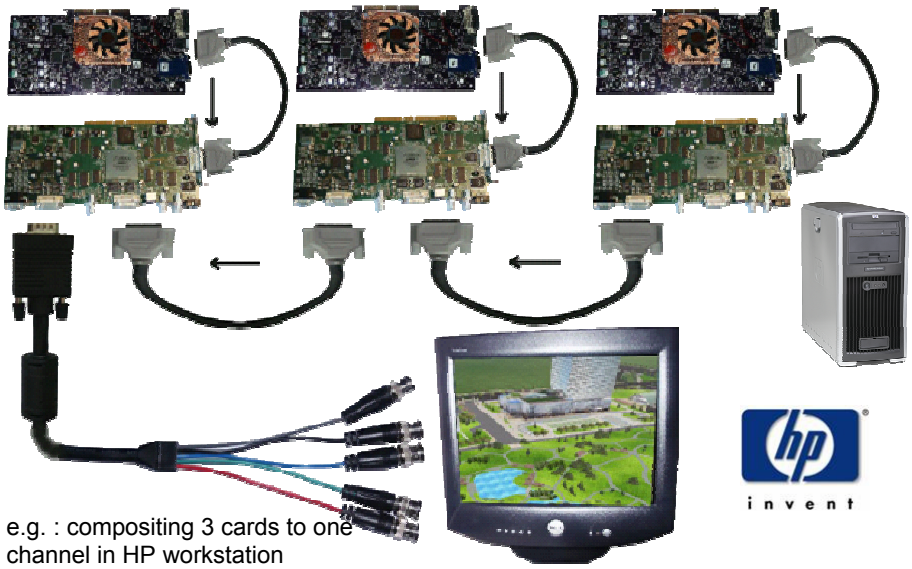
Orad's DVG : solutions for scalable graphics clusters



Graphics Hardware 2004

images courtesy MPI, Barco

Scalable architecture



e.g. : compositing 3 cards to one channel in HP workstation

The DVG architecture

- PC cluster solution for **scalable** performance on multiple or SINGLE displays.

  M nodes drive N displays, M & N are any numbers, M>N
- At the heart of the DVG technology is the DVG board, a highly programmable (FPGA) PCI card.

  => achieving genlocking AND combining.
- The DVG board **directly interacts** with the graphical board allowing **fully linear scaling** of performance
- A dedicated patent-pending pixel bus "DVG-bus" interconnects the DVG units so no graphics , CPU or motherboard resources are sacrificed
- DVG board occupies space of 1 to 2 PCI slots depending on version.

3

Combining (a.k.a. "chaining")

The DVG supports several different combining modes:

- Sample division (for Anti-Aliasing)
- Time division
- Image division
- Eye division etc...

Composition is done on board, no need to move pixels back into memory with associated overhead in performance and # of CPUs.

Various chaining modes can be mixed

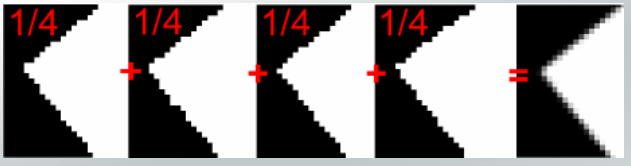
Modes can be switched with no change of wiring

Combining configuration can be chosen according to the critical system/data-base resource

4

More on combining : Anti-aliasing

- Anti-aliasing is critical for image quality
- One DVG rendering unit can implement for instance 4-sample anti-aliasing but with a certain performance cost (fill rate)
- Multiple DVG rendering units connected together (see below) can do up to 16-sample anti-aliasing with same performance as a 4 sample rendering
- This mode does not add delay



5

More on combining : Time-division chaining

- This mode scales performance linearly with the number of units in the chain, without any limit
- With N units in the chain and 60Hz clock each node generates 60/N images per second.
- Therefore each rendering unit has N field times to generate its graphics
- So it can render graphics N times more complex than a single unit could
- The final output of course is still a 60 images per second display
- This mode generates latency of N-1 frame times

6

More on combining : Imaging division chaining

- Each rendering unit is “responsible” for a sub-region of the final output image
- This mode is better suited for applications where performance is limited by pixel fill such as high resolution formats
- No added latency
- Special dynamic load balance algorithms:

  “Quadrant division” with a dynamic rendering time criterion

  “Interleaved division”
- The “Quadrant division” method can also improve the geometry processing using culling

7

Imaging division chaining cont’d

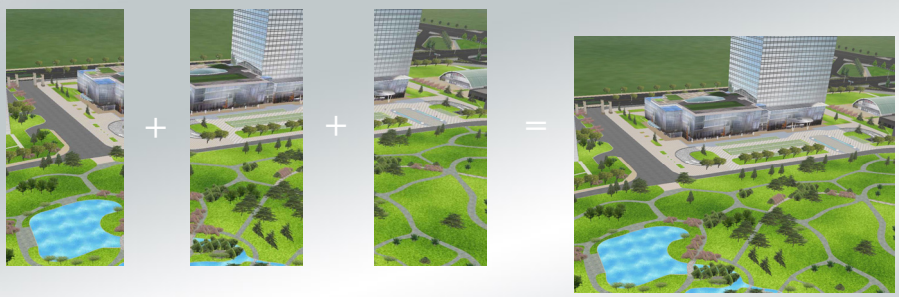
- Image division effectively increases available texture memory, even more so as it is combined with AA chaining :
  - eg : a 1600x1200 renderer at 6 samples uses 95Mb of texture memory for buffering, remaining only **33Mb** (assuming 128 Mb total)

  While 1024 x 768 rendering screen at 2 samples uses 15Mb of texture leaving **133 Mb available !**

=> Image division and anti-aliasing chaining increase available texture memory for graphic textures

8

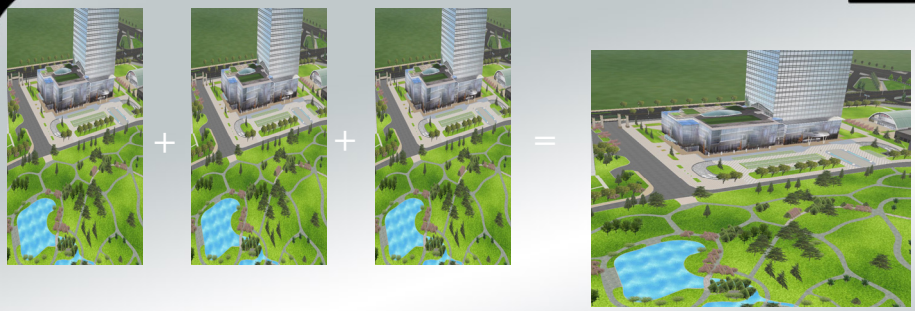
Split image division chaining



- Each unit renders fragment of the scene. Combiner creates output image bigger than input components.
- Both vertical and horizontal image division is allowed
- Application can use view culling to gain geometry rate
- Static load-balanced gain on pixel fill rate

9

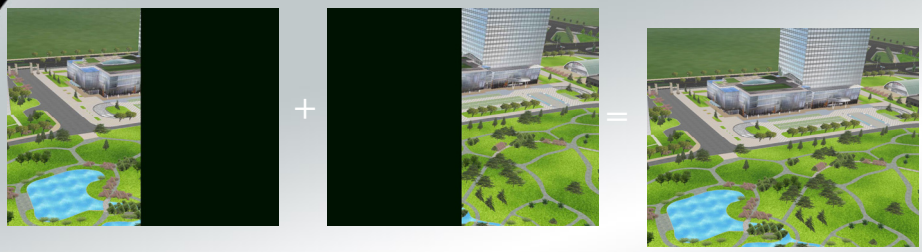
Interleaved image division chaining



- Each unit renders full scene, but in window “squeezed” horizontally
- Each window has projection matrix shifted by a subpixel. Combiner interleaves pixels to produce output image with higher resolution.
- No gain on geometry rate but dynamic load-balanced gain on pixel fill rate
- Cannot use antialiasing of graphic card until programmable sample locations are available.

10

Dynamic image division chaining



- Each unit renders fragment of the scene in viewport smaller than full window. The rest of window is filled with black. Combiner adds images.
- Application can use view culling to gain geometry rate.
- Viewports can be resized on the fly, so application can do dynamic load balancing for pixel fill rate.
- Overhead time (e.g. ‘swapbuffers’) is bigger (because each unit renders in full window).

11

More on combining : Eye division chaining

- **For active stereo (96,110,120hz), left/right eye division : linear performance increase**
- **The first chosen method in CAVE, VRs, as it is one of the most efficient.**
- **No added delay**

12

More on combining : Scene division chaining

- The 3d objects in the scene are split between the different rendering units
- Each pixel's distance to the camera (its Z value) is transmitted between rendering units
- Based on this information, the DVG boards in the chain correctly composite the final image
- Performance scales linearly with the number of rendering units
- Differently from the other methods, this method not only increases total performance, but also effectively increases graphics resources such as texture memory, beyond the capability of a single rendering unit.
- No added delay but no access to FSAA

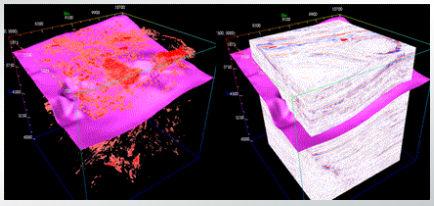
13

More on combining: Volume rendering division

- Many visualization applications, primarily in oil & gas and medical imaging are based on volume, not polygonial rendering
- The process can be implemented on COTS graphics hardware

The cluster linearly increases performance

- no added delay



14

DVG Formats

- The DVG supports all formats up to output pixel frequency of 300 MHz (!!!)
- Supported VESA and other standards:

  All SDI and HD video formats

  All 800x600, 1024x768, 1280x960, 1280x1024 formats

  1280x1024 120Hz (stereo)

  1600x1200 (60-85Hz)

  1792x1344 (60,75Hz)

  1856x1392 (60, 75Hz)

  1920x1440 (60, 75Hz)

  2048x1536 60Hz

15

Software certification and porting

- Easy porting of generic (Vega Prime, Performer, Amira, Opticore Opus, Virtools…) and customized rendering applications, authoring, assembly and data base manipulation tools
- 3 methods for porting :
  - DVGlib – a library used for the DVG system administration providing DVG specific functionality like setting up chain configuration, format, as well as synchronized swap buffer.
  - DVG wrapper – intercepts OpenGL calls in order to configure rendering for compositing
  - DVG wire – allows non-cluster applications (CATIA DMU, EDS/UGS Viz Mockup, PTC DV mockup...) to run in distributed mode on a DVG-based cluster
- Successful porting processes have already been implemented by customers like British Aerospace, Lockheed Martin and others (typical duration – few hours)
- The DVG supports both Windows and Linux based rendering codes
- Orad offers porting assistance services by a dedicated team
- The OpenGL driver is the unmodified driver supplied by the graphics card maker and thus provides the most up-to-date extensions and optimizations as they are released

16

More DVG features

- **Hardware based image post – processing:**
  - NVG / Flir “look”
  - Chromakey (for augmented reality…)
- **Multiple video insertions (mapped on a polygon or as overlay)- Optional**
  - Instructor’s video
  - Cockpit monitors
  - Collaborative session
  - Augmented reality

17

More DVG features

- VIZ CLUSTER MANAGEMENT (“VCM”):

Encompass all the HW and SW technlogies designed by Orad for visualisation cluster management :

=> Multiple renderers can be shared by a group of users over a network. Depending on the day’s use, ressources are allocated across the network to each user (eg : a 4 channel with 2 renderer on each becomes a 2 channel with 4 renderer on each)

=> Cluster Permanent Availability “CPA” software : any broken renderer in a channel can automatically be by-passed so that for instance the channel has lower AA but continues working

VCN is managed by the DVG service software which also Handles the reconfiguration of combining depending on application, as well as selection of formats. Manages VCN from a logical/SW standpoint

18

DVG applications

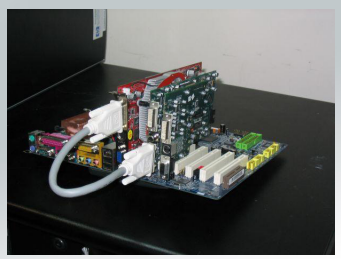
- CAVEs, Workbench, Flight simulators…
- Civil and military simulations
- Mission planning
- Urban planning
- Car design
- Car driving simulation
- Interactive walk through
- Theme parks
- Architectural design
- Scientific/Medical visualization
- Collaborative Engineering
- Museums, Planetariums and Cultural Centers
- Hazard Perception / Disaster Management
- Oil & Gas explorations
- Homeland Security
- Augmented reality

19

ANNEX : miscellaneous supporting examples

20

Form-factors : DVG VR-X (1/2)

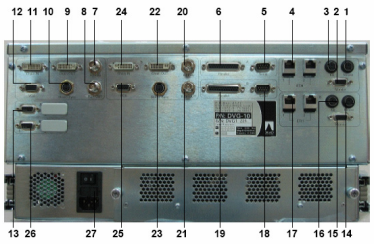


Screen shots from an integration in HP's XW8000 workstation, Orad's privileged partner for workstations

21

Form-factors : DVG10 VR (2/2)

Orad DVG10 VR backpanel. Orad's offering for integrated rack-mounted PCs.



Number on Illustration	Label on DVG-10 VR Back Panel	Description
Render A:		
1	Mouse	Mouse connector
2	Monitor	Monitor out connector
3	KB	Keyboard connector
4	ETH	Ethernet network connector
5	Serial	Serial RS232 connector
6	Parallel	Parallel port connector
7	REF IN	Synchronization multiple DVG chain
8	REF OUT	Synchronization multiple DVG chain
9	CHAIN OUT	Chaining multiple DVG render output
10	STEREO SYNC	3D Glasses control
11	CHAIN IN	Chaining multiple DVG render input
12	VGA	VGA 15-Pin output
13	--	Connector for JTAG cable used for upgrading DVG firmware
Render B:		
14	Mouse	Mouse connector
15	Monitor	Monitor out connector
16	KB	Keyboard connector
17	ETH	Ethernet network connector
18	Serial	Serial RS232 connector
19	Parallel	Parallel port connector
20	REF IN	Synchronization multiple DVG chain
21	REF OUT	Synchronization multiple DVG chain
22	CHAIN OUT	Chaining multiple DVG render output
23	STEREO SYNC	3D Glasses control
24	CHAIN IN	Chaining multiple DVG render input
25	VGA	VGA 15-Pin output
26	--	Connector for JTAG cable used for upgrading DVG firmware
27	--	Power

22