Advanced Micro Devices

Advanced Media Framework – HQ Scaler

Programming Guide

Disclaimer

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information.

Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD’s products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD, the AMD Arrow logo, ATI Radeon™, CrossFireX™, LiquidVR™, TrueAudio™ and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Windows™, Visual Studio and DirectX are trademark of Microsoft Corp.

Copyright Notice

Notice Regarding Standards. AMD does not provide a license or sublicense to any Intellectual Property Rights relating to any standards, including but not limited to any audio and/or video codec technologies such as MPEG-2, MPEG-4; AVC/H.264; HEVC/H.265; AAC decode/FFMPEG; AAC encode/FFMPEG; VC-1; and MP3 (collectively, the “Media Technologies”). For clarity, you will pay any royalties due for such third party technologies, which may include the Media Technologies that are owed as a result of AMD providing the Software to you.

MIT license

Contents

1 Introduction

AMF HQ Scaler is a technique for achieving high-end video upscaling results from lower resolution video inputs. This document provides a complete description of the AMD Advanced Media Framework (AMF) Video HQ Scaler Component. This component performs the following functions:

- HQ Scaling
- Sharpening

2 AMF Video HQ Scaler Component

Video HQ scaler accepts input frames stored in `AMFSurface` objects wrapping DirectX 11/12 textures, Vulkan surfaces, OpenCL surfaces. The output is placed in `AMFSurface` objects wrapping DirectX 11/12 textures, OpenCL surfaces, Vulkan surfaces, depending on the component configuration.

Include `public/include/components/HQScaler.h`

2.1 Component Initialization

The AMF Video HQ Scaler component should be initialized using the following sequence:

1. Create an AMF Context and initialize it for one of the following:
    1. DirectX 11
    2. DirectX 12
    3. Vulkan

4. OpenCL

2. Configure the HQ Scaler component by setting the necessary properties using the `AMFPropertyStorage::SetProperty` method on the HQ Scaler object.
3. Call the `VideoHQScaler::Init` method of the video HQ Scaler object.

2.2 Configuring the HQ Scaler

The HQ scaler supports the following input and output formats:

1. BRGA
2. NV12
3. RGBA
4. R10G10B10A2
5. RGBA_F16
6. P010

The output format must be same as the input and the format conversion is not supported. The parameters of the output stream are set using the following properties:

Name (prefix "AMF_HQ_SCALER_")	Type
ENGINE_TYPE	AMF_MEMORY_TYPE
OUTPUT_SIZE	AMFSize
KEEP_ASPECT_RATIO	Bool
FILL	Bool
FILL_COLOR	AMFColor
ALGORITHM	amf_int64
FROM_SRGB	Bool
SHARPNESS	Float

Table 1. AMF HQ Scaler properties of the output stream

Name: `AMF_HQ_SCALER_ENGINE_TYPE`

Values: `AMF_MEMORY_DX11`, `AMF_MEMORY_DX12`, `AMF_MEMORY_VULKAN`, `AMF_MEMORY_OPENCL`

Default Value: `AMF_MEMORY_DX11`

Description: Specifies the memory type of output surfaces. Surfaces are allocated internally by the HQ Scaler component.

Name: `AMF_HQ_SCALER_OUTPUT_SIZE`

Values: A valid size.

Default Value: `N\A`

Description: Output image resolution specified as `AMFSize`. Scaling will be performed when this property is set.

Name: `AMF_HQ_SCALER_KEEP_ASPECT_RATIO`

Values: `true`, `false`

Default Value: `false`

Description: Force the scaler to keep the aspect ratio of the input image when the output size specified by the `AMF_HQ_SCALER_OUTPUT_SIZE` property has a different aspect ratio.

Name: `AMF_HQ_SCALER_FILL`

Values: `true`, `false`

Default Value: `false`

Description: Specifies whether the output image outside the region of interest, which does not fill the entire output surface should be filled with a solid color. The fill color is specified using the `AMF_HQ_SCALER_FILL_COLOR` property.

Name: `AMF_HQ_SCALER_FILL_COLOR`

Values: `(0,0,0,0)` ... `(255,255,255,255)`

Default Value: `(0,0,0,255)`

Description: Fill color specified as `AMFColor` to fill the area outside the output rectangle. Applicable only when the `AMF_HQ_SCALER_FILL` property is set to `true`.

Name: `AMF_HQ_SCALER_ALGORITHM`

Values:

Name	Description
`AMF_HQ_SCALER_ALGORITHM_BILINEAR`	Bilinear scaling algorithm.
`AMF_HQ_SCALER_ALGORITHM_BICUBIC`	Bicubic scaling algorithm.
`AMF_HQ_SCALER_ALGORITHM_POINT`	Point (nearest-neighbor) scaling algorithm.
`AMF_HQ_SCALER_ALGORITHM_VIDEOSR1_0`	VideoSR1.0 scaling algorithm. This algorithm is based on FSR 1.0.
`AMF_HQ_SCALER_ALGORITHM_VIDEOSR1_1`	VideoSR1.1 scaling algorithm. This algorithm is intended for specific internal integrations and is exposed purely for experimental use. VideoSR1.1 is only supported when `AMF_HQ_SCALER_ENGINE_TYPE` is set to `AMF_MEMORY_DX11`

Name	Description
	or AMF_MEMORY_DX12 and the input and output formats are not NV12 or P010.

Default Value: AMF_HQ_SCALER_ALGORITHM_VIDEOSR1_0

Description: Specifies scaling method.

Name: AMF_HQ_SCALER_FROM_SRGB

Values: true, false

Default Value: true

Description: Convert color space from linear to SRGB.

Name: AMF_HQ_SCALER_SHARPNESS

Values: Float in the range of [0.0, 2.0]

Default Value: 0.5

Description: Control VideoSR scaler sharpening. Applicable only when the AMF_HQ_SCALER_ALGORITHM property is set to AMF_HQ_SCALER_ALGORITHM_VIDEOSR1_0 or AMF_HQ_SCALER_ALGORITHM_VIDEOSR1_1.

2.3 Submitting Input and Retrieving Output

Once the HQ Scaler component is successfully initialized, you may start submitting input samples to it. Input samples must be submitted as AMFSurface objects.

At the same time poll for output by calling AMFComponent::QueryOutput on the HQ Scaler object. Polling for output samples can be done either from the same thread or from another thread.

Suspend submission of input samples briefly when AMFComponent::SubmitInput returns AMF_INPUT_FULL. Continue to poll for output samples and process them as they become available.

2.4 Terminating the HQ Scaler Component

To terminate the HQ Scaler component, call the Terminate method, or simply destroy the object. Ensure that the context used to create the HQ Scaler component still exists during termination.