



**PERCRO** Perceptual  
Robotics Laboratory

# Real-Time Network Streaming of Dynamic 3D Content with In-Frame and Inter-Frame Compression

P. Gasparello, G. Marino, F. Bannò,  
F. Tecchia, M. Bergamasco

15th IEEE/ACM DS-RT 2011  
MediaCITYUK, Salford, UK  
September 4 – 7, 2011

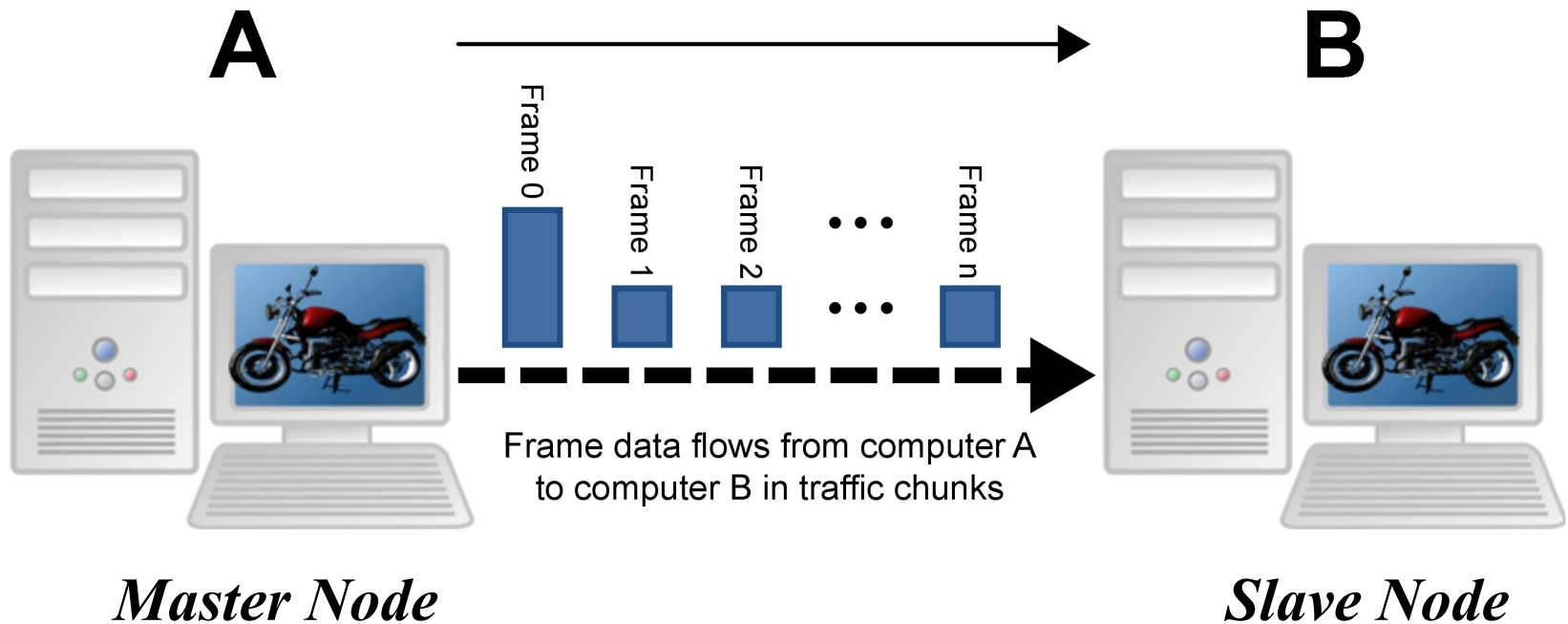


# Introduction

- The main idea is to *capture* an OpenGL command sequence in real-time and to *transmit* it towards one or more remote node to be *replicated*
- The proposal is the exploitation of both In-Frame and Inter-Frame redundancies in order to reduce the amount of data to be sent, and allow the system to work even when the bandwidth is limited



# Overall System Architecture

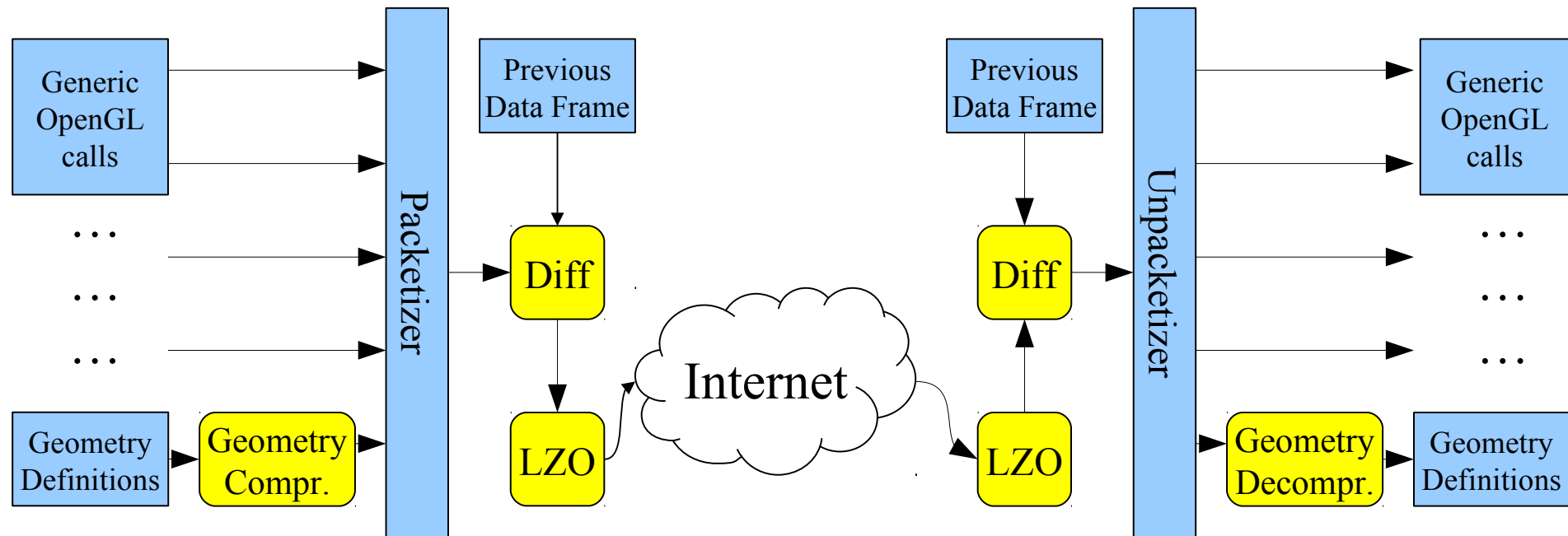


OpenGL commands are captured on the Master Node by means of a custom OpenGL driver. Those calls are then *compressed, packed,* and *sent* to the **one or more** slave nodes.



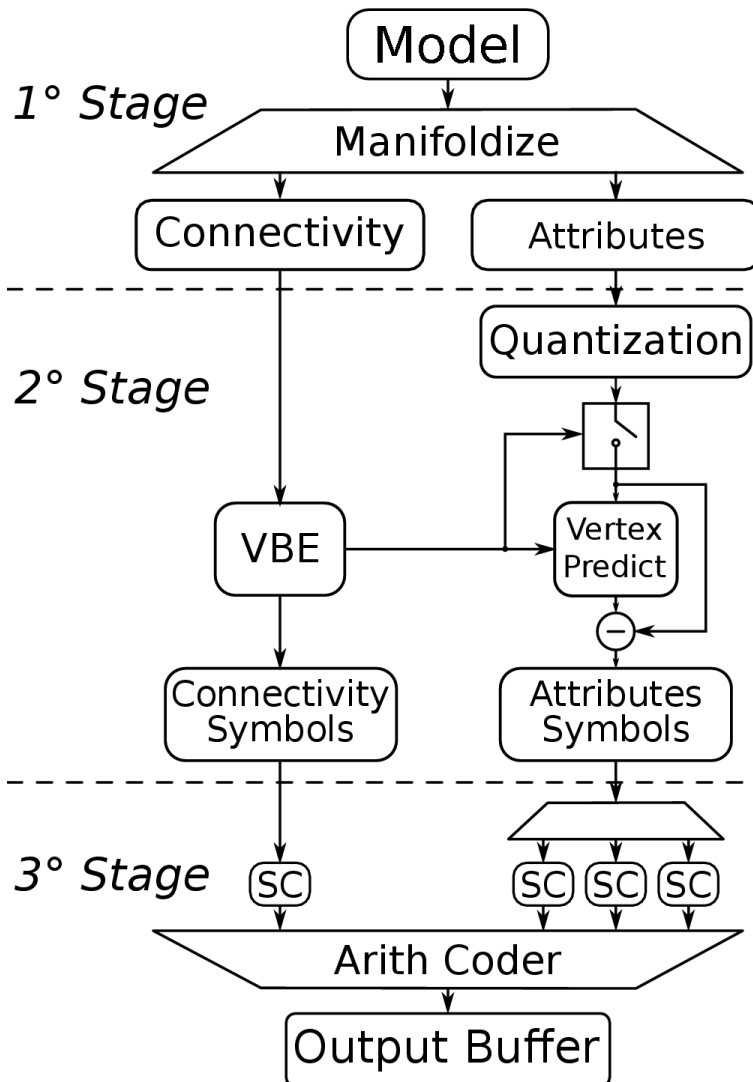
# Compressing Real-time 3D Stream

OpenGL calls are intercepted on the fly (at driver level),  
and compression/decompression occurs in real-time





# Geometry Compression

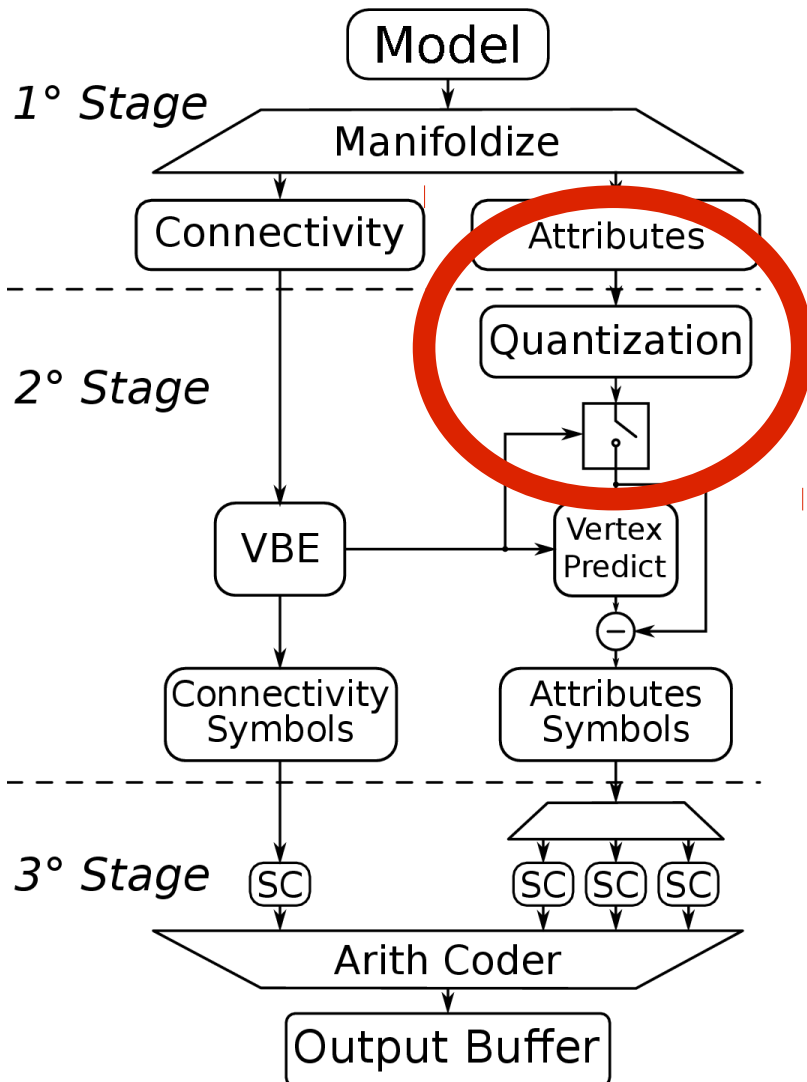


3D Data structures are inferred by analysing the overall OpenGL command stream:

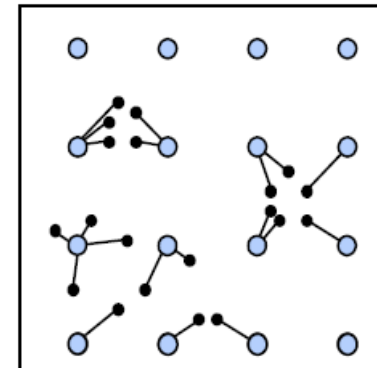
- Immediate mode
- Display lists
- Vertex arrays
- Vertex buffer objects



# Geometry Compression

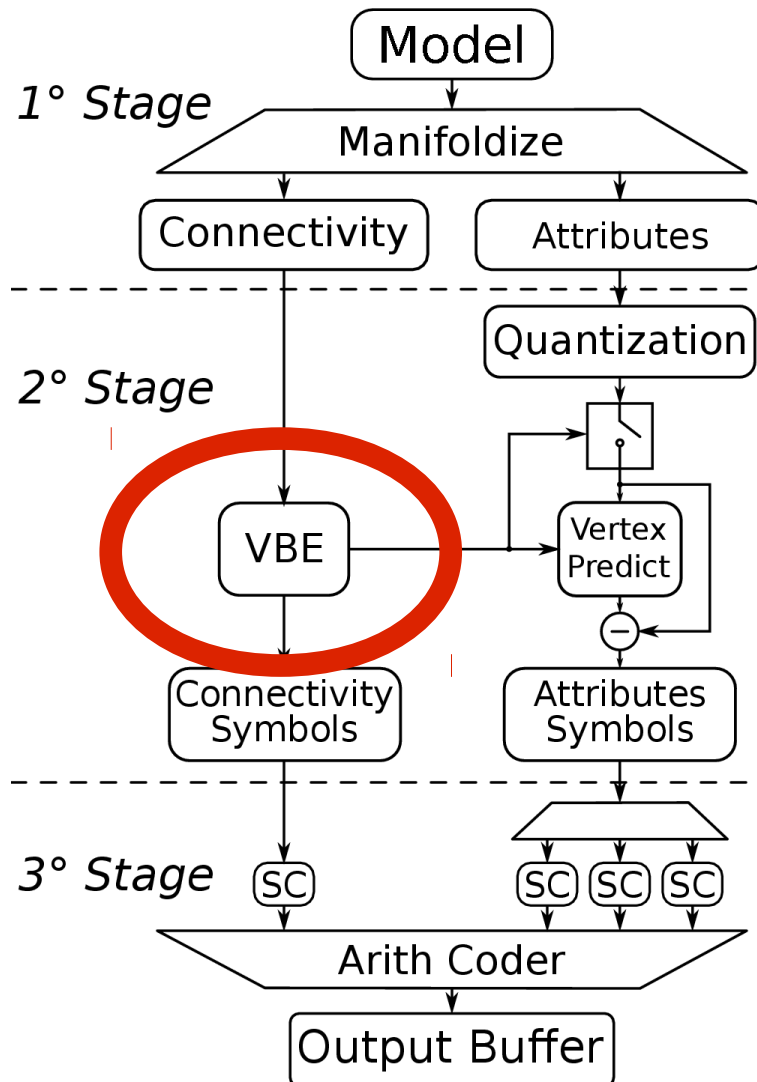


*Uniform Quantization  
of Coordinates*

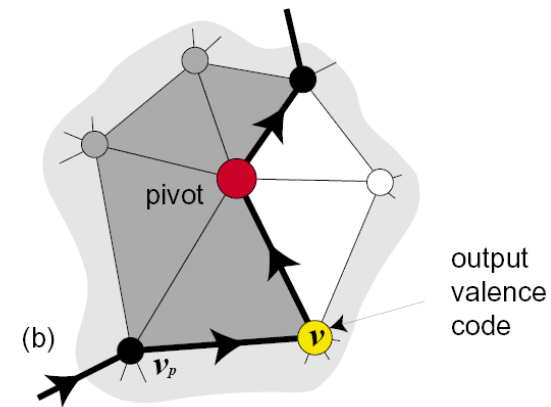
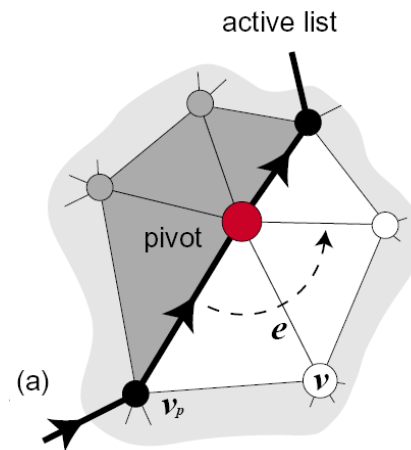




# Geometry Compression



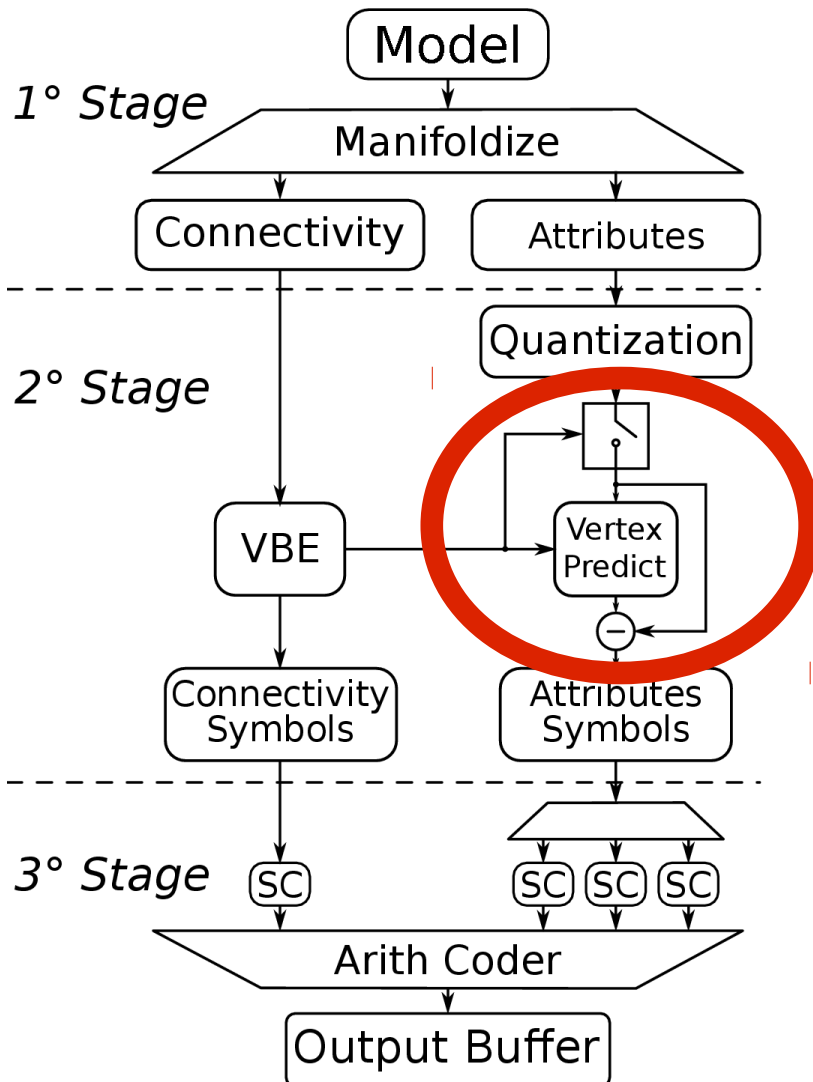
## Single Rate Valence Based Encoding



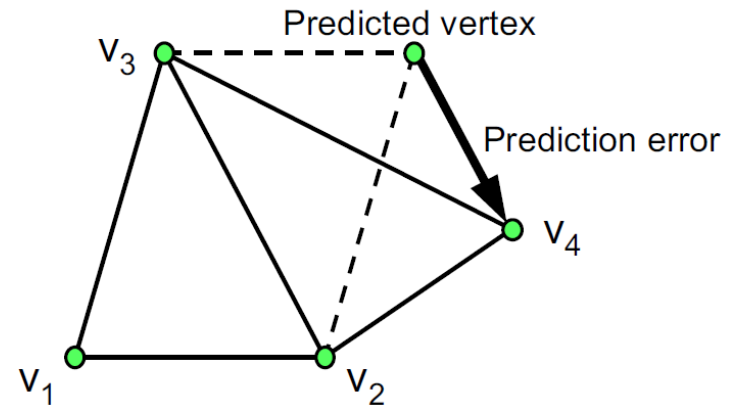
(ADD 6)



# Geometry Compression



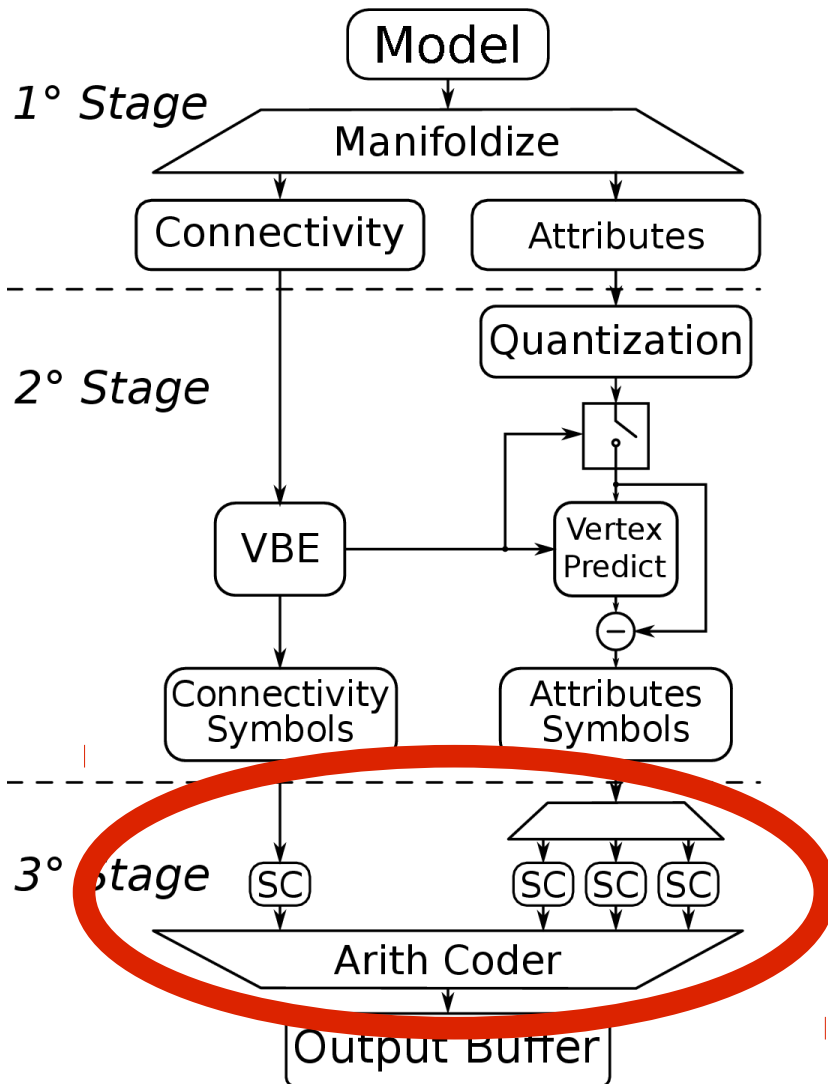
## Vertex Prediction by Parallelogram Rule







# Geometry Compression

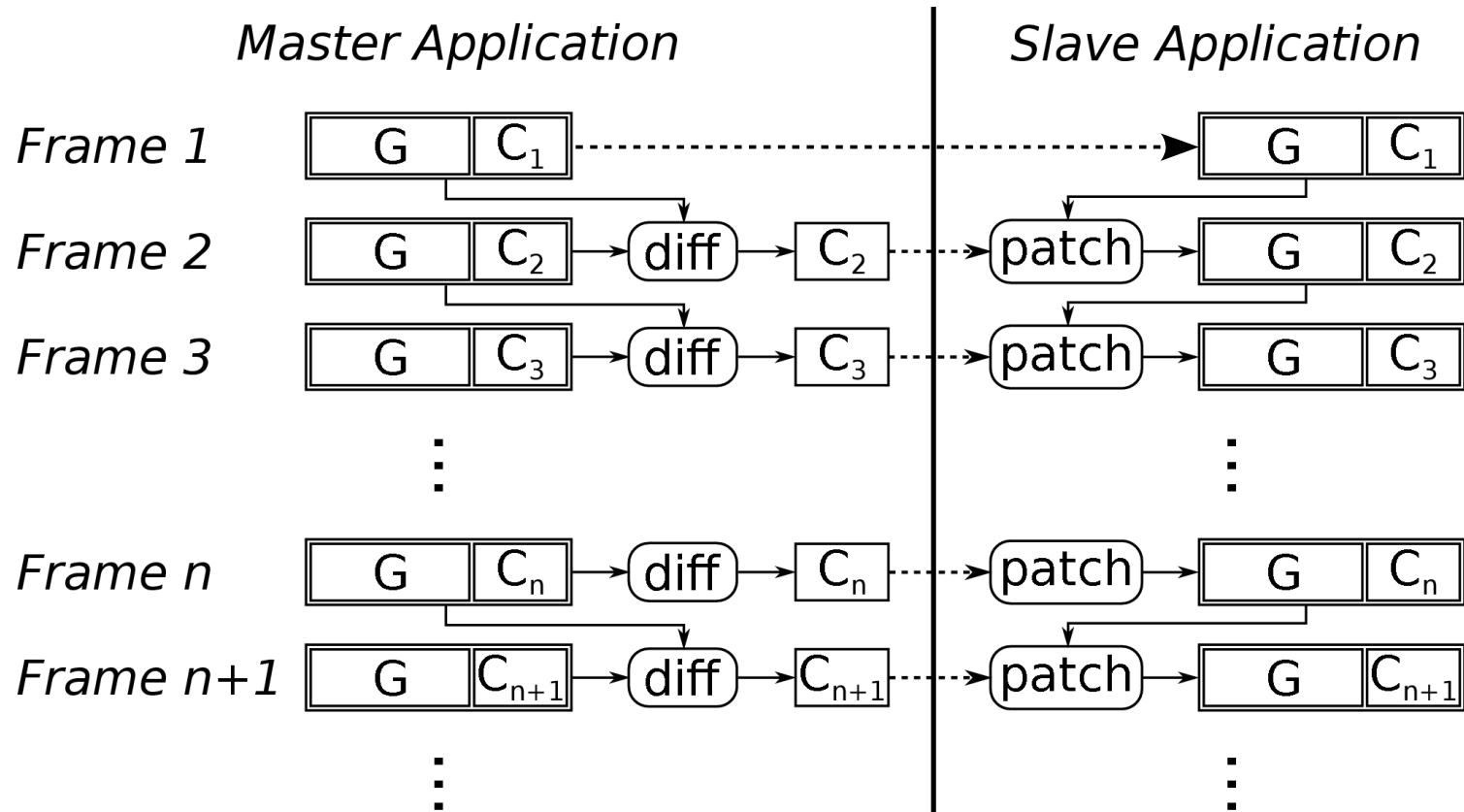


Entropy Compression by a  
*Context Based  
Arithmetic Encoding*

Multiple Contexts for a  
better statistical modeling



# Exploiting Frame-to-Frame Coherence

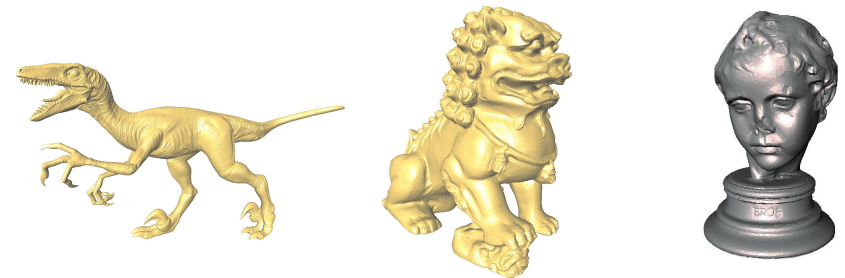
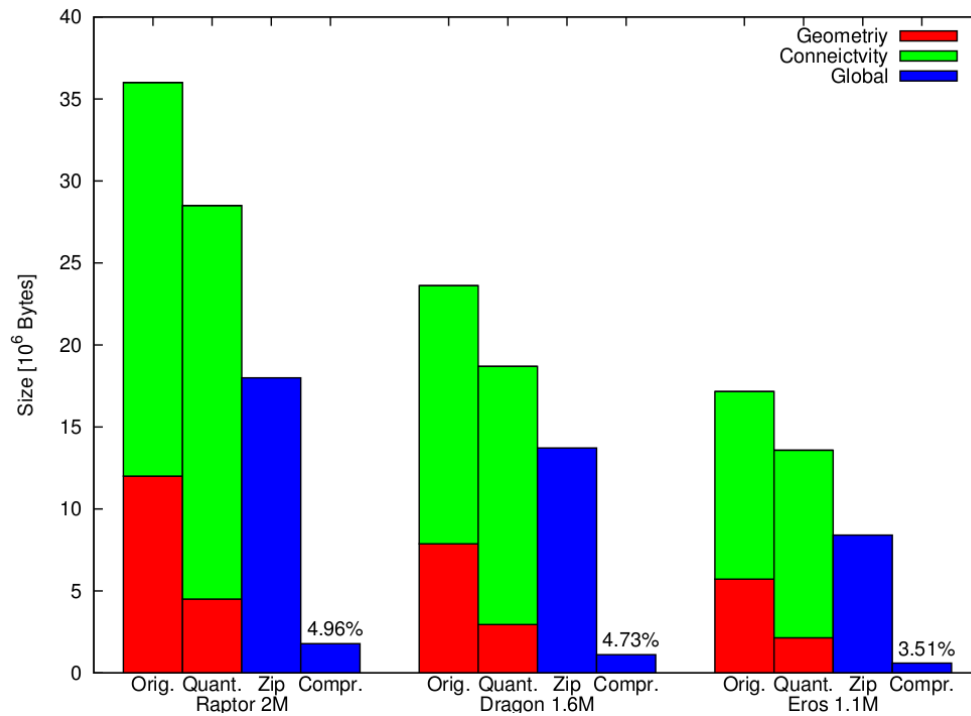


Inter-frame redundancy is exploited by using *VCDIFF* (RFC3284)



# Mesh Compression Results

- Compression Ratio up to 3.5% of the original size (saving up to 96.5%)
- Throuthput of compression is on average from 100K up to 250K vertex per second
- Test Machine:
  - \_ Intel i5 M520 2.4GHz

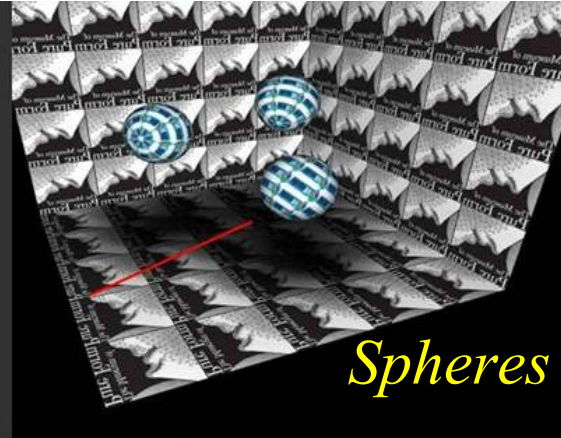


Model	# Vert.	Time [s]	Thr. [vps]
Raptor	1M	4.132	242K
Dragon	656K	5.851	112K
Eros	477K	3.940	121K



# Streaming Measurements

*Motorbike*



*Spheres*



*Rollercoaster*



# Streaming Measurements [Bytes]

With only *LZO*

<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	33,375	4,602	4,602
Motorbike	6,147,768	1,705	1,705
Rollercoaster	18,359,992	18,410	18,410

With *LZO* plus *In-Frame*  
Compression

<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	<b>10,663</b>	4,602	4,602
Motorbike	<b>2,093,678</b>	1,705	1,705
Rollercoaster	<b>7,655,571</b>	18,410	18,410

With *LZO* plus *Inter-Frame*  
Compression

<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	33,375	<b>3,931</b>	<b>3,931</b>
Motorbike	6,147,768	<b>208</b>	<b>208</b>
Rollercoaster	18,359,992	<b>92</b>	<b>92</b>

With *LZO* plus *In-Frame* plus  
*Inter-Frame* Compression

<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	<b>10,663</b>	<b>3,931</b>	<b>3,931</b>
Motorbike	<b>2,093,678</b>	<b>208</b>	<b>208</b>
Rollercoaster	<b>7,655,571</b>	<b>92</b>	<b>92</b>



# Streaming Measurements [ms]

With only *LZO*

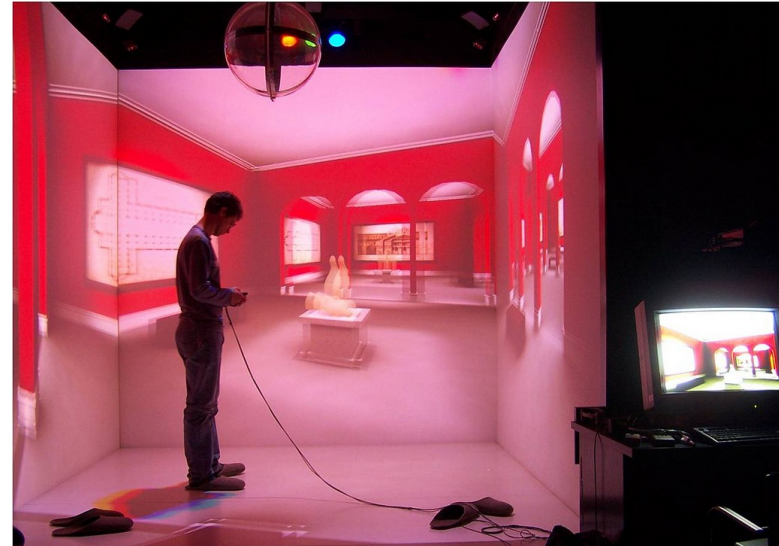
<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	12.6673	4.0159	3.95106
Motorbike	1,167.56	27.4411	29.8014
Rollercoaster	20,523.3	41.9717	32.6774

With *LZO* plus *In-Frame* plus  
*Inter-Frame* Compression

<i>Model</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>
Spheres	80.5976	7.16921	4.81178
Motorbike	9,440.13	17.3465	18.9491
Rollercoaster	32,382.5	54.3147	28.9925



# Sample Applications





## Conclusions and Future Work

- We realized a system capable of distributing real-time generated content using on-the-fly compression/decompression of OpenGL command stream.
- We advanced beyond the state-of-the-art by introducing *in-frame geometry compression* of 3D data structures and by exploiting *inter-frame redundancy*.
- Measurements suggest a significant potential whenever the amount of data bandwidth is limited, for instance in Internet distributed interactive applications.
- Future works are focusing on:
  - investigating techniques for progressive transmission of frame data containing 3D model descriptions.
  - the introduction of a state tracker able to make the system robust in case of data loss. Data packets containing OpenGL state modifications could be sent over TCP connections, while the other ones could be sent with faster UDP datagrams.





# Acknowledgments

- This work was funded by the project BEAMING, acknowledging the financial support of the Seventh Framework Programme for Research of the European Commission DG-INFSO-D2, undergrant number 248620.
- Authors wish to thank the Salford CVE group and the other laboratories that are using our system for the feedback and for the help given on improving our software.



thank you!

[p.gasparello@sssup.it](mailto:p.gasparello@sssup.it)