The 19th VISUALIZATION CONFERENCE

10年目を迎える OpenGL ES と OpenGL最新情報

株式会社ディジタルメディアプロフェッショナル
大渕　栄作

29/Nov/2013

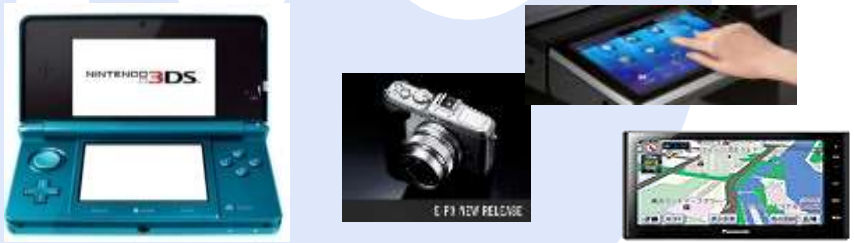
Agenda

- DMP概要
- OpenGL ES 10周年と最新情報
- OpenGL アップデート
- おまけ：glTFのご紹介

ディジタルメディアプロフェッショナル概要

- 設立: 2002年7月
- 所在地: 東京都中野区
- 資　本　金: 8億2千万円
- 主要株主: (株)日本政策投資銀行、他
- 事業内容: 1. ｸﾞﾗﾌｨｯｸｽIPｺｱのﾗｲｾﾝｽ
  2. ｸﾞﾗﾌｨｯｸｽLSIのﾌｧﾌﾞﾚｽ開発・製造

本社(中野セントラルパーク)



当社IPコアを使った顧客製品例

当社LSI製品

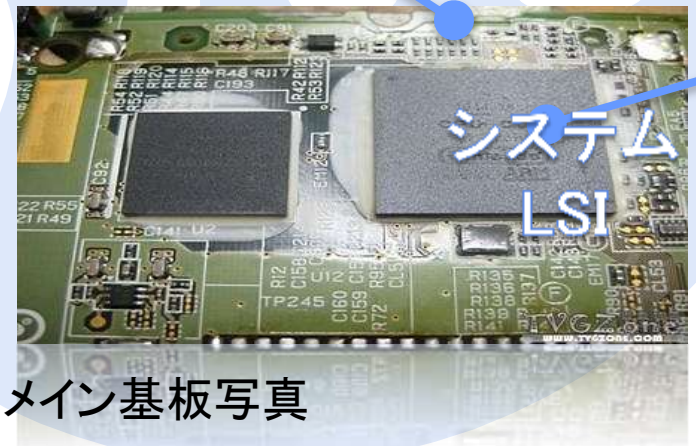
- 特許等: 日米欧において63件の特許申請済み　成立38件

グラフィックスIPとは

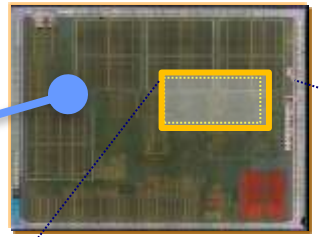
IP（Intellectual Properties） = 知的財産

ﾋﾞｼﾞﾈｽﾓﾃﾞﾙ： DMPはIPの対価としてライセンス料及び製品出荷量に応じたロイヤリティーを受け取る

顧客製品



メイン基板写真



システムLSI内部写真

グラフィックスIPとは

- 画面描画機能を受け持つ部分
- DMPがIP(回路図、ｿﾌﾄｳｪｱ)を提供

DMPグラフィックスIP製品

組込み機器向け高性能・低消費電力グラフィックスIP コア

- 高性能2D/3Dグラフィックス IP
- 低電力モバイルから高性能アミューズメントまでサポート
- ビルディング・ブロック構造によるスケーラブルなアーキテクチャ

（企業部門　最高賞）

フォトリアリスティック
3Dグラフィックス
（OpenGL ES 1.1 互換 + 独自拡張）
PICA200

標準3Dグラフィックス
PICA200Lite (OpenGLES 1.1)
SMAPH-S (OpenGLES 2.0/3.0)

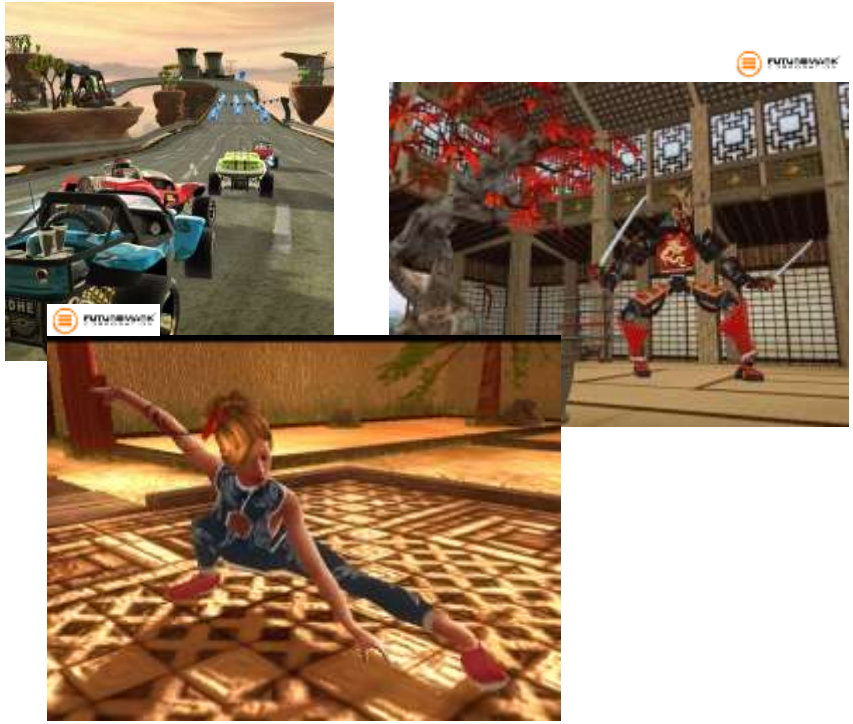
標準2Dベクターグラフィックス
SMAPH-F (OpenVG 1.1)

標準3D/2Dベクターグラフィックス
SMAPH-H (OpenGLES/OpenVG)

OpenGL ES 10周年

これまでの歩み

- **SIGGRAPH 2002**
  - OpenGL ESワーキンググループ立ち上げ
- **SIGGRAPH 2003**
  - OpenGL ES 1.0 API仕様リリース
- **SIGGRAPH 2004**
  - OpenGL ES 1.1 API仕様リリース
- **GDC 2007**
  - OpenGL ES 2.0 API仕様リリース
- **SIGGRAPH 2012**
  - OpenGL ES 3.0 API仕様リリース



OpenGL|ES™

Where it all began..

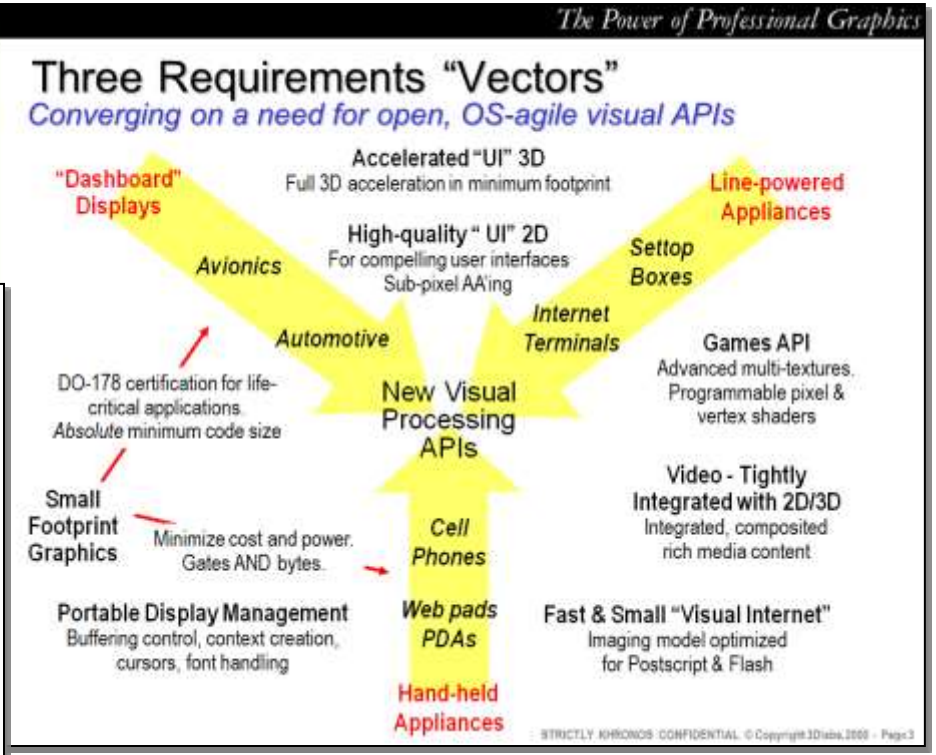
Khronos group

3Dlabs

Embedded Visual APIs
Presentation to Khronos 28Nov00

Neil Trevett & Randi Rost
3Dlabs, Inc

Strictly Khronos Confidential



"The best way to predict the future is to invent it" *Alan Kay*

Where we are today..

Well over 1 BILLION people are using what we all created together - every day…

We are fortunate that the graphics community benefits from an enlightened spirit of cooperation.
"It's the OpenGL Way" **Kurt Akeley**

Fellow Travelers on the journey…

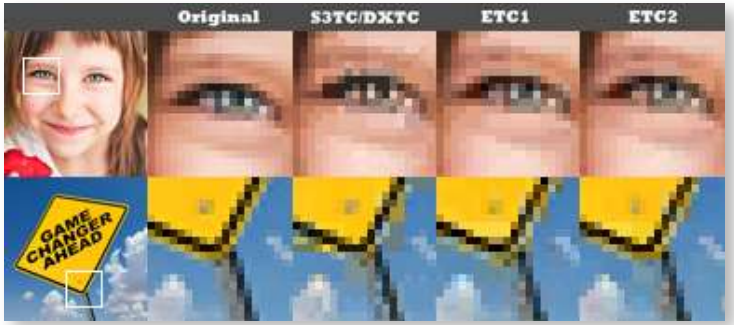
OpenGL ES
Working Group
Austin 2005

OpenGL ES BOF
SIGGRAPH 2007

絵で見るOpenGL ES 3.0

ETC2/EAC テクスチャ圧縮

- ETC2/EACによる高品位なテクスチャ圧縮サポート
- 64のケースで評価
  - 0.8 db higher quality than S3TC/DXTC
  - 1.0 db higher quality than ETC1
  - 1.8 db higher quality than ATC



Compressed Format	Based Internal Internal Format Format	Bits per Texel
COMPRESSED_R11_EAC	RED	4
COMPRESSED_SIGNED_R11_EAC	RED	4
COMPRESSED_RG11_EAC	RG	8
COMPRESSED_SIGNED_RG11_EAC	RG	8
COMPRESSED_RGB8_ETC2	RGB	4
COMPRESSED_SRGB8_ETC2	RGB	4
COMPRESSED_RGB8_PUNCHTHROUGH_ALPHA1_ETC2	RGBA	4
COMPRESSED_SRGB8_PUNCHTHROUGH_ALPHA1_ETC2	RGBA	4
COMPRESSED_RGBA8_ETC2_EAC	RGBA	8
COMPRESSED_SRGB8_ALPHA8_ETC2_EAC	RGBA	8

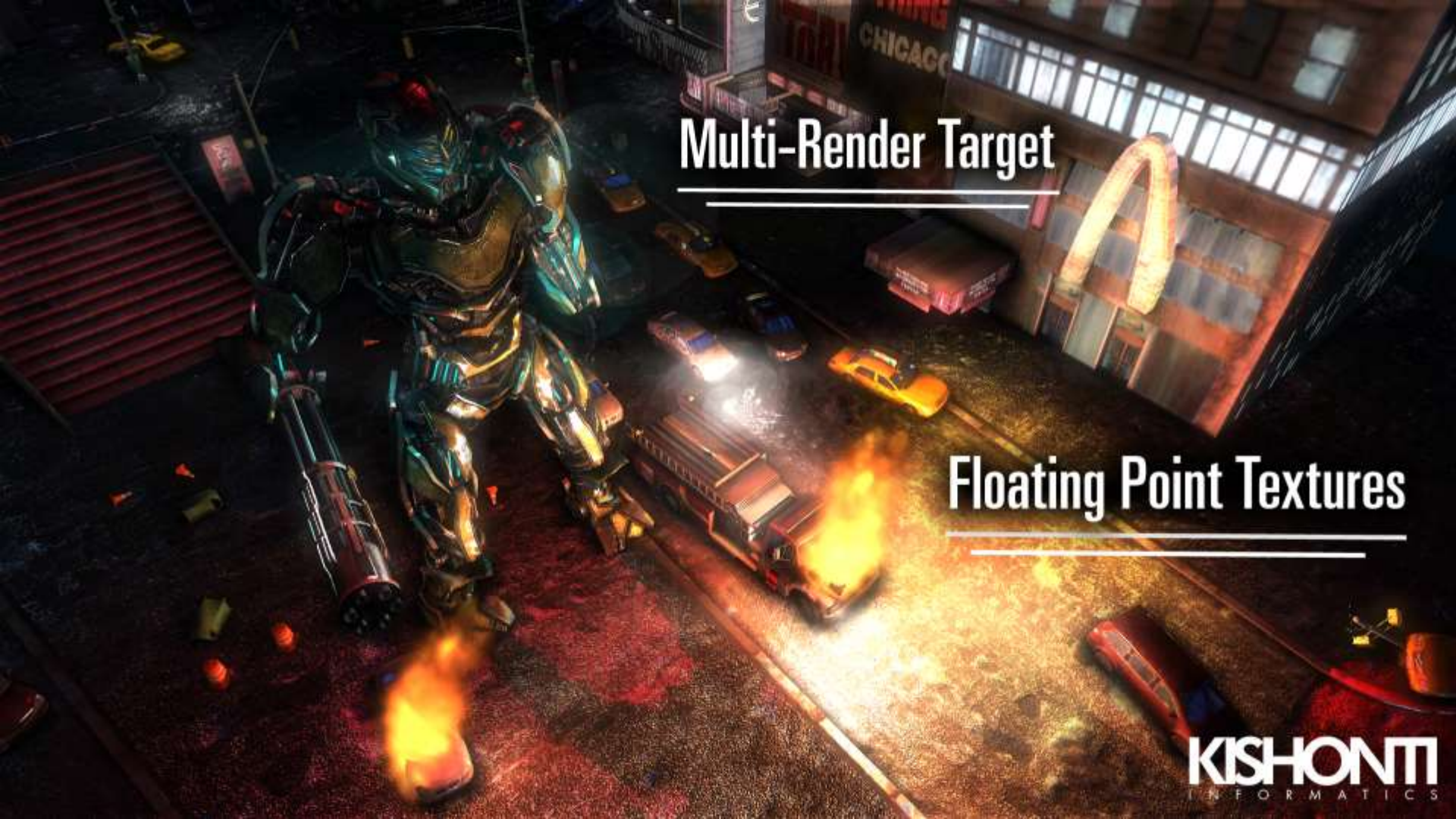
Source: www.Khronos.org

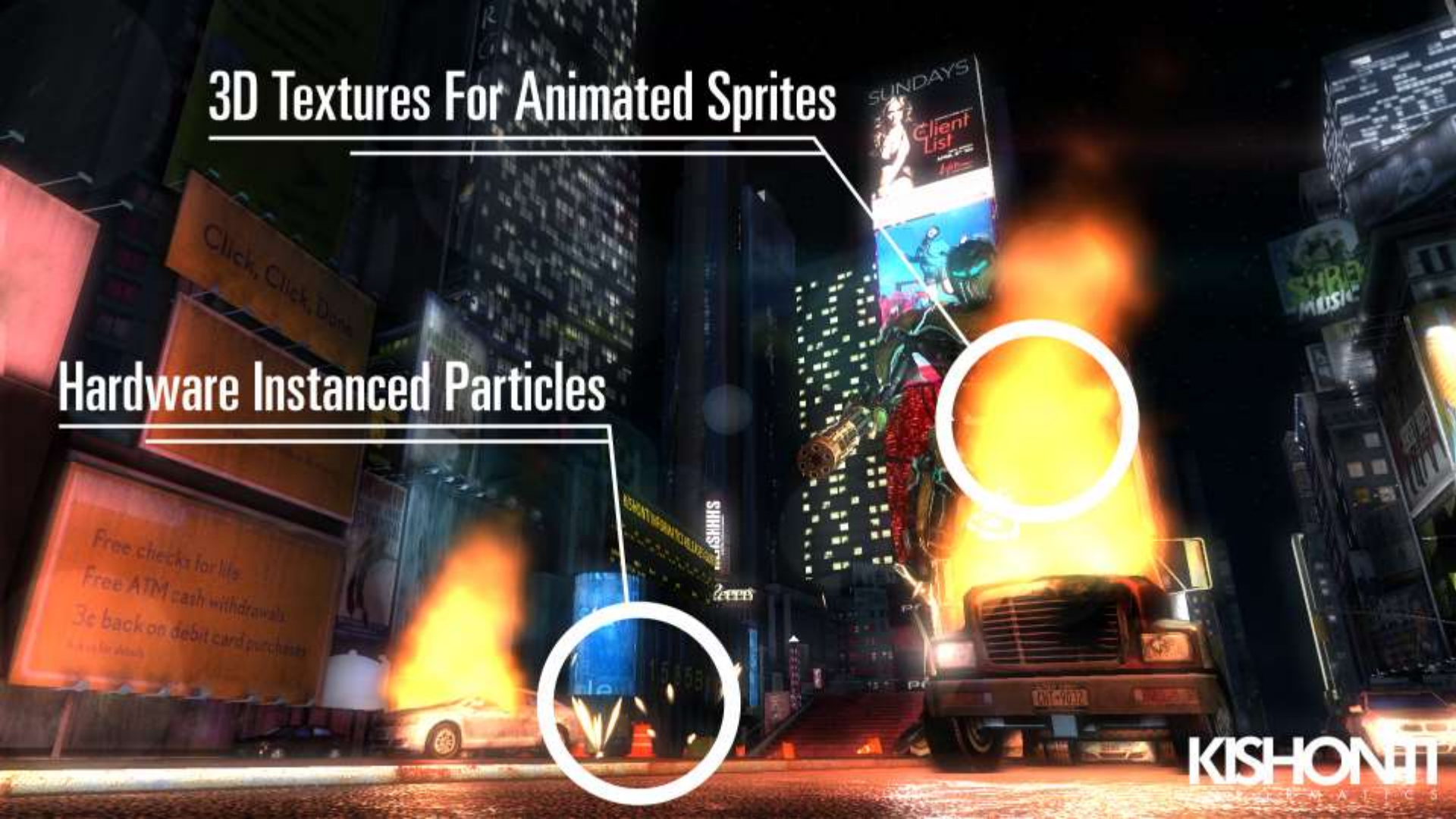
Percentage Closer Filtered (PCF) Shadows

No PCF

PCF

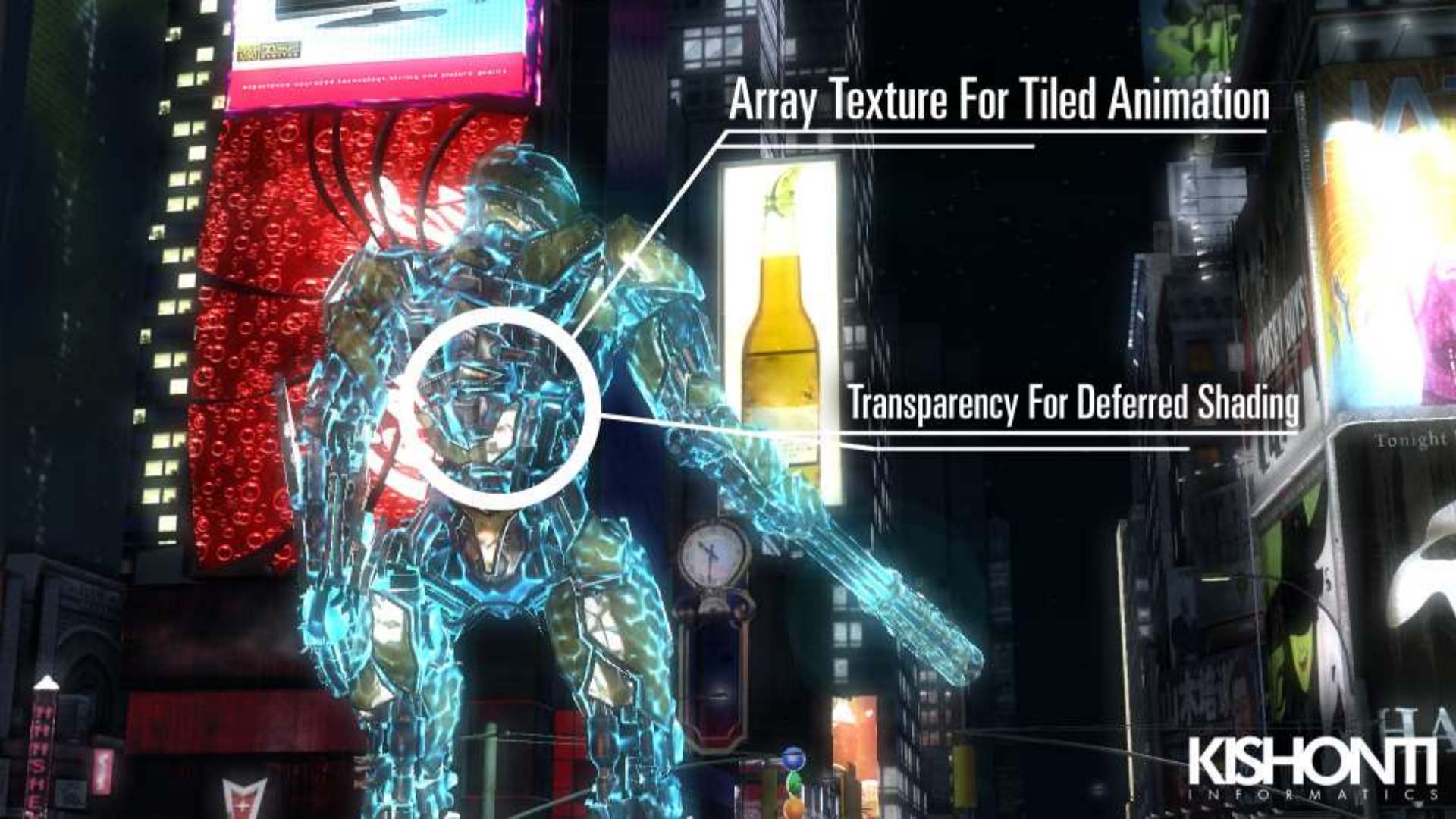
Smoother & Realistic Dynamic Shadows without Ladder Effects


Multi-Render Target
Floating Point Textures
KISHONTI
INFORMATICS


3D Textures For Animated Sprites
Hardware Instanced Particles
SUNDAYS
Client List
Click, Click, Done
Free checks for life
Free ATM cash withdrawals
3c back on debit card purchases
KISHONTI

Geometry Instancing

Render multiple instances of the geometry with a single draw call


Array Texture For Tiled Animation
Transparency For Deferred Shading
KISHONTI
INFORMATICS

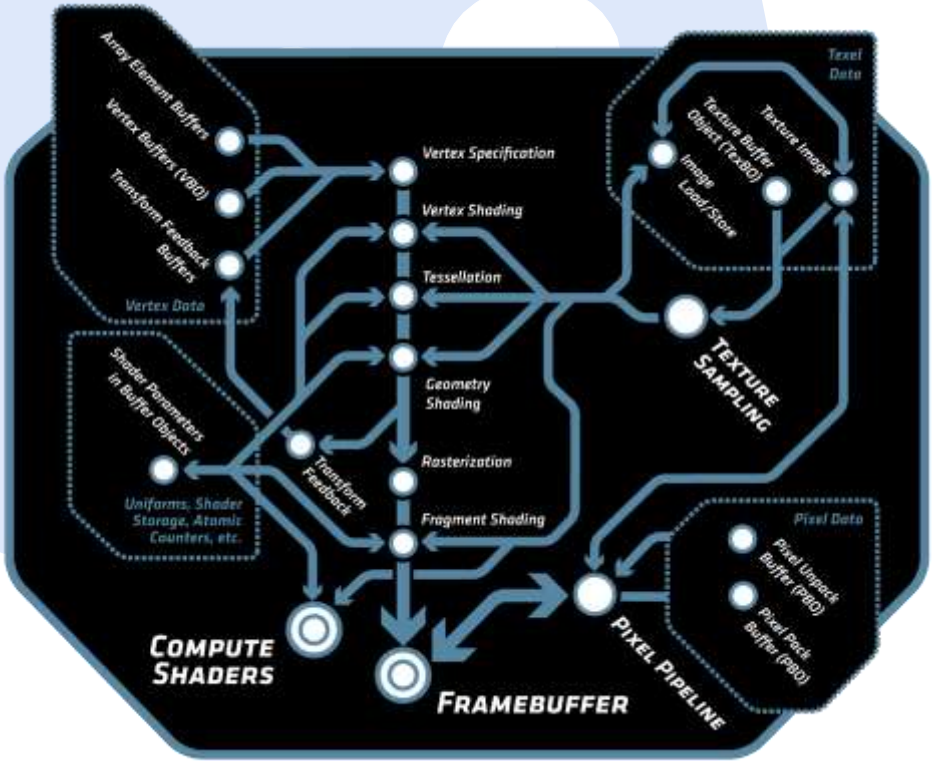
その他重要なOpenGL ES 3.0新機能

- sRGBフォーマット
- Framebuffer Discard
  - tiling architectureなどでシステムメモリにZバッファなどを書き出さないための拡張
- Uniform Buffers
  - シェーダ間でバッファの共有可
- Transform Feedback

OpenGL ES シェーダ言語の主な新機能

- IEEE Floating Point – 32-bit
  - (Also pack and unpack 16-bit)
- integers – 32 bit signed and unsigned
- Appendix A: Limitations – deleted

OpenGLアップデート



OpenGL 4.4

OpenGL 4.4での新機能

- **ARB_buffer_storage**
  - バッファーオブジェクト向け固定ストレージ
    - OpenGL 4.2でテクスチャオブジェクト向けにGL_ARB_texture_storageが追加されたが、それの汎用版
  - バッファ配置・動作について明示的な制御が可能 => ビデオメモリ、システムメモリ、キャッシュ動作
  - 配置済みバッファをGPUが使用可
- **ARB_enhanced_layouts (GLSL)**
  - レイアウト修飾子に関わる拡張
  - コンパイル時コンスタントの使用が可能に
    - const int start = 6;
    - layout(location = start + 2) int vec4 v;
  - シェーダインターフェイス変数の制御のバリエーション追加
  - スカラー型を用いて、ベクトルの効率的なパッキング
  - シェーダ内でのトランスフォームフィードバックバッファに対してoffsetなどの指定
- **ARB_query_buffer_object**
  - クエリ結果をバッファオブジェクト内に格納可能
  - GL APIドライバ経由での取得をおこなわないことでCPU負荷低減

OpenGL 4.4での新機能

- **ARB_clear_texture**
  - 指定された値でテクスチャをクリア
- **ARB_texture_mirror_clamp_to_edge**
  - 負のs,t,r方向でミラー（一回だけの折り返し）のテクスチャ座標をサポート
- **ARB_texture_stencil8**
  - ステンシルのみのテクスチャフォーマットの作成、サンプリング
- **ARB_vertex_type_10f_11f_11f_rev**
  - 10f.11f.11fの32bitパッキング型
- **ARB_multi_bind**
  - マルチバインディングを1回の関数呼び出しで実現（複数コマンドの一本化）
  - CPUの負荷低減

新ARB拡張 (ARB only extension)

- **ARB_bindless_texture**
  - シェーダからハンドラで直接テクスチャ参照が可能（APIを介在しない）
- **ARB_sparse_texture**
  - 物理メモリサイズ以上のテクスチャサイズをサポート
  - 実際に参照する部分を指定して選択
- **ARB_seamless_cubemap_per_texture**
  - キューブマップのテクスチャごとにシームレスの選択制御
- **ARB_indirect_parameters**
  - マルチなindirect drawにおけるデータ量（Transform feedback後のプリミティブ数など）のカウント数などを格納したパラメータバッファの生成

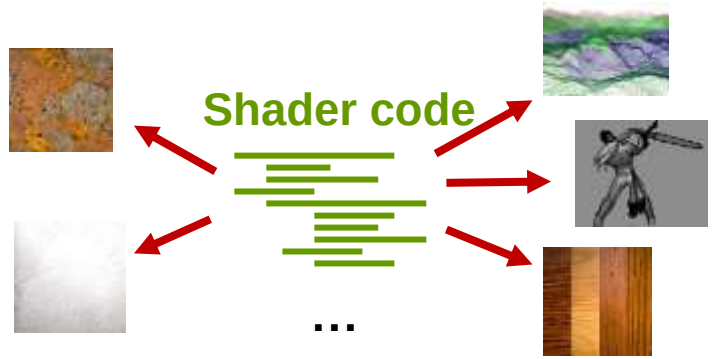
新ARB拡張 (ARB only extension)

- **ARB_compute_variable_group_size**
  - コンピュートシェーダディスパッチ向けワークグループのサイズ指定
- **ARB_shader_draw_parameters**
  - 新GLSLビルトイン: `gl_BaseInstance`, `gl_BaseVertex` and `gl_DrawID`
- **ARB_shader_group_vote**
  - 複数シェーダ実行結果のフラグについてブール計算を実施
  - パイプラインストールを低減

Bindless Textures

- 実行時必要としているテクスチャ数の増加
- 1シーン内で、よりリッチかつ多くのテクスチャ、マテリアルを使用されてきている

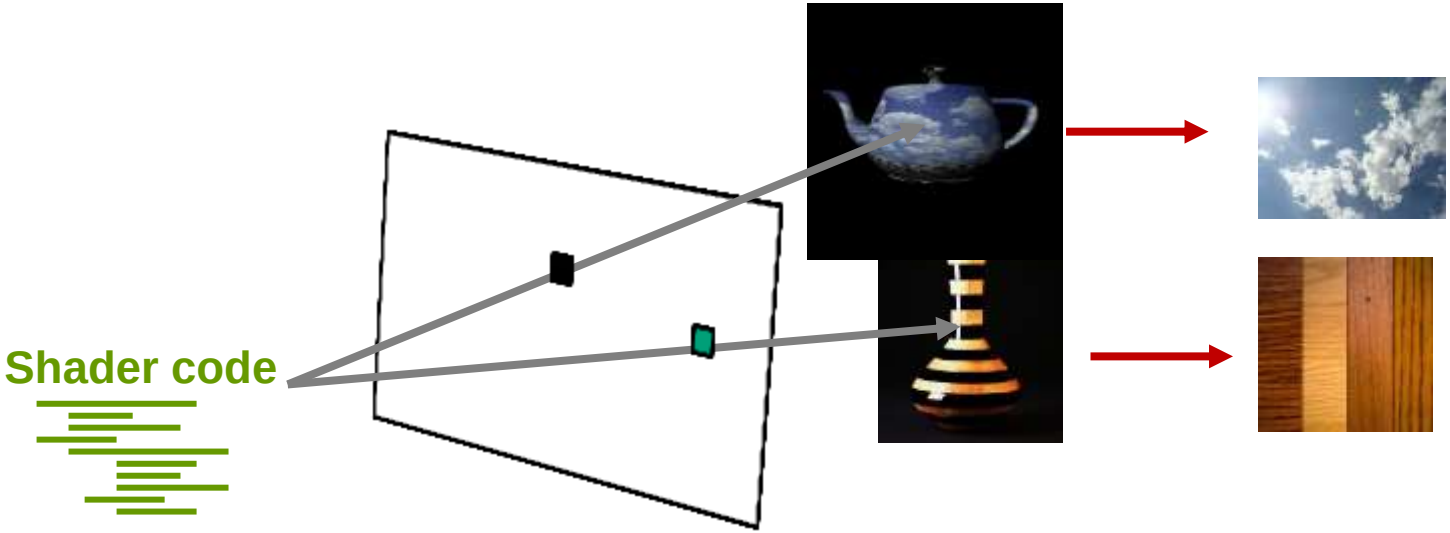
従来のテクスチャバインディングモデル



バインドレステクスチャ
1M枚以上のユニークテクスチャ使用

Bindless Textures

- レイトレやアドバンスなレンダリングを行った場合も、テクスチャロード数ネックになることを防げる可能性



Bindless Textures

従来のテクスチャバインド

CPU

Load texture A
Load texture B
Load texture C
Bind texture A to slot I
Bind texture B to slot J
Draw()

GPU

Read from texture at slot I
Read from texture at slot J

CPU

Bind texture C to slot K
Draw()

GPU

Read from texture at slot K

バインドレステクスチャ

CPU

Load textures A, B, C
Draw()

GPU

Read from texture A
Read from texture B
Read from texture C

CPUの負荷低減により、GPU性能向上を目指せる

graphics library Transmission Format

3Dアセット向け標準ストリームフォーマット

- **3Dアセット向け圧縮、ストリームフォーマットの必要性**
  - モバイル、ネット接続のデバイスで、大きなアセットを扱うニーズの増加
- **3Dは唯一未定義なストリームメディアタイプ**
  - 複雑なデータセット：アセットタイプ・ユースケースの多様性
- **ロイヤリティフリー**
  - ネットビデオフォーマット戦争のシナリオを避けたい
- **最終的にハードウェア実装が可能**
  - 高性能かつ低消費電力の両立

Audio	Video	Images	3D
MP3	H.264	JPEG	?
napster	YouTube	facebook	!

glTFの目標

- **3Dアセット配信向け効率的なバイナリフォーマット**
  - ネットワークバンド幅の低減及び、クライアントでの処理オーバーヘッドの低減
- **ランタイムニュートラル**
  - アプリ、ランタイム(WebGLを想定)のいずれでも使える
- **スケーラブルな圧縮・非圧縮ストリームのハンドリング**
  - Baseline (profile)では圧縮を含まない
- **WebGLでの効率的なダイレクトロード**
  - WebGLクライアントではほとんど処理せずglTFの読み込みをしたい
- **オーサリングフォーマットからのデータ変換**
  - プロトタイプ及び最適化をCOLLADAアセットからの変換で実施

COLLADA / glTF エコシステム

OpenCOLLADA
Importer/Exporter
COLLADA
コンフォーマンステスト

ツール間やりとり

COLLADA

ISO

COLLADA2GLTF
変換

その他オーサリングフォーマット

glTF

Web-based
Tools

Autodesk Maya

blender

各種WebGL展開

WebGL

OpenGL

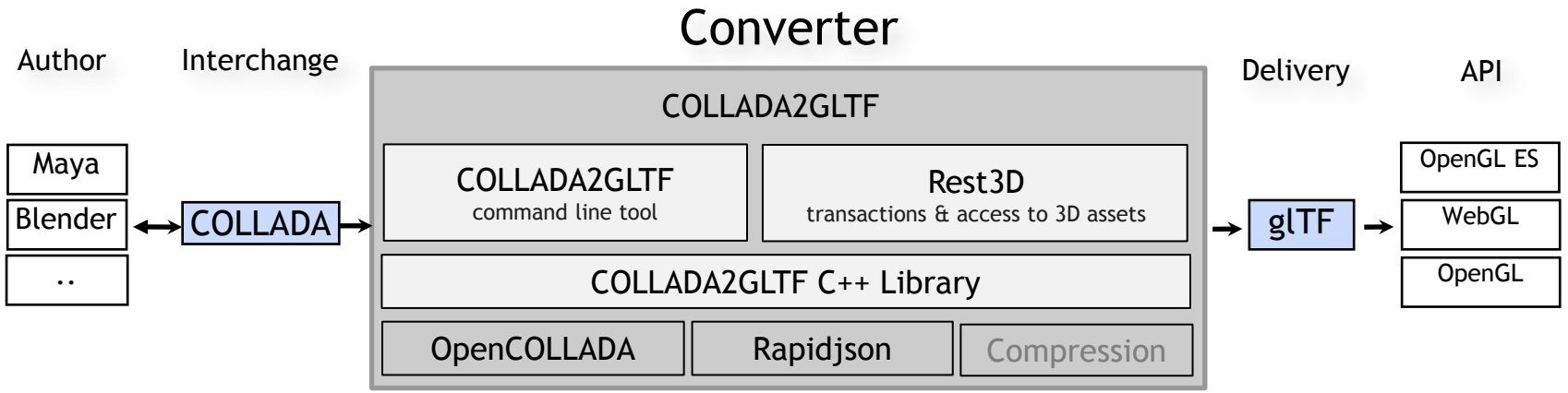
OpenGL|ES

Three.js glTF Importer.
Rest3D initiative

glTFの現状

- COLLADAワーキンググループによるイニシアティブ
- 仕様策定中 - 正式な仕様書はまだ
- コード、仕様策定を同時に進めている
- 現在、最新の議論、コミニティからのフォードバックを仕様書・コンバータ実装に反映をしている。
- フィードバックはGitHubから受付中

glTF Pipeline - overview



glTF Pipeline - COLLADA2GLTF

- 必須の配信プリミティブ
  - Triangulate
  - Unify indices
  - Split meshes
  - Generate shaders
  - Axis up conversion
- 計画中の最適化、配信プリミティブ
  - Minify shaders
  - Flatten hierarchy & combine meshes
  - Convert images
  - Compress meshes
  - Interleave vertices

  ... 提案を受付中

スコープ

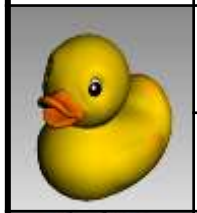
- 1.0
  - Node hierarchy
  - Cameras
  - Lights
  - Key-framed animations
  - Skinning
  - Morphing
  - Medias (images & videos)
- メッシュ圧縮（拡張）
- マルチパスサポートの検討

Open3DGC

- Open3DGCバイナリはMPEG-SC3DMC (Scalable Compression 3D Mesh Compression)による実装
- 参考論文：K. Mammou, T. Zaharia, F. Prêteux, “TFAN: A low complexity 3D mesh compression algorithm” Computer Animation and Virtual Worlds, Vol. 20(2-3), pp. 343-354, 2009
- 2つのモード：バイナリ及びアスキー
  - Open3DGCバイナリは算術ベースエンコードを適用
  - Open3DGCアスキーはGZIP圧縮のため7bitアスキーエンコーディングを適用
    - Open3DGCアスキー自体の圧縮はGZIPをそのまま利用

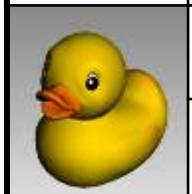
圧縮

- 2つのライブラリでテスト:
  - Open3DGC: https://github.com/amd/rest3d/tree/master/server/o3dgc
  - webgl-loader: https://code.google.com/p/webgl-loader/

Model	COLLADA		glTF		glTF+Open3DGC ascii		glTF+Open3DGC binary	
	XML	gzip	raw	gzip	raw	gzip	raw	.raw bin .gzip JSON
			. bin:102k . JSON:11k	. bin:81k . JSON:2kb	. ascii:29k . JSON:11k	. ascii:19k . JSON:2k	. bin:18k . JSON:11k	. bin:18k . JSON:2k
	336k	106k	113k	83k	40k	21k	29k	20k
			. bin:9220k . JSON:75k	. bin:3220k . JSON:5k	. ascii:3080k . JSON:151k	. ascii:1510k . JSON:11k	. bin: 1622k . JSON:151k	. bin: 1622k . JSON:11k
	19767k	3417k	9295k	3225k	3231k	1521k	1773k	1633k
			. bin:25224k . JSON:183k	. bin:5738k . JSON:8k	. ascii:7793k . JSON:587k	. ascii:1433k . JSON:29k	. bin:3205k . JSON:589k	. bin:3205k . JSON:29k
	56763k	7378k	25407k	5746k	8380k	1462k	3794k	3234k
			. bin:329k . JSON:255k	. bin:99k . JSON:10k	. ascii:122k . JSON:267k	. ascii:61k . JSON:11k	. bin:71k . JSON:267k	. bin:71k . JSON:11k
	794k	133k	584k	109k	389k	77k	338k	88k

“webgl-loader” での圧縮

- Encoder in C++/Decoder in JS
- Fast decoding

Model	COLLADA		glTF+webgl-loader		glTF+Open3DGC ascii		glTF+Open3DGC binary	
	XML	gzip	raw	gzip	raw	gzip	raw	.raw bin .gzip JSON
			. utf8:42k . JSON:12k	. utf8:34k .JSON:2kb	. ascii:29k . JSON:11k	. ascii:19k . JSON:2k	. bin:18k . JSON:11k	. bin:18k . JSON:2k
	336k	106k	54k	36k	40k	21k	29k	20k
			.utf8:8747k . JSON:753k	.utf8:1325k . JSON:29k	. ascii:7793k . JSON:587k	. ascii:1433k . JSON:29k	. bin:3205k . JSON:589k	. bin:3205k . JSON:29k
	56763k	7378k	9500k	1354k	8380k	1462k	3794k	3234k

圧縮: 結果まとめ

	Total Size (Mbytes)	Compression Gain (%)
OBJ	377	0%
GZipped OBJ	106	72%
WebGL Loader UTF8	76	80%
Open3DGC-ASCII7	36	91%
Gzipped WebGL Loader UTF8	27	93%
OpenCTM	23	94%
GZipped Open3DGC-ASCII7	15	96%
Open3DGC-Binary	15	96%

Data from rest3d SIGGRAPH 2013 BOF presentation, Khaled Mammou/Remi Arnaud

圧縮: 今後の予定

- 実装上の修正
  - 現在複数インデックスバッファで頂点バッファがシェア出来ていない（同じ頂点バッファをコピー）
- デコード性能及びメモリ使用量のベンチマーク
- アニメーション、ボーンインデックス・重みの圧縮

glTFリソース

- **Specification**
  - https://github.com/KhronosGroup/glTF/blob/master/specification/README.md
- **Converter**
  - https://github.com/KhronosGroup/glTF/converter
- **montagejs viewer**
  - https://github.com/fabrobinet/glTF-webgl-viewer

Three.js

- **Three.js はもっともメジャーなWebGL開発ライブラリ**
  - ハイレベルシーングラフAPI でWebGL開発が容易に可能
  - 高性能かつ高機能
  - 多くの有名なWebGLアプリケーション、デモで使われている
  - 多くアクティブな参加者; 6000+ commits, 2500+ forks, 12,000 favorites
  - https://github.com/mrdoob/three.js/
- **Three.jsのコンテンツパイプラインは確立していない**
  - シンプルな3Dアセットフォーマットをサポートしているが (.OBJ, .STL)、制約が多い (e.g. no cameras, lights, scenes, animations…)
  - COLLADAもサポートしているけどローダが古くかつ、バギー
  - Three.jsは独自フォーマットもサポートしているが、アドホックかつ、非標準
  - 結果、開発者常に独自パイプライン、急ごしらえのフォーマットを構築し、レイアウト、ライティング、アニメーションはプログラマによるハンドコーディング

Three.js向けglTFローダー

- **glTFは理想的な完全かつオープンかつ業界標準かつ近代的なフォーマット**
  - 最近の3Dアプリ使われる機能を網羅: scenes; cameras; lights; animations (articulated, skinned, morphs); shaders; multi-pass techniques.
  - JSON+バイナリベースアプローチはとってもウェブフレンドリー
  - コンパクトなフォーマットによりCOLLADAに比べロードタイム短縮、小さなメモリフットプリントを期待
  - COLLADA2GLTFコンバータによって既存COLLADAコンテンツからの変換パスを提供

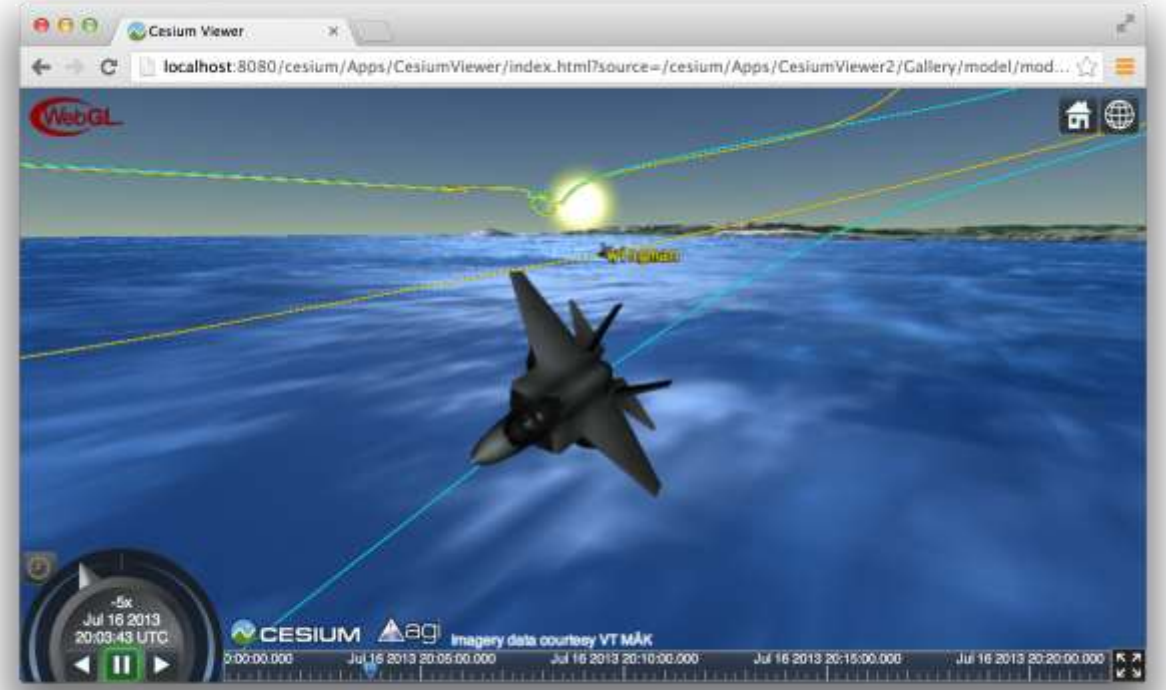
プロジェクト

- glTFプロジェクトブランチ on Github
  - https://github.com/KhronosGroup/glTF
  - Branch ‘threejsloader’
- まずは既存OBJ, COLLADAサンプルの変換が目標
  - e.g. http://threejs.org/examples/webgl_loader_collada.html
- 最初の実装完了は今年8月をターゲット
- 実装後Githubにコミット予定

Status

- ほとんどのglTF機能をサポート済み
  - Triangle Meshes
  - Materials – diffuse, specular, emissive, ambient, textures, environment maps
  - Cameras – perspective and orthographic
  - Lights – ambient, spot, point, directional
  - Matrix transforms
  - Scene structure
  - Animation
    - Key frame/articulated only
  - Shaders
    - Uses “common profile” techniques - common lighting models such as Phong and Lambert, and their parameters, are mapped to existing Three.js material types
- 今後実装する機能
  - Arbitrary GLSL shaders (via Three.js ShaderMaterial)
  - Skinned animations and morphs

glTF in Cesium



http://cesium.agi.com/

Thank you!