation.

2.10 Release 1.3.0

2.10.1 Existing API Modifications

There are no API Modifications in this release. However, _v2 versions of the following APIs have been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest _v2 version of the specific API.

fcvGeomHomographyEvaluatef32_v2
fcvMserExtu8_v2
fcvMseru8_v2
fcvPyramidCreatef32_v2
fcvPyramidCreateu8_v2
fcvTrackLKOpticalFlowu8_v2

2.10.2 New API Additions

The 1.3.0 release of the fastCV library contains a substantial set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcv3ChannelTransformAffineClippedBCu8	fcvImageHistogramEqualizeu8
fcvAddf32	fcvImageMomentsf32
fcvAddu16	fcvImageMomentsu8
fcvAddu8u16	fcvImageSegmentationRegionGrow
fcvBilateralFilter5x5u8	fcvImageSpatialHistogramu8
fcvBilateralFilter7x7u8	fcvIntegrateImageu8u64
fcvBilateralFilter9x9u8	fcvMatrixMultiplyf32
fcvBlockDotProductf32	fcvMatrixMultiplies8s32
fcvBlockDotProductu8	fcvMserExtNN8u8
fcvClusterEuclideanu8	fcvMserNN8Init
fcvColorCbCrSwapu8	fcvMserNN8u8
fcvConfigBGCodeBookModel	fcvScaleDownBLu8
fcvDotProductf32	fcvSolveLDLf32
fcvElementMultiplyf32	fcvSumOfSquaredDiffsf32
fcvElementMultiplyu8u16	fcvSumOfSquaredDiffss8
fcvFilterThresholdOtsuu8	fcvSumOfSquaredDiffsu8
fcvFindConvexHull	fcvTransposef32
fcvFlipu16	fcvTransposeu16
fcvFlipu8	fcvTransposeu8
fcvHoughLineu8	
fcvImageDetectEdgePixelu8	

2.10.3 Bug Fixes

There are several bug fixes that are included in this release.

1. Update `fcvSolveCholeskyf32` to detect more unsolved scenario.
2. Fix precision issue at `fcvGeomHomographyFitf32`
3. Fix rounding calculation and memory leak at `fcvTrackLKOpticalFlowu8` and `fcvTrackLKOpticalFlowf32`
4. Fix saturation problem at `fcvScaleDownMNInterleaveu8`

These functions may show limited performance degradation due to the bug fixes.

2.10.3.1 FastCV for Snapdragon

The bug fix in *FastCV for Snapdragon* implementations is

- Fix out of bound memory access at `fcvBitwiseOrs32`

2.10.4 Performance Improvements

Most of the newly added 1.3.0 functions have been accelerated in Qualcomm Snapdragon platform.

2.10.5 Other Changes

Consistency has been maintained in keeping alignment checks (with respect to address, width and stride) for all new functions added or modified.

2.10.5.1 `fcvMemAlloc` and `fcvMemFree`

An automatic mechanism to choose type of memory is added at `fcvMemAlloc` and `fcvMemFree`.

Depending on the device and operation mode, the memory can be allocated from a shared memory that will improve FastCV QDSP and GPU implementation.

`fcvMemAlloc` and `fcvMemFree` may show some performance degradation.

2.11 Release 1.4.0

2.11.1 Existing API Modifications

There are new feature for below existing API to support in-place implementation.

- `fcvFlipu8`
- `fcvFlipu16`

There are slight API modifications for these 2 APIs

Previous API:

```

FASTCV_API void
fcvFlipu8(  const uint8_t * restrict  src,
            uint32_t                srcWidth,
            uint32_t                srcHeight,
            uint32_t                srcStride,
            uint8_t * restrict      dst,
            uint32_t                dstStride,
            fcvFlipDir              dir );

```

```

FASTCV_API void
fcvFlipu16( const uint16_t * restrict  src,
            uint32_t                srcWidth,
            uint32_t                srcHeight,
            uint32_t                srcStride,
            uint16_t * restrict      dst,
            uint32_t                dstStride,
            fcvFlipDir              dir );

```

New API:

```

FASTCV_API void
fcvFlipu8(  const uint8_t *          src,
            uint32_t                srcWidth,
            uint32_t                srcHeight,
            uint32_t                srcStride,
            uint8_t *                dst,
            uint32_t                dstStride,
            fcvFlipDir              dir );

```

```

FASTCV_API void
fcvFlipu16( const uint16_t * src,
            uint32_t          srcWidth,
            uint32_t          srcHeight,
            uint32_t          srcStride,
            uint16_t *        dst,
            uint32_t          dstStride,
            fcvFlipDir        dir );

```

_v3 version of the following APIs has been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest _v3 version of the specific API.

fcvFilterCanny3x3u8_v3
fcvMserExtu8_v3

2.11.2 New API Additions

The 1.4.0 release of the fastCV library contains a set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcvFilterCorrSepNxNs16	fcvCornerHarrisAdaptiveu8
------------------------	---------------------------

2.11.3 Bug Fixes

There are several bug fixes that are included in this release.

1. Fix width issue at fcvFilterCanny3x3u8_v2.
2. Fix crash issue at fcvScaleUpPolyu8.
3. Fix possible out of bound memory access at fcvScaleDownBLu8.
4. Reduce over segmentation at fcvImageSegmentationRegionGrow.
5. Fix possible out of bound memory access at fcvTrackBMOpticalFlow16x16u8

These functions may show limited performance degradation due to the bug fixes.

2.11.3.1 FastCV for Snapdragon

The bug fix in *FastCV for Snapdragon* implementations is

- Update fcvIntegrateImageu8u64 for large input images.

2.11.4 Performance Improvements

All newly added 1.4.0 functions have been accelerated in Qualcomm Snapdragon platform.

Several existing functions' performances have been further improved in this release. They include functions in both *FastCV for ARM* and *FastCV for Snapdragon* implementations.

fcvImageHistogramEqualizeu8	fcvScaleDownBLu8
fcvAdaptiveThresholdGaussian3x3u8	fcvAdaptiveThresholdMean3x3u8
fcvAdaptiveThresholdGaussian5x5u8	fcvAdaptiveThresholdMean5x5u8
fcvAdaptiveThresholdGaussian11x11u8	fcvAdaptiveThresholdMean11x11u8

2.11.4.1 FastCV for Snapdragon

Performance of these functions in *FastCV for Snapdragon* implementation were further improved in this release.

fcvBilateralFilter5x5u8	fcvFindContours (variants)
fcvBilateralFilter7x7u8	fcvScaleDownu8_v2
fcvBilateralFilter9x9u8	fcvSetElementsf32
fcvFilterGaussian5x5u8	fcvSetElementss32
fcvFilterGaussian11x11u8	fcvConvValids16

2.11.5 Other Changes

Consistency has been maintained in keeping alignment checks (with respect to address, width and stride) for all new functions added or modified.

2.11.5.1 *fcvMemInit and fcvMemDeInit*

A scratch memory management that is used by FastCV sub-system. The sub-system handles scratch memory requirements from several FastCV functions that do not have parameters supplied by the users of those functions.

Calling `fcvMemInit` once before calling any `fastcv` API will give some performance increase.

`fcvMemDeinit` must be called at the end of user's application to avoid memory leak.

2.11.5.2 *New fcvOperationMode option*

`FASTCV_OP_CPU_PERFORMANCE` has been added in this release. This operation mode will select CPU highest performance implementation.

Below are the all available options

Operation mode	Description
<code>FASTCV_OP_LOW_POWER</code>	The QDSP implementation will be used unless the QDSP speed is 3 times slower than CPU speed.
<code>FASTCV_OP_PERFORMANCE</code>	The fastest implementation will be used.
<code>FASTCV_OP_CPU_OFFLOAD</code>	The QDSP implementation will be used when it's available, otherwise it will find for GPU and CPU implementation.
<code>FASTCV_OP_CPU_PERFORMANCE</code>	The CPU fastest implementation will be used.

2.12 Release 1.5.0

2.12.1 Existing API Modifications

There are slight API modifications for these 2 APIs to support element square calculation from the same source (`src1` equal to `src2`)

- `fcvElementMultiplyu8u16`
- `fcvElementMultiplyf32`

Previous API:

```
FASTCV_API void
fcvElementMultiplyu8u16( const uint8_t * __restrict  src1,
                        uint32_t                width,
                        uint32_t                height,
                        uint32_t                src1Stride,
```

```

        const uint8_t * __restrict src2,
        uint32_t src2Stride,
        uint16_t * __restrict dst,
        uint32_t dstStride );

```

```

FASTCV_API void
fcvElementMultiplyf32( const float32_t * __restrict src1,
        uint32_t width,
        uint32_t height,
        uint32_t src1Stride,
        const float32_t * __restrict src2,
        uint32_t src2Stride,
        float32_t * __restrict dst,
        uint32_t dstStride );

```

New API:

```

FASTCV_API void
fcvElementMultipliyu8u16( const uint8_t * src1,
        uint32_t width,
        uint32_t height,
        uint32_t src1Stride,
        const uint8_t * src2,
        uint32_t src2Stride,
        uint16_t * __restrict dst,
        uint32_t dstStride );

```

```

FASTCV_API void
fcvElementMultiplyf32( const float32_t * src1,
        uint32_t width,
        uint32_t height,
        uint32_t src1Stride,
        const float32_t * src2,
        uint32_t src2Stride,
        float32_t * __restrict dst,
        uint32_t dstStride );

```

A newer version of the following APIs has been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest version of the specific API.

fcvAdds16_v2	fcvIntegratePatchu8_v3
fcvBilateralFilter5x5u8_v2	fcvTransformAffineClippedu8_v2
fcvBilateralFilter7x7u8_v2	fcvWarpPerspectiveu8_v3
fcvBilateralFilter9x9u8_v2	

2.12.2 New API Additions

The 1.5.0 release of the fastCV library contains a set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcvAddScalarf32	fcvFilterSobel3x3u8s16
fcvAddScalars16	fcvFilterSobel5x5u8s16
fcvAddSquaredu8u16	fcvFilterSobel7x7u8s16
fcvAddu16u8u16	fcvFilterThresholdRangeu8
fcvAddu8	fcvFlipRGB888u8
fcvAddWeightedu8	fcvFloodfillMergedu8
fcvBilateralFilterRecursiveu8	fcvGeomHomographyRobustFitf32
fcvBitwiseAndu8	fcvGLBPu8
fcvBitwiseNotu8	fcvGoodFeatureToTracku8
fcvBitwiseXoru8	fcvIFFTf32
fcvBoxFilterNxNf32	fcvImageSegmentationSeedRegionGrows16
fcvChannelCombine2Planesu8	fcvInvertAffineTransform
fcvChannelCombine3Planesu8	fcvMagnitudes16
fcvChannelCombine4Planesu8	fcvMinMaxLocf32
fcvChannelExtractu8	fcvMinMaxLocs16
fcvColorRGB888ToHSV888u8	fcvMinMaxLocs32
fcvConvertDepths16u8	fcvMinMaxLocu16
fcvConvertDepthu8s16	fcvMinMaxLocu32
fcvCornerRefineSubPixu8	fcvMinMaxLocu8
fcvCrossProduct3x1f32	fcvMultiplyScalarf32
fcvElementMultiplies16	fcvMultiplyScalars16
fcvElementMultiplu8	fcvNCCPatchesOnRectu8
fcvElementMultiplu8s16	fcvNormalizeLocalBoxf32
fcvFFTu8	fcvNormalizeLocalBoxu8
fcvFilterCannyu8	fcvPhases16
fcvFilterConvolveMxNu8	fcvRemapu8
fcvFilterConvolveMxNu8s16	fcvScaleu8
fcvFilterCorrNxNu8	fcvSolveLUf32
fcvFilterCorrNxNu8f32	fcvSubtracts16
fcvFilterCorrNxNu8s16	fcvSubtractu8
fcvFilterCorrSepMxNu8	fcvSubtractu8s16
fcvFilterDilateNxNu8	fcvSVMPredict2Classf32
fcvFilterErodeNxNu8	fcvTableLookupu8
fcvFilterMedianMxNs16	fcvTransposeRGB888u8

2.12.3 Bug Fixes

There are several bug fixes that are included in this release.

1. Fix stride issue (change from in pixels to in bytes) at fcvDrawContourInterleavedu8.
2. Fix odd height issue at fcvFlipu8 and fcvFlipu16 in place
3. Fix rounding precision issue at fcvTrackLKOpticalFlowu8

These functions may show limited performance degradation due to the bug fixes.

2.12.3.1 FastCV for ARM

Some of the bug fixes in *FastCV for ARM* implementations are

1. Fix segmentation fault issue with small image at fcvScaleUpBy2Gaussian5x5u8
2. Added [0,255] clipping check for barrier at fcvCornerFast9u8 and fcvCornerFast9InMasku8

2.12.3.2 FastCV for Snapdragon

The bug fix in *FastCV for Snapdragon* implementations is

- Fix out of bound memory access for width<8 case at fcvScaleDownMNU8

2.12.4 Performance Improvements

All newly added functions have been accelerated in Qualcomm Snapdragon platform. Several existing functions' performances have been further improved in this release.

2.12.4.1 FastCV for Snapdragon

Performance of these functions in *FastCV for Snapdragon* implementation were further improved in this release.

fcvBitCount32x1u8	fcvHammingDistance64x1u8a4
fcvBitCount32x4u8	fcvHammingDistance64x4u8
fcvBitCount64x1u8	fcvHammingDistance64x4u8a4
fcvBitCount64x4u8	fcvHammingDistanceu8
fcvBitCountu32	fcvImageGradientSobelPlanars16
fcvBitCountu8	fcvImageGradientSobelPlanars16_v2
fcvBitwiseOrs32	fcvImageGradientSobelPlanars16_v3
fcvBitwiseOru8	fcvImageIntensityStats
fcvBoxFilter3x3u8	fcvRemapRGBA8888NNU8
fcvBoxFilter5x5u8	fcvScaleDownNNU8
fcvColorYCrCbH1V1toRGB888u8	fcvSort8Scoresf32
fcvColorYCrCbH2V1toRGB888u8	fcvSumOfAbsoluteDiff16x16u8
fcvColorYCrCbH2V2toRGB888u8	fcvSumOfAbsoluteDiff8x8u8
fcvHammingDistance32x1u8	fcvSumOfAbsoluteDiffs16x16u8
fcvHammingDistance32x1u8a4	fcvSumOfAbsoluteDiffs8x8u8_v2

fcvHammingDistance32x4u8a4	fcvSumOfSquaredDiffs36xNs8
fcvHammingDistance64x1u8	

2.12.5 Other Changes

2.12.5.1 *fcvStatus* return type

All new API addition at FastCV 1.5.0 has *fcvStatus* as the return type. Please see individual *fcvStatus* documentation in header file (*fastcv.h*) for further information.

2.12.5.2 *Inline file (fastcv.inl) removal*

New architecture for mapping each API with CPU or HW implementation was introduced. With the new architecture, *fastcv.inl* is not required anymore and was removed from FastCV1.5.0.

2.12.5.3 *Friendlier hardware limitation*

The alignment and width requirements described at *fastcv.inl* (e.g. 128 bit alignment of the source/destination pointer or multiple of 8-stride) are still required to run the HW implementation. However, if such requirements couldn't be satisfied, FastCV library will automatically choose the best CPU alternative solution.

2.12.5.4 *Run time FastCV android log option*

Using Android system property, a new option was added to enable or disable additional FastCV library log. Once it's enabled, *fastcv* library logs should be appeared at the Android logcat.

2.12.5.4.1 *To enable the log*

Before running the application, a system property on the android phone should be set by calling "adb shell setprop runtime.fastcv.log 1"

2.12.5.4.2 *To disable the log*

Before running the application, a system property on the android phone should be set by calling "adb shell setprop runtime.fastcv.log 0"

2.13 Release 1.6.0

2.13.1 Existing API Modifications

There are slight API modifications for *fcvBoxFilterNxNf3* API to support in place implementation and even kernel.

Previous API:

```
FASTCV_API void
fcvBoxFilterNxNf32(const float32_t* __restrict src,
                  uint32_t srcWidth,
```

```

uint32_t      srcHeight,
uint32_t      srcStride,
uint32_t      N,
float32_t*    __restrict dst,
uint32_t      dstStride);

```

New API:

```

FASTCV_API void
fcvBoxFilterNxNf32(const float32_t* src,
                  uint32_t      srcWidth,
                  uint32_t      srcHeight,
                  uint32_t      srcStride,
                  uint32_t      N,
                  float32_t*    dst,
                  uint32_t      dstStride);

```

A newer version of the following APIs has been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest version of the specific API.

fcvBoxFilter3x3u8_v2	fcvFilterThresholdRangeu8_v2
fcvFilterDilate3x3u8_v3	fcvFilterThresholdu8_v3
fcvFilterErode3x3u8_v3	fcvGeomHomographyFitf32_v2
fcvFilterGaussian3x3u8_v3	fcvGeomHomographyRobustFitf32_v2
fcvFilterMedian3x3u8_v3	fcvImageSpatialHistogramu8_v2

2.13.2 New API Additions

The 1.6.0 release of the fastCV library contains a set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcv2PlaneWarpPerspectiveu8	fcvICPJacobianErrorSE3f32
fcvDepthFusion8x8x8xNs16	fcvImageDetectLineSegmentsu8
fcvFindMultipleMaximau8	fcvScaleDownBy2Gaussian3x3u8

2.13.3 Bug Fixes

There are several bug fixes that are included in this release.

- More precision at fcvGeomHomographyRobustFitf32

2.13.3.1 FastCV for Snapdragon

The bug fix in *FastCV for Snapdragon* implementations is

- Fix out of bound memory access for odd width case at `fcvImageGradientSobel` variants
- Fix out of bound memory access for specific odd width case at `fcvTableLookupu8`
- Fix out of bound memory access at `fcvFilterCorrNxNu8f32`
- Fix crash issue at `fcvNCCPatchesOnRectu8`
- Fix crash issue at several `fcvCornerFast` APIs for the case: width or height less than $2 * \text{border}$

2.13.4 Performance Improvements

All newly added functions have been accelerated in Qualcomm Snapdragon platform.

2.14 Release 1.7.0

2.14.1 Existing API Modifications

There are new feature for below existing API to support in-place implementation.

fcvFilterThresholdu8	fcvFilterThresholdOtsuu8
fcvFilterThresholdu8_v2	fcvFilterThresholdRangeu8
fcvFilterThresholdu8_v3	fcvFilterThresholdRangeu8_v2

A newer version of the following APIs has been added, following the same strategy as mentioned in [2.5.1](#). Developers are encouraged to move to the latest version of the specific API.

fcvImageIntensityStats_v2	fcvPyramidAllocate_v3
fcvMinMaxLocf32_v2	fcvPyramidCreateu8_v3
fcvMinMaxLocs16_v2	fcvRemapu8_v2
fcvMinMaxLocs32_v2	fcvScaleu8_v2
fcvMinMaxLocu16_v2	fcvTrackLKOpticalFlowu8_v3
fcvMinMaxLocu32_v2	fcvTransformAffineClippedu8_v3
fcvMinMaxLocu8_v2	fcvWarpPerspectiveu8_v4

2.14.2 New API Additions

The 1.7.0 release of the fastCV library contains a set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcvCornerHarrisScoreu8	fcvRotateImageInterleavedu8
fcvConvertDepthImageToPointCloudf32	fcvRotateImageu8
fcvRegisterDepthImagef32	fcvUndistortDisparityConvertDepthf32

2.14.3 Bug Fixes

There are several bug fixes that are included in this release.

- More optimization at fcvImageDetectLineSegmentsu8
- Support multiple (smaller) dimension without requiring to reinitialize at fcvMser variants
- Fixed incorrect result and more optimization at fcvFilterDilateNxNu8
- Fixed incorrect result and more optimization at fcvFilterErodeNxNu8
- Fixed incorrect result at fcvNCCPatchesOnRectu8
- Fixed incorrect result at fcvFilterCanny3x3u8_v2

- Fixed incorrect result at fcvFilterCanny3x3u8_v3
- Fixed incorrect result at fcvChannelExtractu8 with FASTCV_NV12 and FASTCV_NV21
- Fixed incorrect result at fcvMagnitudes16
- Fixed incorrect result at fcvFilterSobel3x3u8s16 border case
- Fixed fcvFilterMedian3x3u8_v3 for constant border and/or srcStride != dstStride cases

Some of these functions may show limited performance degradation due to the bug fixes.

2.14.4 Performance Improvements

All newly added functions have been accelerated in Qualcomm Snapdragon platform, except fcvMinMaxLoc_v2 variants.

2.14.5 Other Changes

2.14.5.1 Support both 32 bit and 64 bit library for Android

Starting from FastCV1.7.0, both 32 bit and 64 bit library for Android are supported.

2.14.5.2 Compiler

LLVM compiler is used to build both 32 bit and 64 bit library.

LLVM 3.6 toolchain is used to build FastCV 1.7.0 32 bit and 64 bit library.

2.15 Release 1.7.1

2.15.1 New API Additions

The 1.7.1 release of the fastCV library contains a set of new functions. The following table identifies the new functions which are added to the fastCV library. Please see individual function documentation in header file (fastcv.h) for further information.

fcvExtractHOGu16	fcvHOGInit
fcvHOGDelnit	

2.15.2 Bug Fixes

There are several bug fixes that are included in this release.

- Fixed memory allocation at fcvCornerFast10InMasku8
- Fixed incorrect result at fcvGeomAffinefitf32
- Fixed incorrect result at fcvFilterMedian3x3u8_v2, fcvFilterMedian3x3u8_v3

Some of these functions may show limited performance degradation due to the bug fixes.

2.15.3 Performance Improvements

All newly added functions have been accelerated in Qualcomm Snapdragon

platform Other Changes

2.15.3.1 Support both 32 bit and 64 bit library for Android

Starting from FastCV1.7.0, both 32 bit and 64 bit library for Android are supported.

2.15.3.2 Compiler

LLVM compiler is used to build both 32 bit and 64 bit library.

LLVM 3.7 toolchain is used to build FastCV 1.7.1 32 bit and 64 bit library.