NVIDIA NVAPI Interfaces for per pixel Intensity and Warping

Starting with one of the R302 NVIDIA driver releases we will provide NVAPI Interfaces on Windows 7 (32 and 64 bit) to control per pixel intensity, including black level adjustment, and warping of the desktop.

The new function calls are:

```
typedef struct
{
    NvU32                           version;              //IN version of this structure
                                                          //   NV_SCANOUT_INTENSITY_DATA_V2
    NvU32                           width;                //IN width of the input texture
    NvU32                           height;               //IN height of the input texture
    float*                          blendingTexture;      //IN array of floating values
                                                          //   building an intensity RGB
                                                          //   texture
    float*                          offsetTexture;        //IN array of floating values
                                                          //   building an offset texture,
                                                          //   setting to NULL if the black
                                                          //   level texture is not in use
    NvU32                           offsetTexChannels;    //IN number of channels per pixel
                                                          //   in the offset texture, either
                                                          //   1 or 3
} NV_SCANOUT_INTENSITY_DATA;

NVAPI_INTERFACE NvAPI_Gpu_SetScanoutIntensity(
    NvU32                           displayId;             // IN combined physical display
                                                           //    and gpu identifier of the
                                                           //    display to apply the
                                                           //    intensity control on.
    NV_SCANOUT_INTENSITY_DATA      *pScanoutIntensityData; // IN pointer to the intensity
                                                           //    texture data
    int                            *pbSticky);             // OUT indicates whether the
                                                           //     settings will be kept over
                                                           //     a reboot or not.

typedef enum _NV_GPU_WARPING_VERTICE_FORMAT
{
    NV_GPU_WARPING_VERTICE_FORMAT_TRINAGLESTRIP_XYUVRQ  = 0,  // 2d vertex + 4d texture
                                                              // -coordinates, the vertices
                                                              // will form a triangle strip.
    NV_GPU_WARPING_VERTICE_FORMAT_TRINAGLES_XYUVRQ      = 1,  // groups of 2d vertex + 4d
                                                              // texture coordinates, forming
                                                              // independent triangles.
} NV_GPU_WARPING_VERTICE_FORMAT;

typedef struct
{
    NvU32                           version;               // IN version of this structure
                                                           //    NV_SCANOUT_WARPING_VER
    Float                           *vertices;             // IN width of the input texture
    NV_GPU_WARPING_VERTICE_FORMAT   vertexFormat;          // IN format of the input vertices
    int                             numVertices;           // IN number of the input vertices
    NvSBox                          *textureRect;          // IN rectangle in desktop
                                                           //    coordinates describing the
                                                           //    source area for the warping
} NV_SCANOUT_WARPING_DATA;

NVAPI_INTERFACE NvAPI_Gpu_SetScanoutWarping(
    NvU32                            displayId;            // IN  combined physical display
                                                           //     and gpu identifier of the
                                                           //     display to apply the
                                                           //     intensity control on.
    NV_SCANOUT_WARPING_DATA         *pScanoutWarpingData;  // IN  pointer to the warping data
```

```
    int                             *piMaxNumVertices,      // OUT indicates how many vertices
                                                            //     are allowed as maximum for
                                                            //     this function.
    int                             *pbSticky);             // OUT indicates whether the
                                                            //     settings will be kept
                                                            //     over a reboot or not.
Additional information for display assignments:

NVAPI_INTERFACE NvAPI_Gpu_GetScanoutConfiguration(
    NvU32                           displayId;              // IN  combined physical display
                                                            //     and gpu identifier of the
                                                            //     display to be queried.
    NvPhysicalGpuHandle             hPhysicalGpu,           // IN  handle of the physical gpu
                                                            //     to apply the keystoning on.
    NvU32                           outputId,               // IN  deviceMask of display to
                                                            //     apply the keystoning on.
    NvSBox                          desktopRect,            // OUT Desktop relative area of the
                                                            //     windows desktop which is
                                                            //     taken into account for the
                                                            //     scan out.
    NvSBox                          scanoutRect);           // OUT Scan out mode relative area
                                                            //     to which the desktopRect is
                                                            //     scanned out.
```

These functions are to be used in conjunction with the existing NVAPI functions to enumerate gpus and displays as reference for example on:

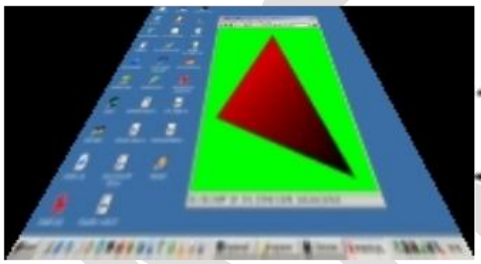
http://http.developer.nvidia.com/nvapi/index.html

The warping and intensity control functions will have an effect on physically attached displays per physical gpu as opposed to the logical displays and logical gpus which may group one or more physical display or gpu.

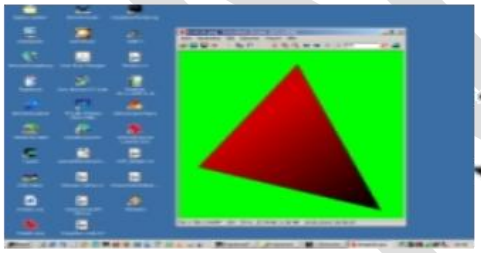
In the 1st step the intensity and warping functions will be executed after the present step (which may contain a downsample step of antialiased rendering) to the desktop and a possible scaling operation but before a possible gamma adjustment. A diagram is shown in the next page

Scanout with
gamma correction

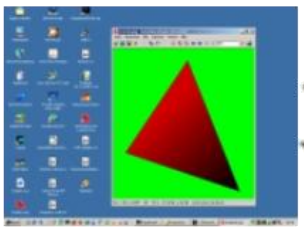
Per pixel
intensity
control



Warping



Scaling



Presenting the
3D content to
the desktop

Rendering
3D content