The Future of Game Engines

Towards Real-time Photo-realistic Rendering and Natural Character Animation

Xiaomao Wu

R&D Software Engineer, Crytek

Shanghai, Aug. 1st, 2010

Overview

- Current technologies in game engines
- Emerging technologies and the future

Current Technologies in Game Engines

Global Illunimation (GI)

- CryENGINE® 3: Light propagation volume (Dynamic scene + dynamic lights)
- LPV-based GI in CryENGINE® 3 was proposed by Anton Kaplanyan (Crytek) in 2009
- Unreal 3 Lightmass (static lighting and effects)
- Middleware solutions are also available:
  - Geomerics
  - Beast (used by Unity engine)
  - Lightsprint

GI in CryENGINE®3

Video

LightsMark by Stepan Hrbek
(Now called *LightSprint*)

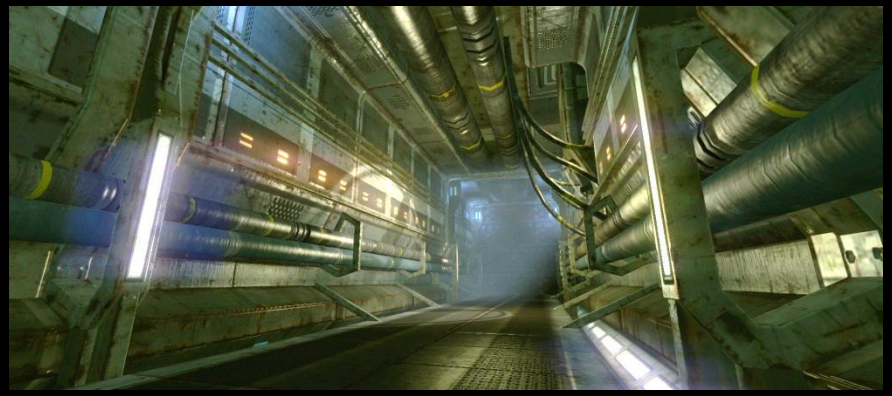
- Real-time
- Tested with *World of Padman*
- Provides test binary

SSAO

- Widely used in video games
- First proposed by Vladimir Kajalin at Crytek (2006)
- Used for the first time in Crysis® (2007)
- Very popular in games: Crysis®, Crysis Warhead™, Gears of War 2, Uncharted 2, Halo, Starcraft II …

SSAO

- Pros:
  - High quality
  - No pre-processing, no loading time allocations
  - Memory efficient
  - Independent on scene
- Cons:
  - Only visible occluders on the screen are considered → May produce artifacts



Without SSAO

With SSAO

Deferred rendering

- Proposed by Michael Deering and his colleagues at SIGGRAPH ’88
- Decouple geometric scene complexity from shader
- Computation and memory bandwidth reduced to those visible portions
- More complex to handle transparency
  - Depth peeling for order-independent transparency -> additional batches and g-buffer size
  - But DX10, 11 can perform batches fast, no less effective than forward shading

Deferred rendering

Image rendered by CryENGINE®

Linear z value of GBuffer

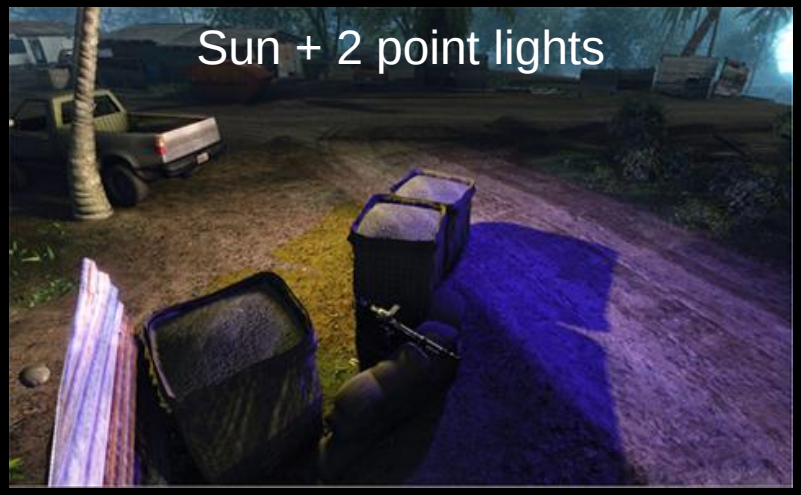
SSAO from GBuffer

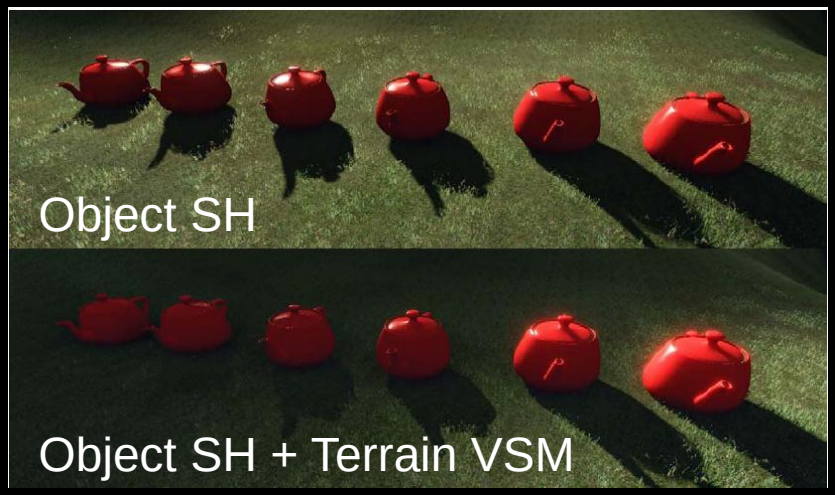
Shadows

- Dynamic Occlusion Maps
- Shadow maps with screen-space randomized look-up
- Shadow maps with light-space randomized look-up
- Shadow mask texture

Shadows

- Shadow maps for directional light sources (Cascaded shadow maps)
- Deferred shadow mask generation
- Unwrapped shadow maps for point lights
- Variance shadow maps (for terrain)



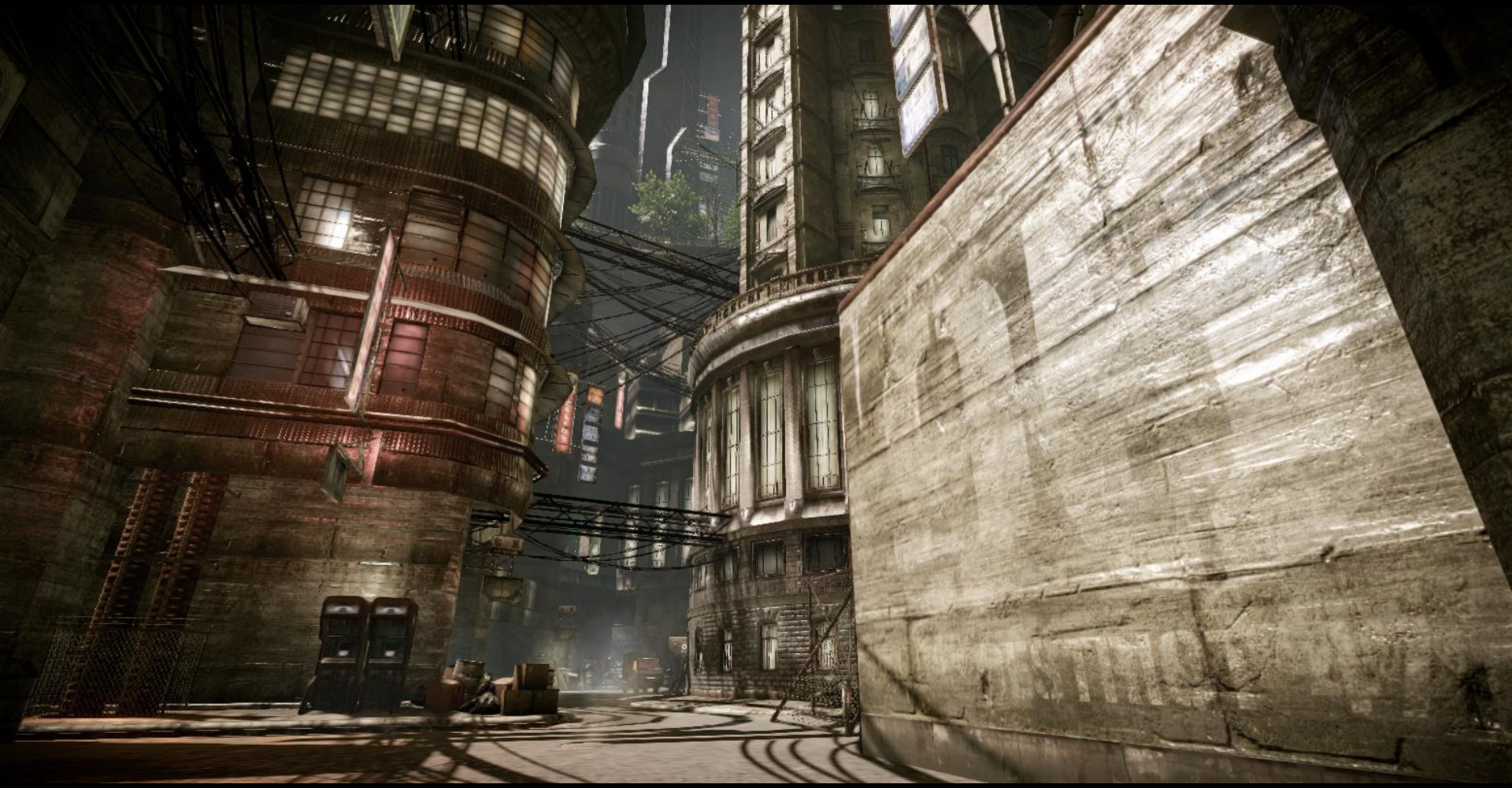


Anti-aliasing

- Edge AA in CryENGINE®
- Morphological anti-aliasing (God of War III)

CryENGINE®3, Without EdgeAA

CRYTEK



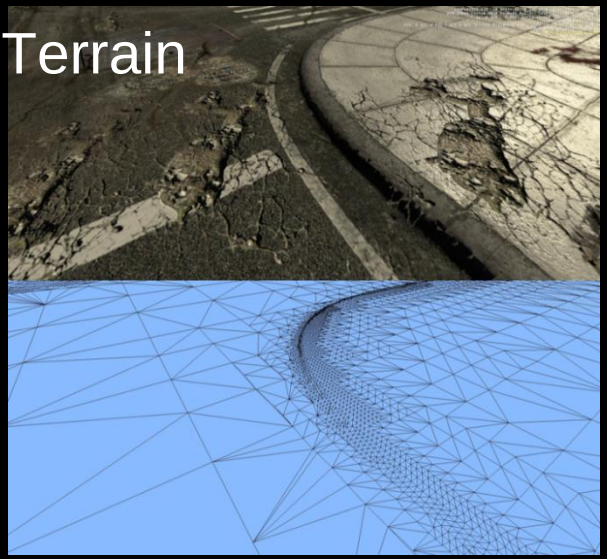
CryENGINE®3, with EdgeAA

CRYTEK

Copyright by Sony Entertainment
http://uk.ps3.ign.com/dor/objects/886158/god-of-war-iii/images/god-of-war-iii-20100114113513333.html

Volumetric environment

Copyright by Moro et al. A Fast Rendering Method for Shafts of Light in Outdoor Scene. http://nis-lab.is.s.u-tokyo.ac.jp/nis/cdrom/cgi/NICO06full.pdf



Post Effect

- Depth of field (with bokeh)
- Motion blur
- Lens flares
- Bloom/glow
- Color grading

DOF with bokeh

Copyright by Sebastian @Crymod
http://crymod.com/thread.php?threadid=59961

Motion blur

Lens flare

Copyright by Xzero @Crymod
http://img39.imageshack.us/img39/674/screenshot0729.jpg

Color Grading

- Change the color style of a game in real-time
  - User provide a reference chart
  - The engine change the game scene with the color chart
  - Useful to change the game style, and to create expressive tones

Color grading in CryENGINE® 3

Motion control

- Real-time Locomotion Groups in CryENGINE®
  - Time warping
  - Parameterization
  - Real-time motion control (player, AI)
  - Weight-shift, uphill/downhill
  - UE engine
  - Animation tree

LMG-based Control in CryENGINE®

Video

Real-time Inverse Kinematics

- CryENGINE®
  - Analytic IK (up to 8 DOF for Aliens in Crysis® 2)
  - Example-based IK
  - CCD IK
  - Jacobian IK
- Human-IK
- IK types
  - Foot IK
  - Look IK
  - Aim IK
  - Full-body IK

IK demo video (CryENGINE®)

Integrated with physics

- Rag-doll
- Hit-reaction
- Smooth transition between physics and animation data

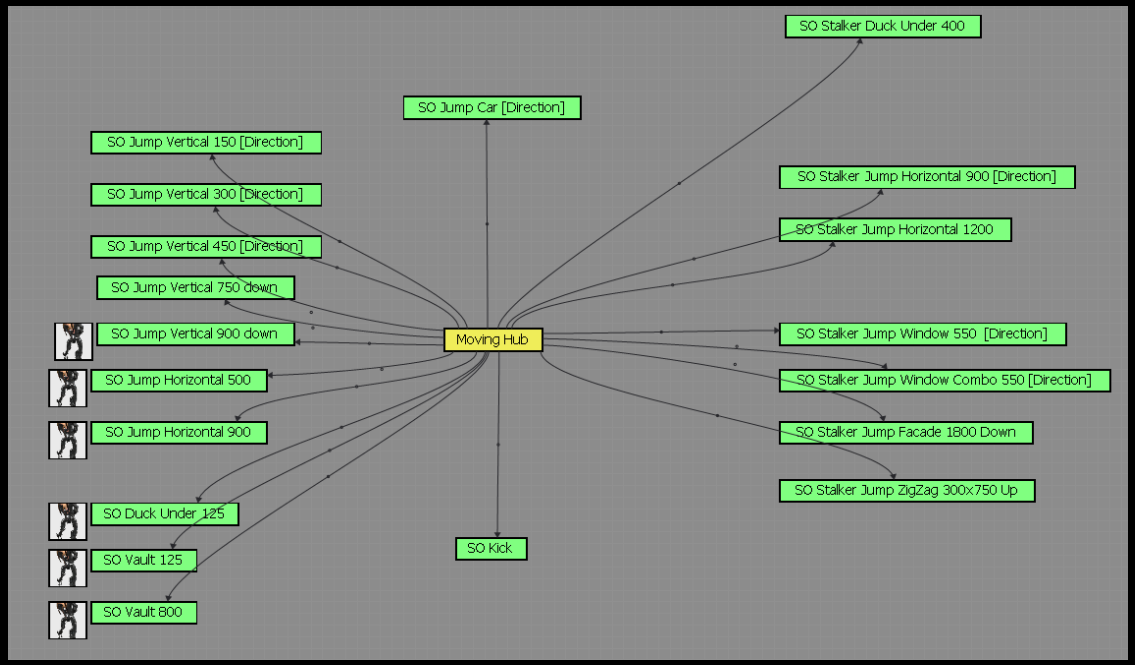
Additive Animations

- Additive animations are relative animations: add to the base animation
- Usage:
  - Breathing
  - Looking around
  - Flinching
  - Posture change
  - And more…

Video

Animation Behavior

- Behavior tree
  - A table that defines a character's behavior
- Script-based AI logic
  - Lua script defines the behavior of an NPC
- Animation Graph
  - A flow graph which defines the behavior



Animation graph (CryENGINE®)

1	Signal	Parent	Values Change
2	OnEnemySeen		a.Cover3["Target_Status"] = "Confirmed"
3	OnSomethingSeen		a.Cover3["Target_Status"] = "Possible"
4	OnHeardSound		
5	OnGroupStateChange		a.Cover3["Group_State"] = AI.GetGroupTacticState(userTbl.id, 0, GE_GROUP_STATE);
6	OnGrenadeDanger		a.Cover3["Explosives_Danger"] = true
7	ResolvedExplosivesDanger		a.Cover3["Explosives_Danger"] = false
8	OnFallAndPlay		a.Cover3["FallAndPlay"] = true
9	OnFallAndPlayWakeUp		a.Cover3["FallAndPlay"] = false
10	OnNoTargetVisible		
11	OnOutOfAmmo		
12	OnReloadDone		
13	entered_vehicle		
14	exited_vehicle		
15	entered_vehicle_gunner		
16	exited_vehicle_investigate		

Animation behavior (CryENGINE®)

Skinning

- Linear blend skinning
  - Collapsing
  - Candy wrapping
- Dual quaternions
  - Preserve rigidity: Reduced skin collapsing
  - GPU friendly
  - Coordinate Invariant
  - Flipping artifact (discontinuity >180 rotation)
- The authors' TOG '08 paper proposed solution to eliminate the "flipping" artifact

Multiple CPU/GPU

- Windows multi-threading API
- ATI cross fire, Nvidia SLI
- OpenMP
  - May not be efficient for game production
- Intel TBB
  - UE
  - Dreamworks
  - Autodesk Maya
- OpenCL
- CUDA

Emerging Technologies And the Future

DX 11 and opengl 4.0

- New features
  - DirectCompute
  - Tessellation
  - Improved multi-threading
  - Shader Model 5.0
- Chances
  - Real-time high-quality subdivision surfaces
  - Real-time micro-polygon rendering
  - Convenient dynamic shader binding
- Risk
  - Pipeline change (code side and art side)

New Rendering Approach

-Point Based Rendering

- Pros
  - Naturally fits to automatic LOD schemes
  - Freeform modeling
  - Anti Aliasing
- Cons
  - Makes sense only on super high detailed scenes
  - Editing tools still rely greatly on triangles
  - GPUs focus on rasterization

New Rendering Approach

-Real-time ray-tracing

- Pros
  - Easy Parallelizeable
  - Naturally leads to high quality shadows and lighting
  - Rendering pipeline simplification
  - Anti Aliasing
- Cons
  - Still too slow for realtime application
  - GPUs focus on rasterization

New Rendering Approach

-Real-time ray-tracing

- Stephan Reiter, Real–time Ray Tracing of Dynamic Scenes, June, 2008
  - That thesis is interesting, complex scene with interactive speed without heavy optimization and recent HW

Copyright by Johannes Kepler Universitat and Stephan Reiter
http://stephanreiter.info/wp-content/uploads/2010/02/thesis-lq.pdf

New Rendering Approach

-Real-time ray-tracing

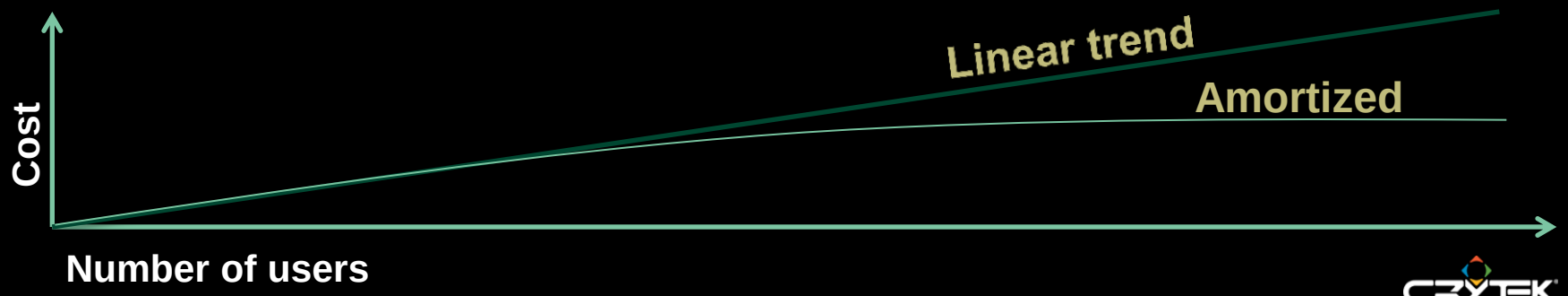
Copyright by Nvidia®, captured from the publically available Optix SDK

PantaRay (Used for Avatar)

- Nvidia ported Weta's PantaRay engine to a CUDA-based GPU version
- 25 times faster by utilizing an Nvidia Tesla S1070 GPU-based server

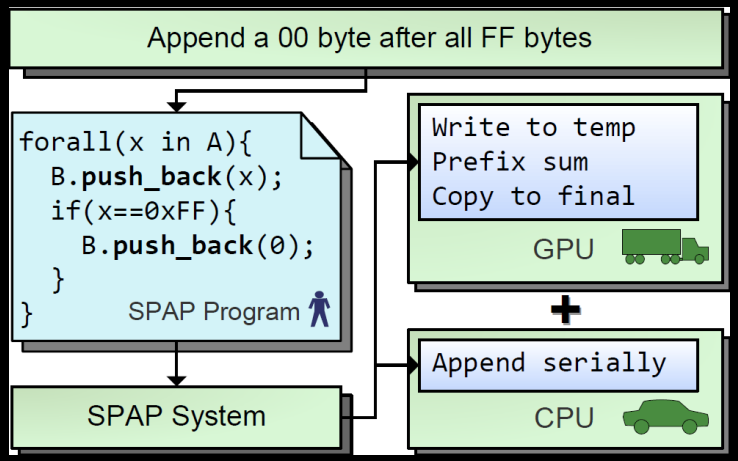
Server-side rendering

- 4G networks have a good ground for that
  - Low ping – a strong requirement for real-time games
  - Will be widely deployed in 5-7 years
- Compression of synthesized video
  - Temporally decompose the video details
  - Use perception-based importance
    - Salience maps + user-side eye-tracking
- Need to amortize cloud-rendering cost per user:

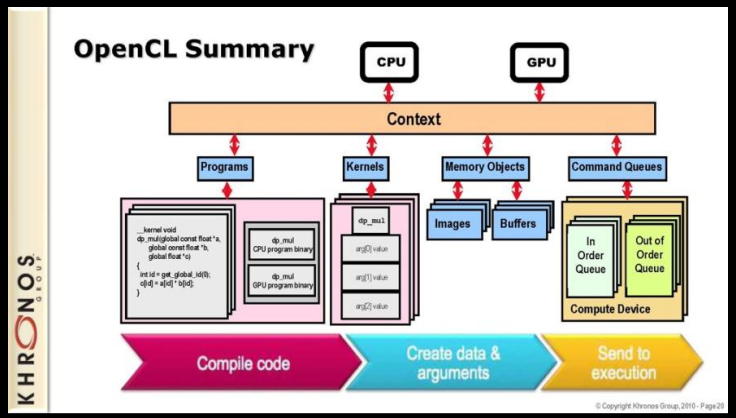


Heterogeneous system

- OpenCL
  - Not as transparent as SPAP
- SPAP (Same language for all processors, GPU+CPU)
  - Research product



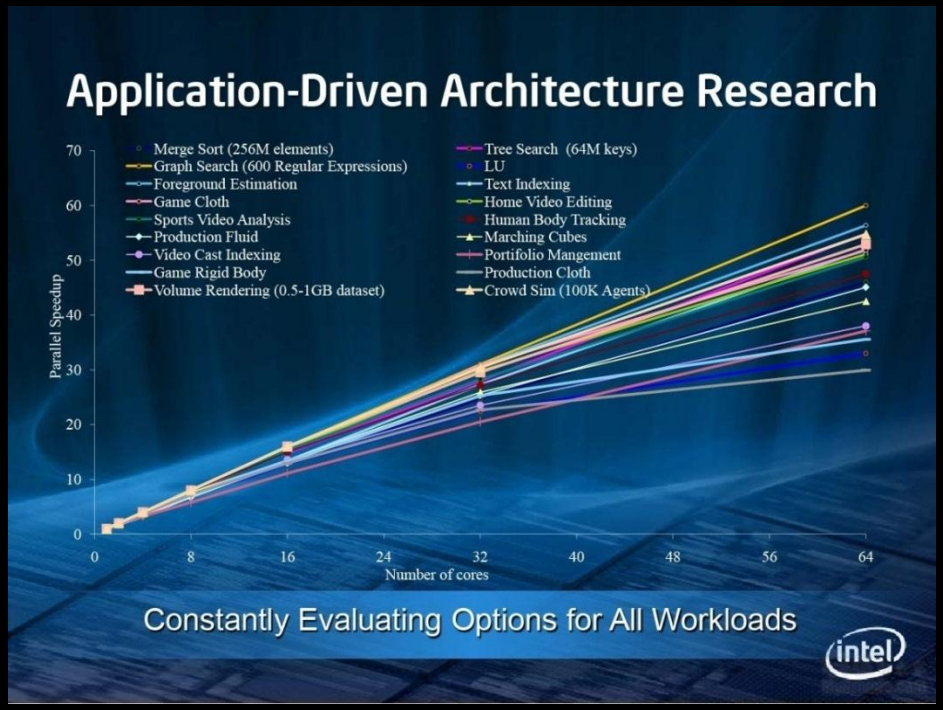
Copyright by Q. Hou, K, Zhou and B. Guo
http://www.kunzhou.net/2010/SPAP-TR.pdf



Copyright by the KHRONOS Group
http://www.khronos.org/opencl/

Heterogeneous system

- **Intel MIC: 22nm, 50+ Cores**
  - **Code Knights Corner**



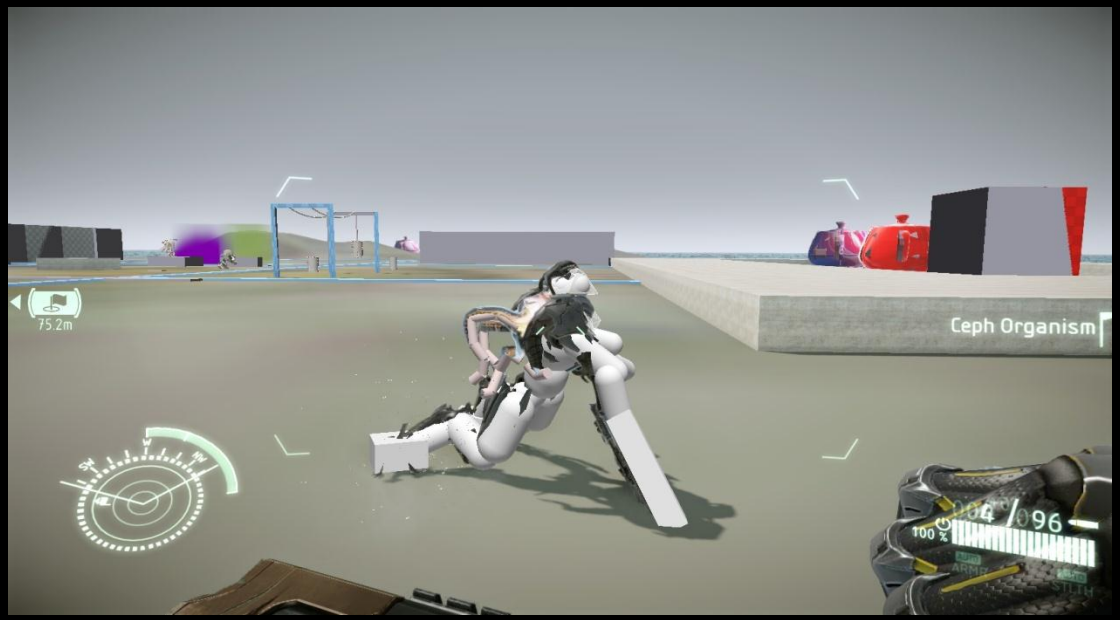
http://download.intel.com/pressroom/archive/reference/ISC_2010_Skaugen_keynote.pdf

General and accurate motion control

- Geostatistics (Kriging)
  - Optimizing both kernel parameters and weights: better fitting
  - More accurate interpolation than RBF, Inverse weighting, k-nearest neighbors ….
  - About half speed as RBF: optimization tricks
- Optimal control theory
- Probability motion control
- Recent machine learning R&D are driving this direction

Interactive & Dynamic Ragdoll

- Make it more realistic
- CryPhysics and Nvidia Physx both support it
- Challenges: real-time performance, stability



Natural and Real-time IK

- Combine with biomechanics info
- Learning prio from large motion databases
- Extend 12-DOF analytic IK
- UniversalIK?
  - HumanIK, MonsterIK, AlienIK, AnimalIK ….
  - Learning from artist-defined sparse poses
  - Use data from biometrics study

More Automated Rigging System

- Possible solutions
  - Region-of-Influence (J. P. Lewis et al. 2007)
  - Heat-transfer (Baran and Popvic, 2000)
  - Skinning by example
- Pure automatic skinning won’t available in 2-3 years, but an highly automated method that can decrease manual weight adjustment to a large extent, is possible

Robust and Real-time Motion Retargeting

- Different skeleton topology
- Allow manual interaction in game editor
- Directions:
  - Decompose and re-combine
  - Analyze the behaviors of different creatures
  - Through an intermediate skeleton
  - Rely on a fast, robust full-body IK solution to maintain constrains
  - A simplified workflow of the SIGGRAPH ’08 paper [Chris Hecker et al.]

Smarter Characters

- Perception (speech, facial expression, environment)
- Social intelligence
  - Predict the actions of others by their motives and emotional states. Game theory, decision theory, emotion modeling
  - Learning-based: learning game players' behavior and apply to the AI characters: online and offline
  - Keep updating AI characters' skills and knowledge at run-time and make them human-like
  - Neural network
  - Reinforcement learning
  - Products: DrivatarTM(Gaming research team, Microsoft Research Cambridge)

Thank you! Qestions?

- Crytek is hiring: www.crytek.com/career/
- For any questions contact me:
  - xiaomao@crytek.com
  - Special thanks to

Cevat Yerli (CEO, Co-funder of Crytek)

Carl Jones (Global Director of BD - CryENGINE)

Ury Zhilinsky (R&D Manager)

Anton Kaplanyan (Research Lead)

Ivo Herzeg (Lead R&D animation Engineer)

Scott Fitzgerald (Technical Designer)

Vaclav Kyba (R&D Engineer)

The whole R&D team at Crytek

Three reference slides from Cevat & Anton‘s HPG ’10 talk

How to design for the future?

- Facts
  - Fixed Resolution for Gaming till 2012
  - HD 1920 x 1080 @ 60 fps
    - Stereoscopic 3D experience: 30 fps per eye
  - Limited by current consoles hardware
  - Risk of “Uncanny Valley“ for contents
    - Perception-driven approaches!
  - Till 2012 majority of games must use artistic style, physics and AI to differentiate!
    - What‘s the current artistic style? Desaturate colors?

Challenges of Future

- Technology challenges
  - Switching to a scaleable codebase
    - Think of parallelism & async jobs
    - Multithreading, scheduling
    - Larger codebases, multiple platforms & APIs
- **Production challenges**
  - Cost of assets increase by ~50% annually
    - Content, quality increases. More & more „interactive“
    - Think to improve Tools, Pipelines & Bottlenecks to counter-effect , automate Source Back-Ends ➔ Resource Compilers
    - The better the tools, the cheaper and/or the better your output

Opportunities in Future

- Short-term user impact opportunities till 2012
  - The delta in visual opportunities is limited, BUT...
  - for the next 3 Years: Huge gains are possible in Physics, AI and Simulation of Special Effects
    - ➔ Focus around that knowledge can lead to very different designs
- Mid-erm 2013+ creative opportunities
  - Future console generations
    - ➔ New Rendering Methods will become available
    - ➔ The renaissance of graphics will arrive
  - Allows new visual development directions that will rival full CGI feature films qualty
  - Action point: Link yourself to console cycle