Video Image Presentation through Sprite Plane for Improving Graphic Performance

A solution with Intel® Embedded Media and Graphics Driver (Intel® EMGD) VAAPI and XVvideo

May, 2014

Document Number: 330608-001US
Executive Summary

An In-Vehicle Infotainment (IVI) is a low-power system installed in automobiles that provides audio and video entertainment, and an automotive navigation system. The IVI depends on the graphics components, graphic software driver and device, to run the 3D-navigation system and to present video streams to user. The graphics components in the IVI use a 3D-graphics engine to display the video image by converting the original YUV video image to an RGB image, and then blending that RGB to produce a final image for output to the display. The original method uses a great deal of graphic 3D engine resources. This document offers a new approach for video image conversion and presentation using sprite to reduce graphic resource use and improve performance.

This solution presents a new way for video image conversion and presentation that will reduce graphic resource use and improve performance.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel’s tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today.

Contents

Background ............................................................................................................. 4
Solution .................................................................................................................. 6
Applying and Testing the Solution .............................................................................. 7
    Prerequisites ................................................................................................. 7
    Preparing the OTC’s Video Driver ............................................................. 8

Figures

Figure 1. 3D Process to Convert Video Image and Present to Display ....................... 5
Figure 2. Improved Method to Present the Video Image. ........................................... 6
Intel® EMGD for Intel® Atom™ processor E38XX Series uses the Intel Open Source Technology Center (OTC) VAAPPI media solution for hardware accelerated video decoding and video image presentation. The video image presentation method provided by the OTC’s VAAPPI media stack uses a traditional method that places heavy usage on 3D graphic engines.

A graphic chipset can be categorized into three major functions:

- A display engine that takes an image from memory and displays it to screen
- A 2D/3D graphic engine that performs 2D/3D operations and images conversion
- A video decode/encode engine that processes video data

In most graphic media chipsets today, video decode engines output video images in NV12 format. This is YUV 4:2:0, in which each pixel is approximately 1.5 bytes in length, and the U and V data are shared among 4 pixels. However, display outputs for IVI are generally RGB 24/32 format, in which each pixel is 4 bytes in length and has its own R, G, and B. The following is an example of conversion from YUV to RGB:

\[
R = 1.164(Y - 16) + 1.596(V - 128) \\
G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\
B = 1.164(Y - 16) + 2.018(U - 128)
\]
The original VA-API media solution converts this NV12 video image to an RGB format image and sends it to X server. The X server then blends the RGB video image to a final output image for display. The following graphic illustrates this simple process:

**Figure 1. 3D Process to Convert Video Image and Present to Display**

This traditional video presentation method poses a few problems:

1. The RGB image has a larger size compared to NV12; graphic components need more resources to access it.
2. The conversion method takes about three writes (each R, G, and B) and seven reads (Y, U, and V).
3. It needs a floating point number, for which the conversion generally uses matrix multiplication, thus using more 3D GPU cycles.
4. Two steps are required to produce the final output image: a conversion, and then a blend.

Most of these issues consume more resources on the graphic 3D engines.
To reduce the resource use on the graphic 3D engine, as described in the background section, Intel® EMGD provides a solution on top of Intel OTC’s VAAPI media stack and Linux* X11 using the sprite plane in the graphic display engine to present the video image, instead of using the 3D graphic engine and converting it to RGB format.

The sprite is an internal display plane in the graphic display engine that supports a YUY2 image format. YUY2 are YUV 4:2:2, for which each pixel is about 2 bytes in length, and the U and V data are shared between two pixels. The size is smaller than RGB 24/32 and the conversion from NV12 to YUY2 directly copies the Y, U, and V data, saving significant 3D engine resources.

The display engine displays the YUY2 image directly through the sprite. The display engine reads the YUY2 image through the sprite plane, the RGB image through the display plane, and blends them together, which pipes and outputs it to the screen.

This blending is done in the display engine and does not require any 3D engine resources.
Applying and Testing the Solution

This approach requires an X-Window system. To enable this solution, the user must patch the OTC’s video driver. The patches can be found in the Intel® EMGD package for Linux®. Currently, Intel® EMGD provides patches for OTC’s video driver version 1.2.0 and version 1.2.1 (the same patch works on both versions). The solution for the Intel® EMGD DDX driver for Linux® X11 is already included in the open source XOrg server package.

The following sections provide detailed instructions for patching the OTC’s video driver and the video playback tool GStreamer-VAAPI. Generally, these instructions work as described if the specified software versions are installed on the target system. In cases where the installed software versions are different from those described here, some manual patching may be required and the instructions can still be used as a rough guideline.

Prerequisites

- Development tools package
- Intel® EMGD Driver package for Linux®
- libXv development package (libXv-devel)
- Git
- GStreamer 1.0.7 or later
- GStreamer base, and a good plug-in
- Patch tool
Preparing the OTC’s Video Driver

1. Choose one of the download options.
   a. Download the package intel-driver-1.2.1 as release tar.gz.
      i. Download the following link.
         http://cgit.freedesktop.org/vaapi/intel-driver/snapshot/intel-driver-1.2.1.tar.gz
      ii. Untar the tar ball.
          # tar zxf <filename>.tar.gz
   b. Clone the driver.
      i. Git clone.
         # git clone git://anongit.freedesktop.org/vaapi/intel-driver
      ii. Checkout version 1.2.1.
         # git checkout 1.2.1

2. Go into intel-driver directory.
   # cd intel-driver

3. Locate the patch file Intel_VA_Driver_1_2_0_Add_VAXV_Feature_Patch.
   It is located in the patches\common\VA_Driver_i965\VAXV folder of
   IEMGD_HEAD_Linux.tgz in the driver's release package.

4. Patch the intel-driver project.
   # patch -p1 < Intel_VA_Driver_1_2_0_Add_VAXV_Feature_Patch

5. Autogen and configure the intel-driver.
   # ./autogen.sh
   The user will see the "xv" enabled as one of the windowing systems at the end, as shown in the following example:
   autoreconf: Entering directory `.
   autoreconf: configure.ac: not using Gettext
   ...
   ...
   Windowing systems ..................... : drm x11 xv wayland

6. Build and install the intel-driver.
   # make
   # make install
Preparing Gstreamer-VAAPI to Test the Solution

1. Clone Gstreamer-VAAPI.
   
   ```bash
   # git clone git://gitorious.org/vaapi/gstreamer-vaapi.git
   ```

2. Checkout the specific commit, this commit is used during testing and enabling.
   
   ```bash
   # git checkout 2e356b0f7efae33fb943ad11204020dcdbf1b04f
   ```

3. Go to the gstreamer-vaapi directory:
   
   ```bash
   # cd gstreamer-vaapi
   ```

4. Locate the patch file
   
   ```bash
   Gst_Plugins_VAAPI_pre_0_5_8_Adding_Rendermode_Patch.
   ```
   
   It is located in the patches\common\VA_Driver_i965\VAXV folder of IEMGD_HEAD_Linux.tgz in the driver’s release package.

5. Patch the Gstreamer-VAAPI project.
   
   ```bash
   # patch -p1 < Gst_Plugins_VAAPI_pre_0_5_8_Adding_Rendermode_Patch
   ```

6. Autogen and configure the intel-driver.
   
   ```bash
   # ./autogen.sh
   ```

7. Build and install intel-driver.
   
   ```bash
   # make
   # make install
   ```

Running the Test

To test on GStreamer-VAAPI, add an additional option after vaaapisink, render-mode=overlay.

Example:

```bash
# gst-launch-1.0 filesrc location=<video clip> ! <demuxer> ! vaapidecode ! vaaapisink render-mode=overlay
```


Authors

Teng En Ung is a Graphic Software Engineer at Intel Corporation.

Boon Wooi Tay is a Graphic Software Engineer at Intel Corporation.

Acronyms

DDX Device Dependent X

IVI In-Vehicle Infotainment

OTC Open Source Technology Center

RGB Color encoded using Red, Green, and Blue

VAAPI Video Acceleration API
Video Image Presentation through Sprite Plane
for Improving Graphic Performance

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.
You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel, Intel Atom, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2014 Intel Corporation. All rights reserved.
Video Image Presentation through Sprite Plane for Improving Graphic Performance

§