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

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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:
    a. DirectX 11
    b. DirectX 12
    c. Vulkan
    d. 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

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:

- `AMF_HQ_SCALER_ENGINE_TYPE` – specifies the memory type of output surfaces (surfaces are allocated internally by the HQ Scaler component). Can be one of the following values:
    - *DX11* – place output in a DirectX 11 texture
    - *DX12* – place output in a DirectX 12 texture
    - *Vulkan* – place output in a Vulkan surface
    - *OpenCL* – place output in an OpenCL surface

- `AMF_HQ_SCALER_OUTPUT_SIZE` – output image resolution specified as *AMFSize.* Scaling will be performed when this property is set.
- `AMF_HQ_SCALER_KEEP_ASPECT_RATIO` – Boolean: 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.
- `AMF_HQ_SCALER_FILL` – Boolean: 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.
- `AMF_HQ_SCALER_FILL_COLOR` – 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*.
- `AMF_HQ_SCALER_ALGORITHM` – specifies scaling method. This property can have one of the following values:

  - AMF_HQ_SCALER_ALGORITHM_BILINEAR – use a bilinear scaler
  - AMF_HQ_SCALER_ALGORITHM_BICUBIC – use a bicubic scaler
  - AMF_HQ_SCALER_ALGORITHM_FSR – use a FSR1.0 scaler
- AMF_HQ_SCALER_FROM_SRGB – convert color space from linear to SRGB
- AMF_HQ_SCALER_SHARPNESS– control FSR scaler sharpening. The range of the sharpness is from 0 to 2. Default = 0.5

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.