

SOFTIMAGE®



SOFTIMAGE が提唱する次世代の
フェイシャル表現とゲーム開発環境

アビッドテクノロジー株式会社

アビッドテクノロジー株式会社

ソフトイメージ
アプリケーションエンジニア
渡辺 揮之

アビッドテクノロジー株式会社

本社
ジャパン
設立
納入国

米国マサチューセッツ州テュークスベリー市
東京都港区赤坂2-11-7
1987年(1993年にNASDAQに上場)
約100ヶ国

Avid
computer graphics



-Avid DS Nitris
-Avid Media Composer
-Avid Adrenaline
-Avid Xpress Pro

SOFTIMAGE[®]



-SOFTIMAGE|XSI
-SOFTIMAGE|FACE ROBOT

Alienbrain[®]



-Alienbrain studio

本日のアジェンダ

- SOFTIMAGE|XSIのGATORを利用した非破壊の制作プロセス（20分）
- SOFTIMAGE|FACE ROBOTによる次世代のフェイシャル表現（40分）
- v.5.11 での SDK の新機能、リアルタイムシェーダ更新情報、COLLADA の入出力（30分）

SOFTIMAGE®



GATORを利用した非破壊の制作プロセス

非破壊、ノンリニアのデータ構造が もたらす生産性

- テクスチャを張った後にモデリングの修正
- エンベロープ後にモデリングの修正
- エンベロープ後にボーンの追加、削除
- アニメーション後にリグの修正
- シェープアニメーション後にモデリング修正
- オリジナルのアニメーションを壊さずに修正
- 実機と同じ挙動をXSI上で再現



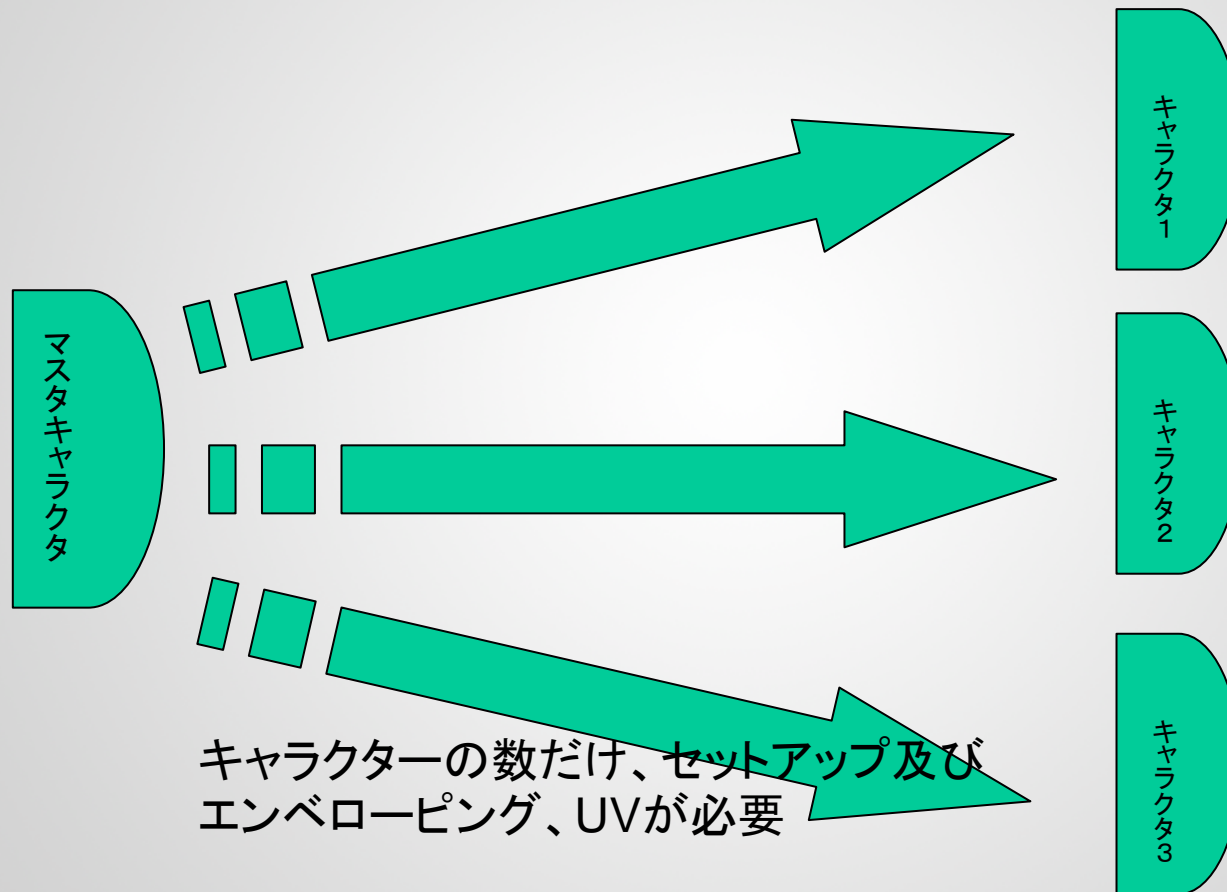
GATOR

(Generalized Attribute Transfer)

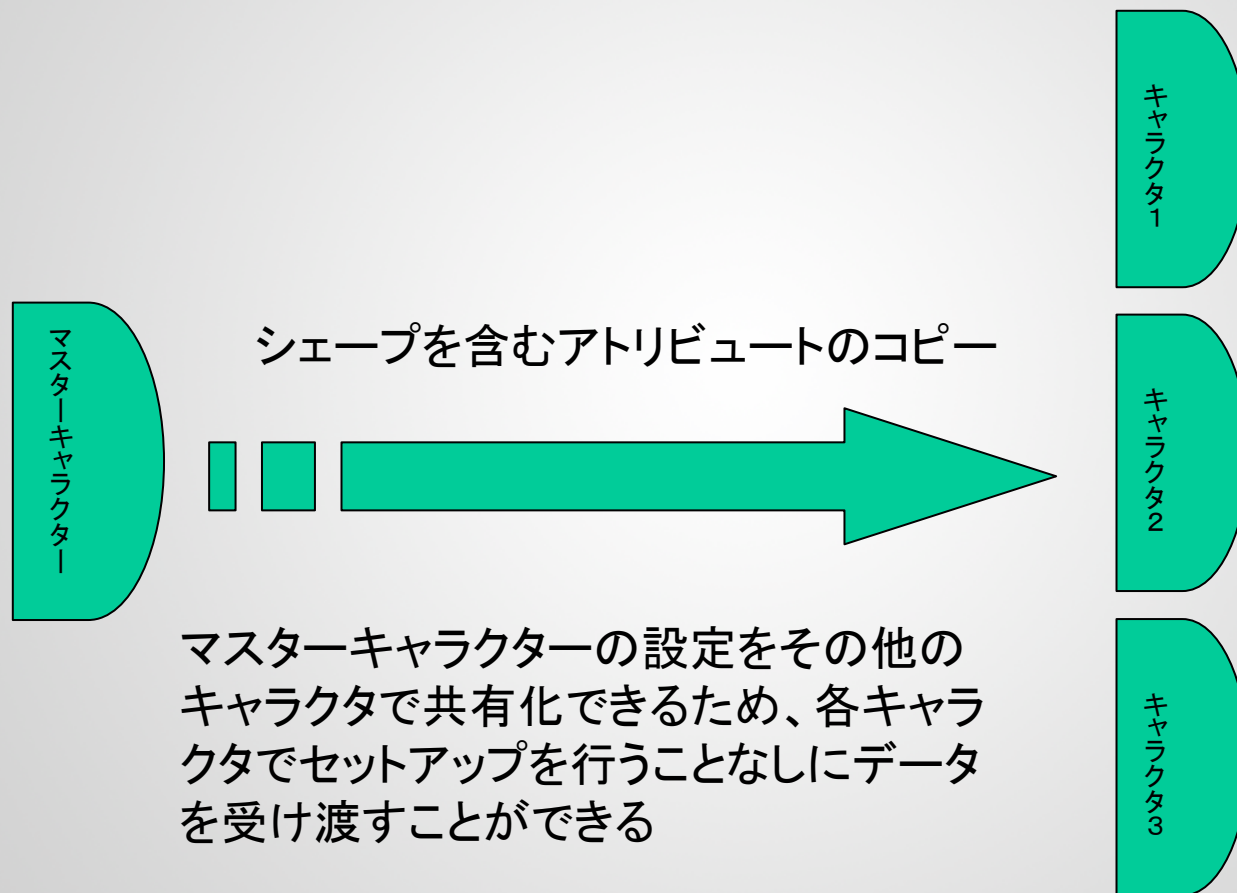
非破壊、ノンリニアプロセスの進化系

- UV、テクスチャ、マテリアル、エンベロープ、シェイプアニメーションのコピー
- トポロジーが異なるオブジェクト間でもコピーが可能
- オブジェクトの数が異なっても対応可能

従来のキャラクターアニメーション



GATORによるキャラクターアニメーション



XSIのGATOR デモンストレーション

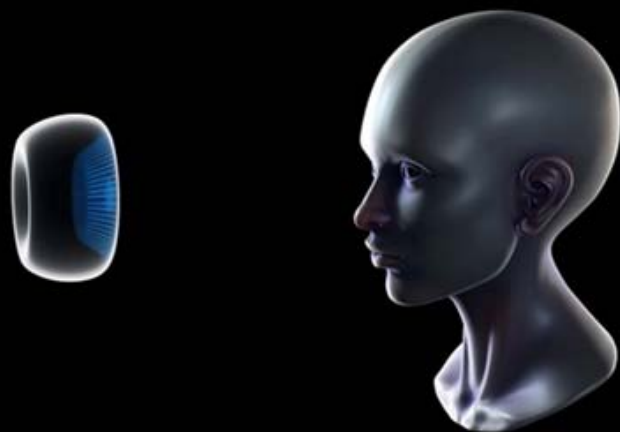
SOFTIMAGE®

FACE ROBOT

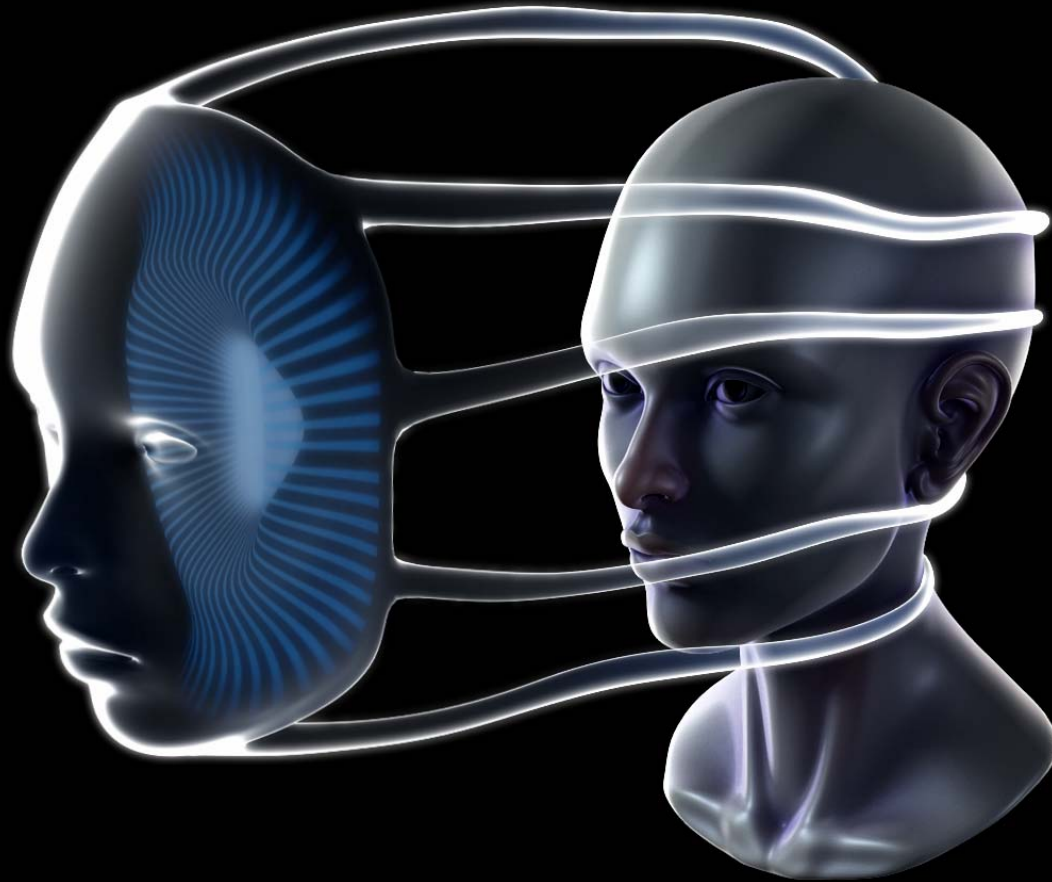
FACE ROBOT 概要説明

FACE ROBOTとは？

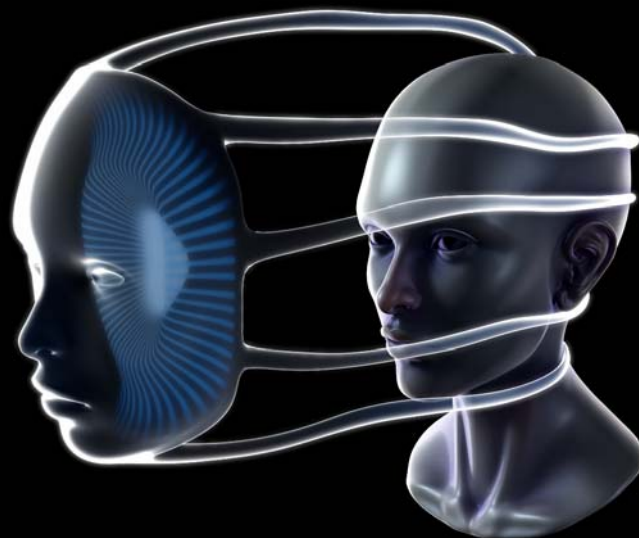
アニメーションを行う際に有機的な 肌の動きを作り出せるツール



ソルバー

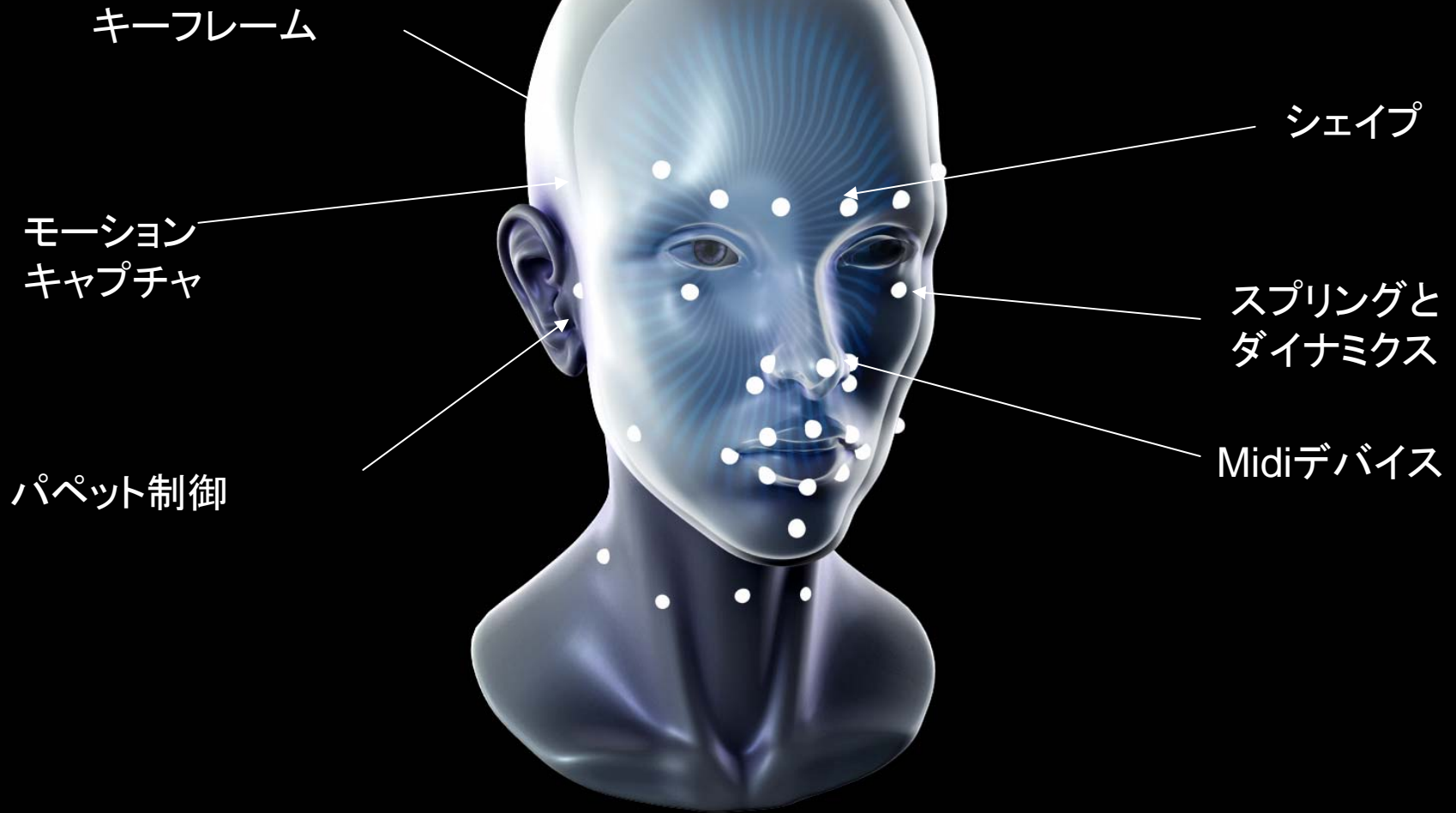


ソルバー



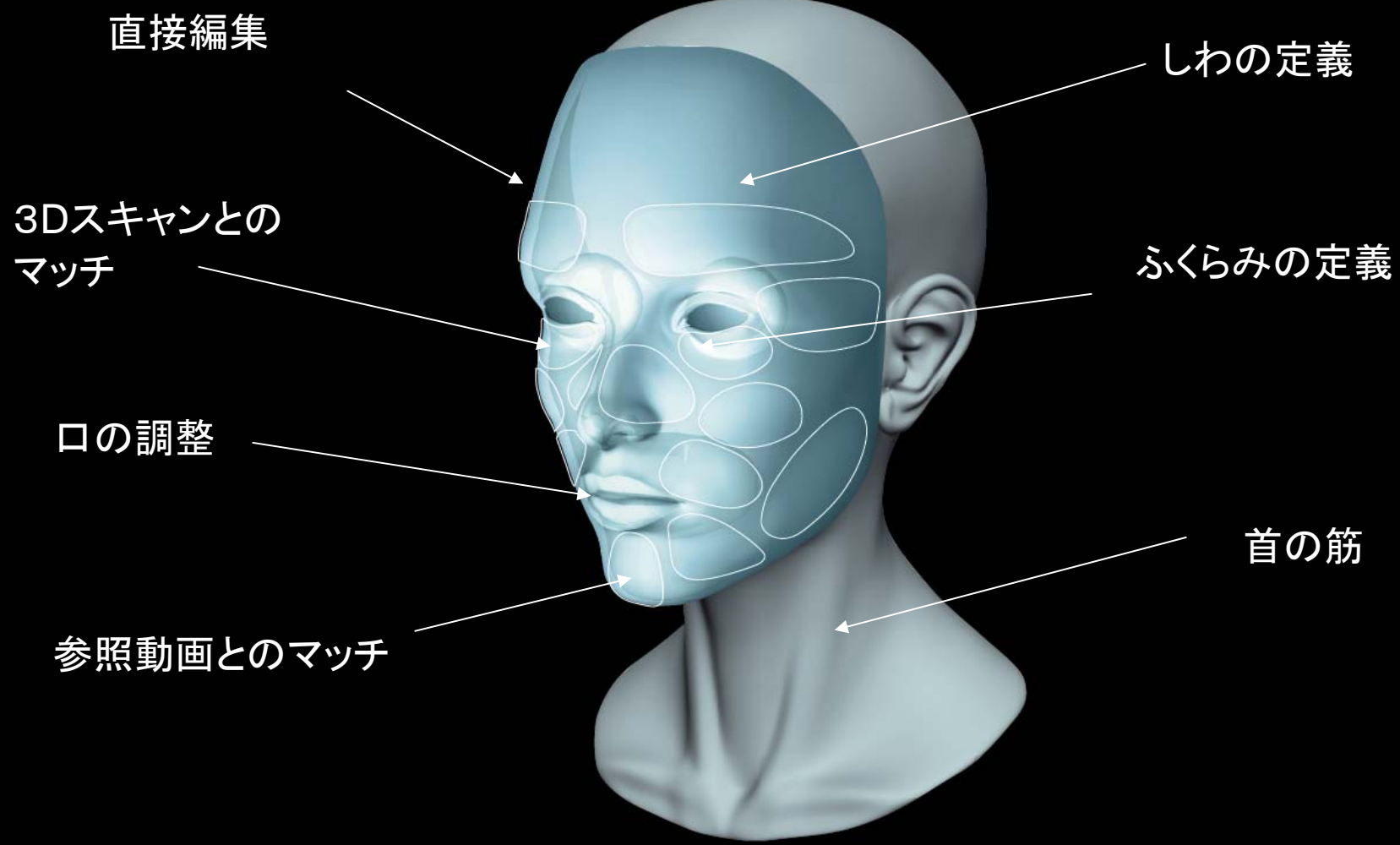
- フェイスに接続
- フェイス動作への準備
- フェイスを変形

フェイスをアニメーション

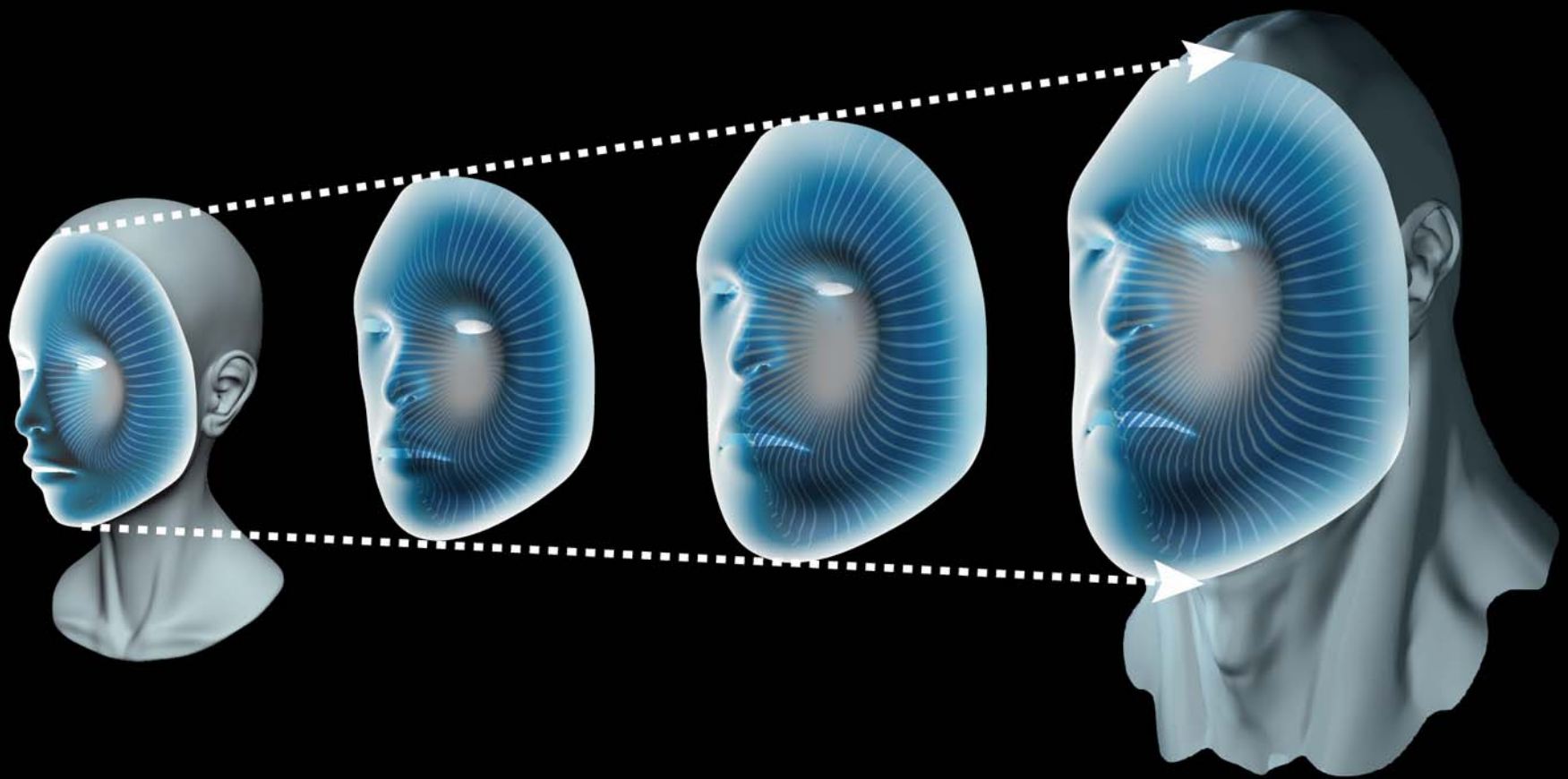


オブジェクトを稼動するにはどの方法にも有効

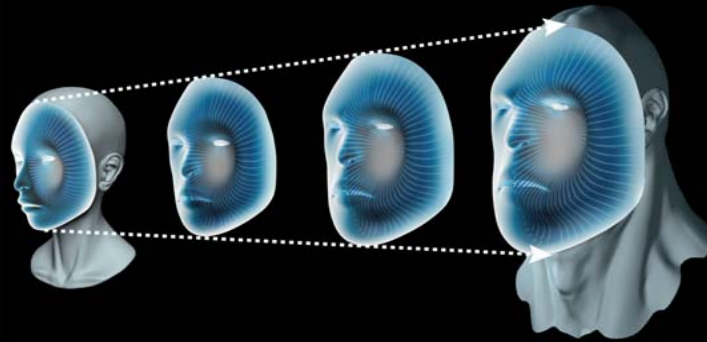
TDがスキンの挙動を調整



フェイシャル・リターゲティング

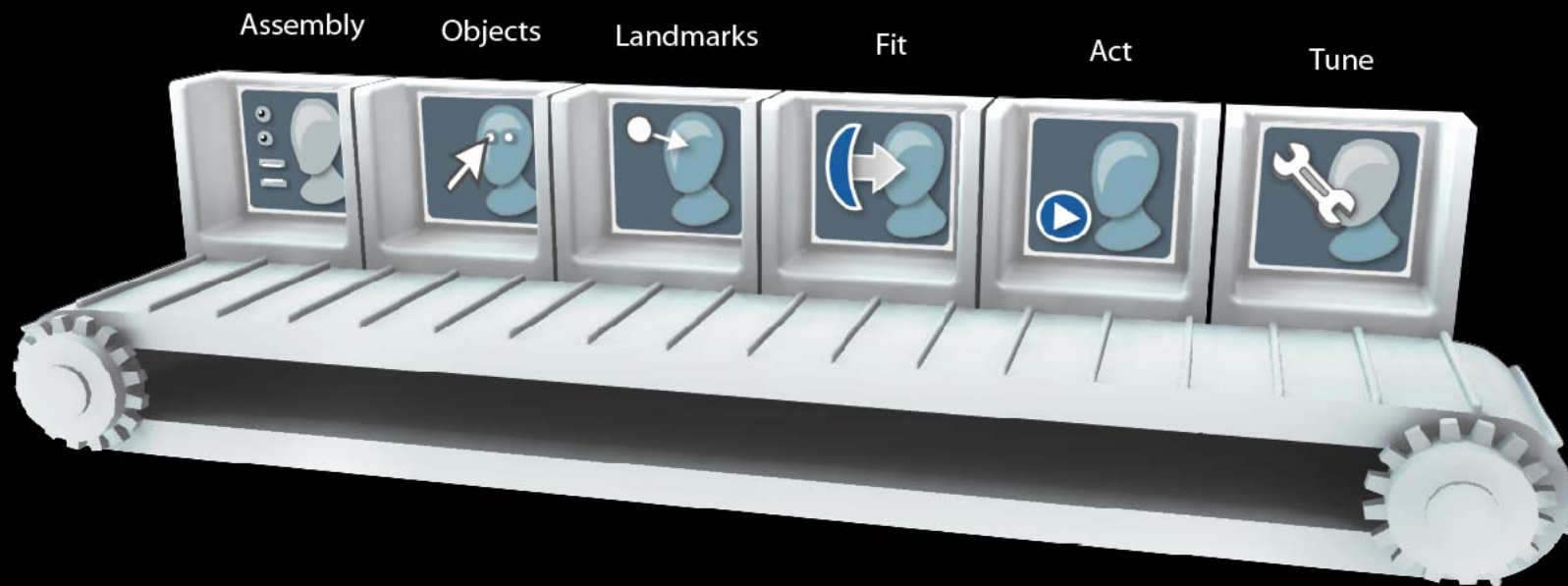


フェイシャル・リターゲティング



- モーションの転送
- ソルバーの整列
- パフォーマンスを変化させるための調整

フェイス・ファクトリー



フェイス・ファクトリー



- フェイスパイプラインの自動化、一般化
- 焦点を絞った調整環境
- シンプルな処理ステップ
- 次のアクションが予期可能
- 明瞭なプロセスのゴール

製品

Animator



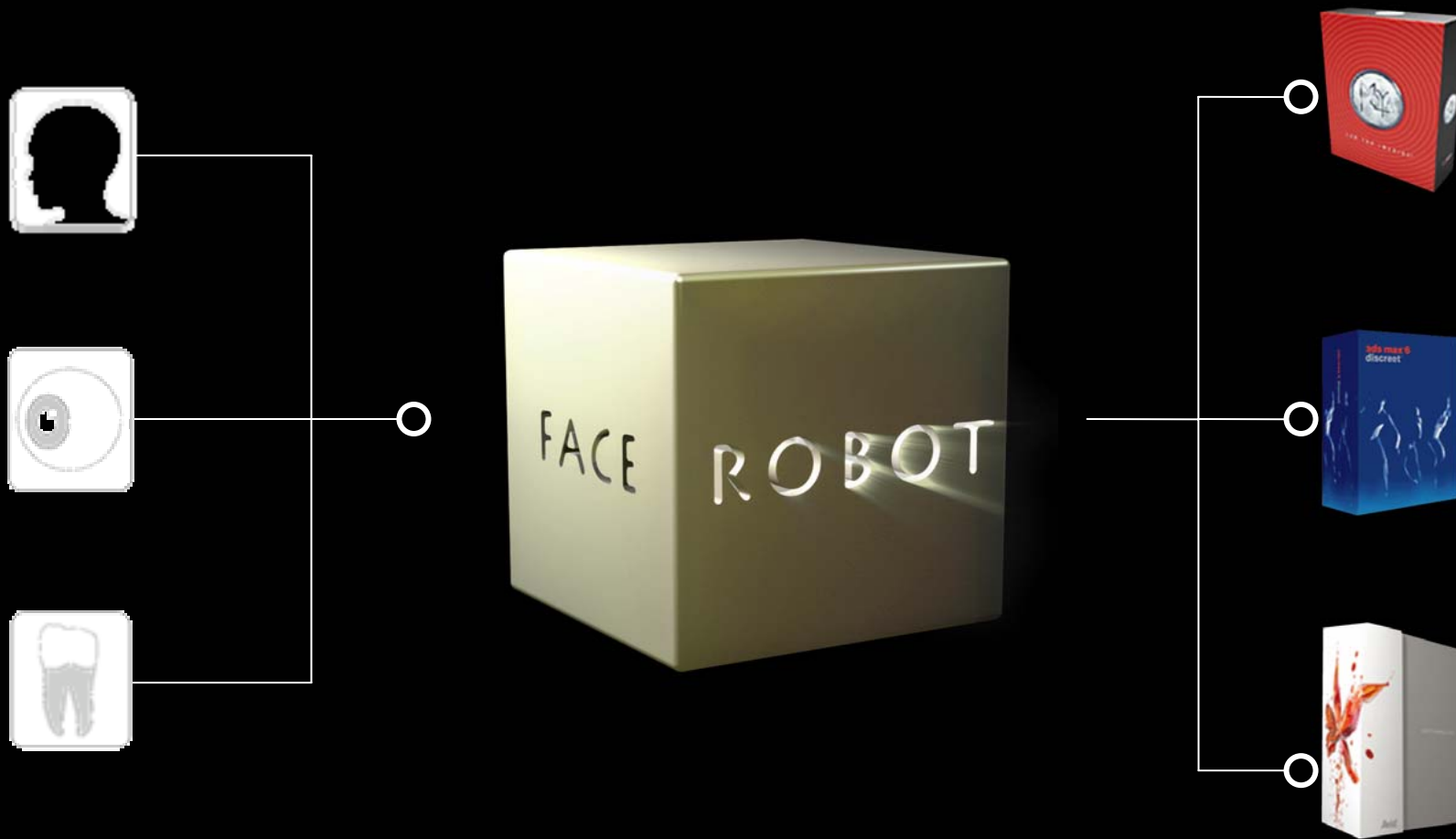
フェイシャルアニメーション
が可能なツール

Designer

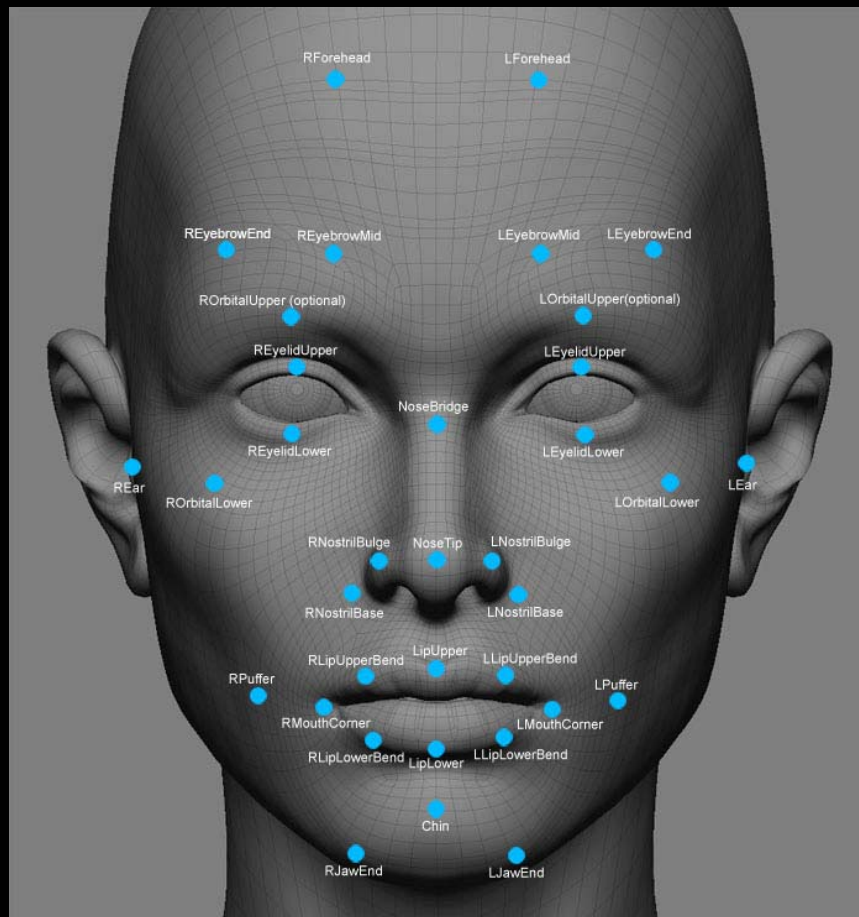


フェイスのソルバー適応と
アニメーションが可能なツール

フェイスツール



フェイシャルキャプチャ



Old Kitty

- スペシャルプロジェクト & Pixologic社
(Dilip Singh & Ryan Kingslien)
- しわと老い
- ストレスマップ

FACE ROBOT デモンストレーション

SOFTIMAGE®

FACE ROBOT

FACE ROBOT
ゲームエクスポートへの対応

Face Robot

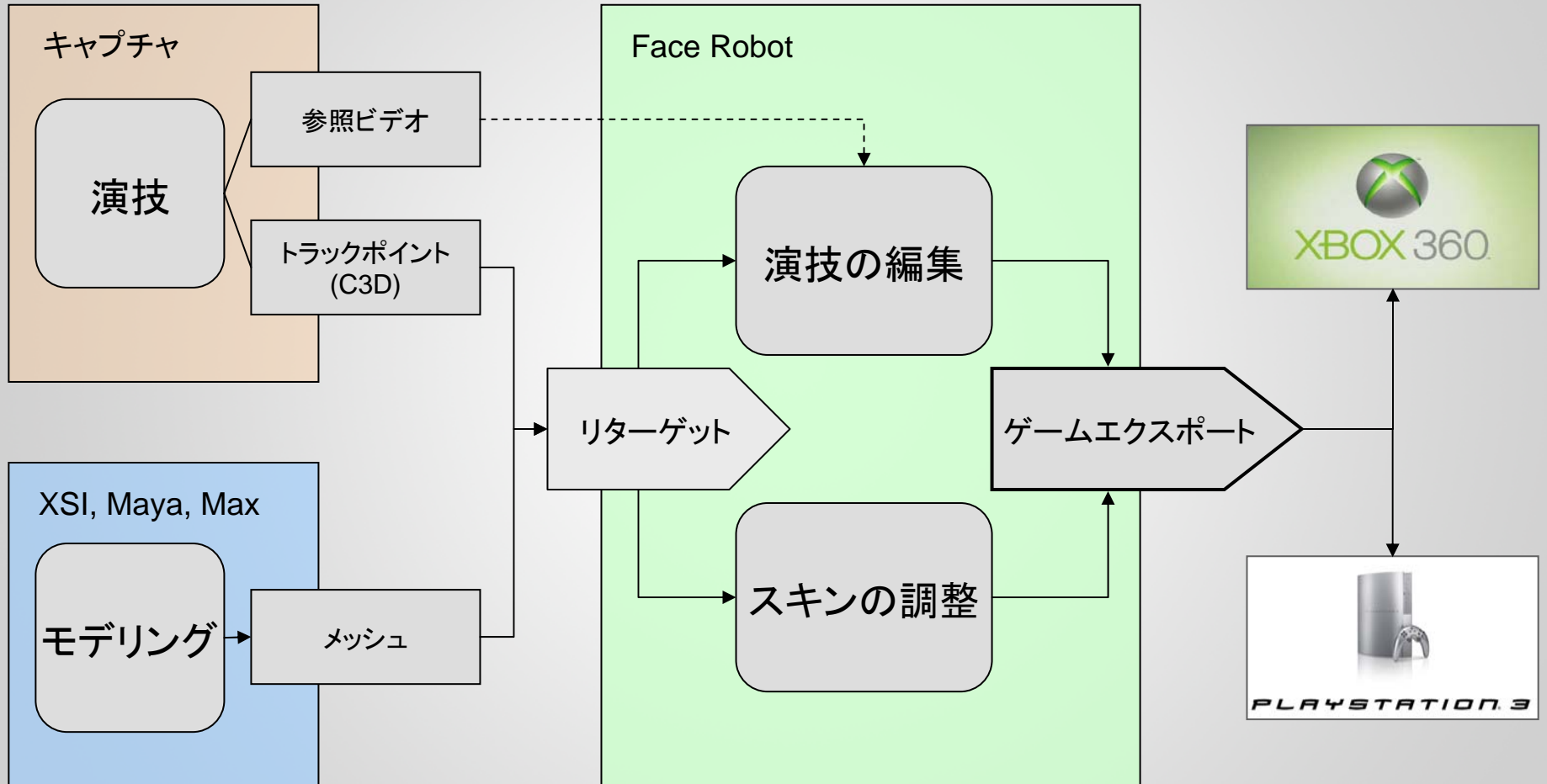
ゲームエクスポートパイプライン



開発の思想

- 生きているかのような豊かな表情をもつキャラクターをゲーム上で再現する
- 主要ゲームプラットフォームをサポート
- エンベロープや法線マップのといった基本技術を用い再現
- アーティストによる制御が可能である

Face Robot パイプライン



ゲームエクスポート ソリューション

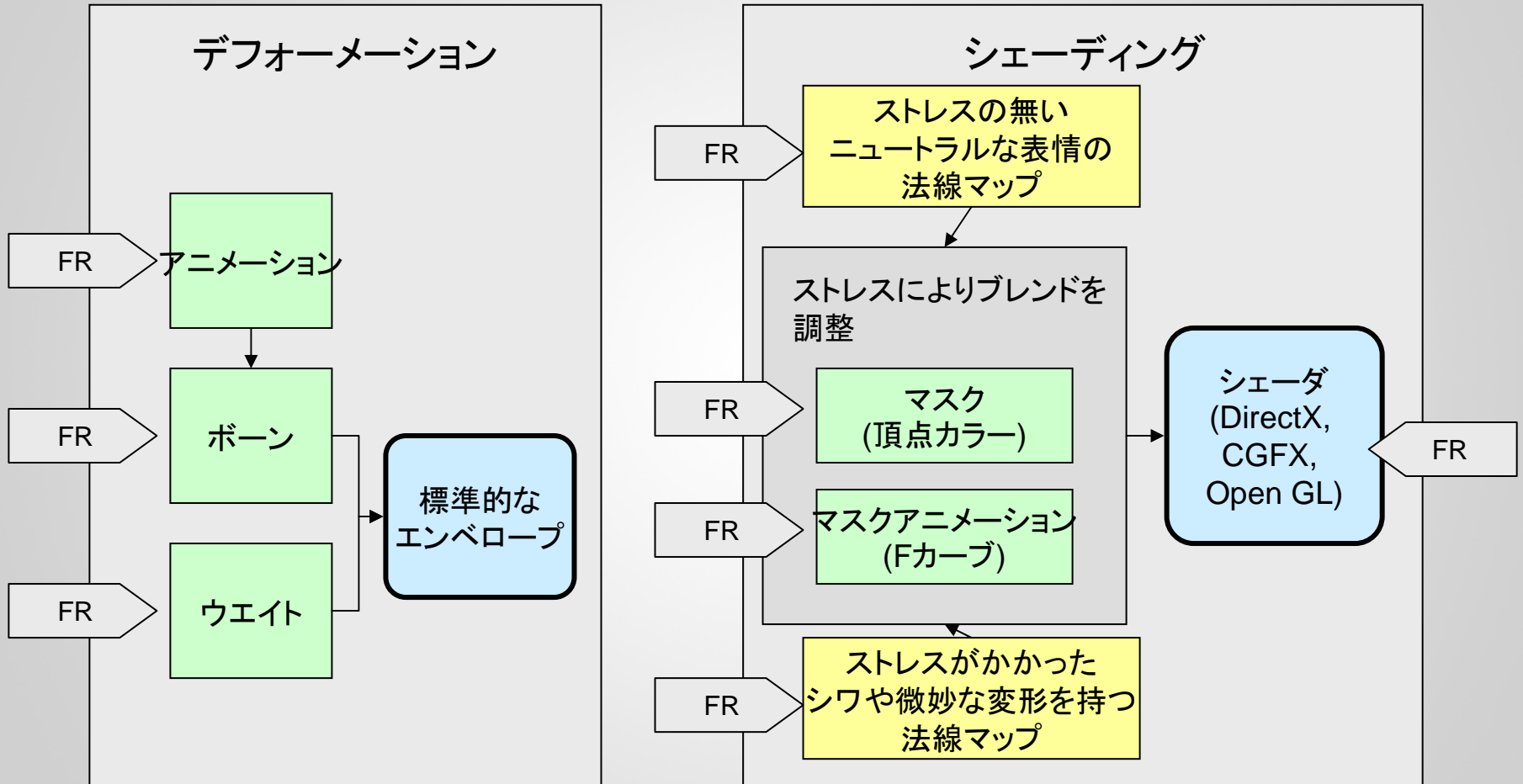
デフォメーション

- 顔の動きをランタイム上に変換
- 基本技術であるボーンとエンベロープウェイトによるアニメーション

シェーディング

- シワや肌の微妙なニュアンスをリアルタイムシェーダで再現
- ストレス個所をもとにマップをブレンド

ランタイムプロセス



アドバンテージ

クオリティ

- 微妙なニュアンスすらランタイム上で再現

スピード

- 簡潔なエクスポートのプロセス

最適化

- 少ないボーン数での正確な再現性

主要機能

自動エンベロープ

- 口の周りの微妙なニュアンスの再現。最小限のボーン数でボリューム感のある唇のアニメーションの再現

ペイントベースのわかりやすい編集

- アーティストがボーン的位置や影響範囲を指定できる

リアルタイムデータの調整

- 出力後にリアルタイムシェーダのパラメータによりシワの深さなどを再設定可能

カスタムデフォメーションのサポート

- レイヤー化されたリグや、シェイプ、シミュレーションなどが組みまれていても同様に出力することが可能

エクスポートプロセスの デモンストレーション

SOFTIMAGE®



v.5.11 での SDK の新機能、リアルタイム
シェーダ更新情報、COLLADA の入出力

アビッドテクノロジー株式会社

ソフトイメージ
テクニカルスペシャリスト

梅澤 孝司

v.5.11 での SDK の新機能

- 自己インストールカスタムオペレータ
 - スクリプト、コンパイルオペレータの構築を容易に
 - Custom Operator Wizard
- カスタムSPDLでレジストリへのアクセス無し
 - ワークグループ内のプラグインがAdministrator や Power User 以外でもインストール可能に

v.5.11 での SDK の新機能

- FCurve の改善
 - FCurveEditor 内の Fit と Smooth の機能
 - FCurve のコピーが Set で可能
- Math Library の改善
 - SIVector3、SIRotation、SITransformation の初期化が生成時に可能
 - SITransformation内の値の取得が容易に

v.5.11 での SDK の新機能

- C++ API の改善
 - OM の XSIUtils に対応する CUtils クラス
 - CValue の Int 用のコンストラクタ
 - CDoubleArray と CIntArray の操作が CString::Split と CLongArray::GetAsText 関数で容易に
 - CValue と CRef とその他のクラス間の変換が自動でハンドルされる

v.5.11 での SDK の新機能

- パラメータアクセスの最適化
 - C++ API を利用したコード内で、パラメータにアクセスするためのティップスと新関数
- SDK ガイド → C++ API Development
 - Programming with the C++ API
 - Parameter Performance Optimization

v.5.11 での SDK の新機能

- パラメータアクセスの最適化ティップス

- 悪い例

```
Application app;  
Model model = app.GetActiveSceneRoot();  
ProjectItem camera = model.GetChild(L"Camera");  
Parameter posx = camera.GetParameters().GetItem("posx");  
app.LogMessage( posx.GetValue().GetAsText() );
```

- 良い例

```
Application app;  
Model model = app.GetActiveSceneRoot();  
Camera camera = model.GetChild(L"Camera");  
KinematicState ks = camera.GetKinematics().GetLocal();  
Parameter posx = ks.GetParameters().GetItem("posx");  
app.LogMessage( posx.GetValue().GetAsText() );
```

v.5.11 での SDK の新機能

- パラメータアクセスの最適化新関数
 - GetParameterValue

```
Application app;  
Model model = app.GetActiveSceneRoot();  
Camera camera = model.GetChild(L"Camera");
```

```
// Wrong way of getting a single parameter value. This  
// will traverse the entire camera parameter hierarchy and  
// cache all parameters  
Parameter fov = camera.GetParameters().GetItem(L"fov");  
app.LogMessage( fov.GetValue().GetAsText() );
```

```
// Suggested way of getting a single parameter value.  
CValue val = camera.GetParameterValue(L"fov");  
app.LogMessage( val.GetAsText() );
```

v.5.11 での SDK の新機能

- 新しいイベント
 - siOnSourcePathChange
 - siOnTimeChange
 - siOnActivate
 - siOnMinimize
 - siOnMaximize
 - siOnChangeProject
- Custom Display Host 用のイベント
 - siOnObjectAdded
 - siOnObjectRemoved

v.5.11 での SDK の新機能

- その他
 - メニューのアンカーポイント追加
 - カスタムコマンド用のフラグ、`siAllowNotifications` の追加
 - `Primitive.GetGeometry` で2つの引数を受け取るように
 - フレーム
 - コンストラクションモード

v.5.11 での SDK の新機能

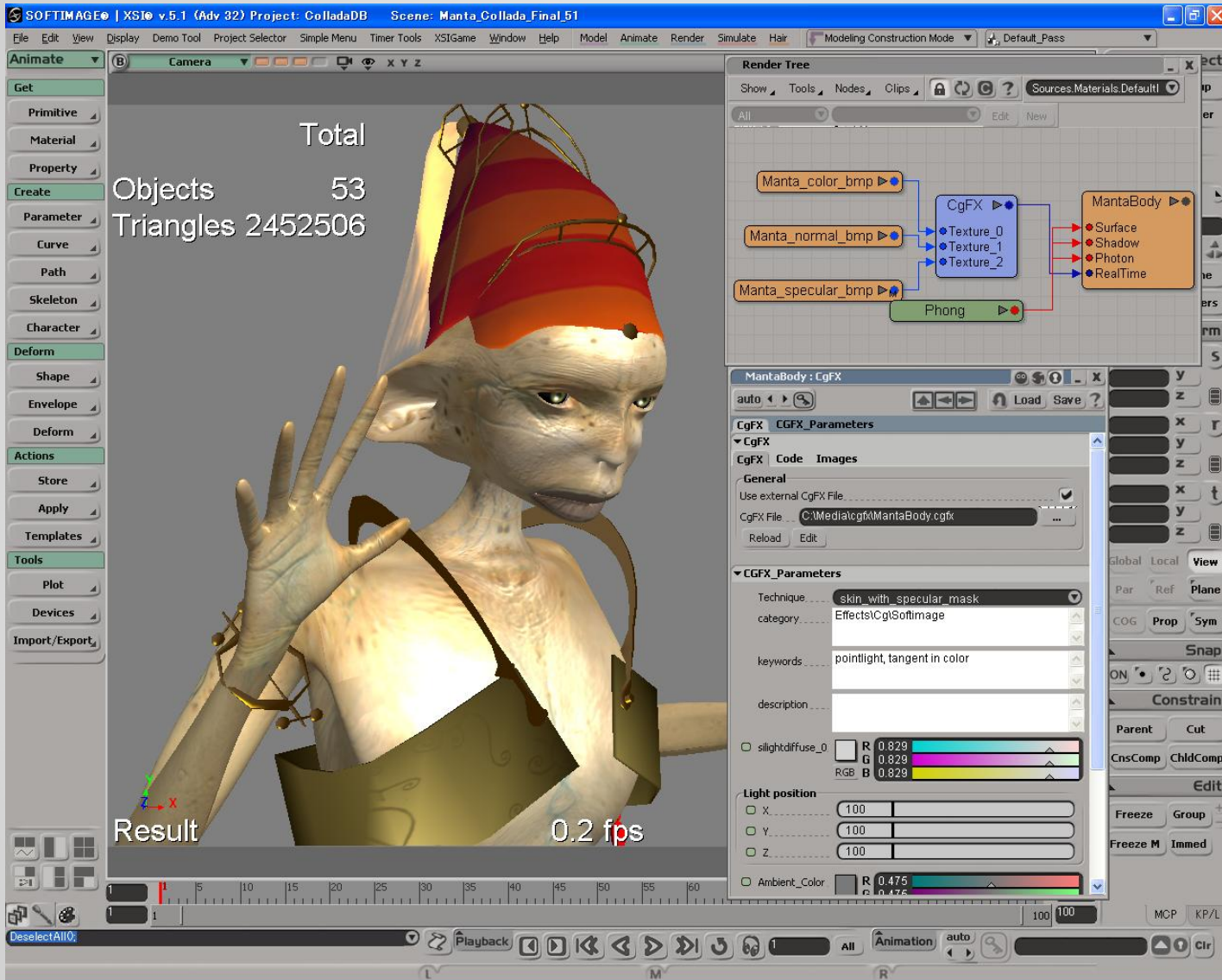
- オブジェクトモデルの更新
 - 新オブジェクト – 2
 - 新メソッド – 14
 - 新プロパティ – 18
 - 新定数 – 2
- C++ API の更新
 - 新クラス – 5
 - 新関数 – 65
- コマンドの更新
 - 新コマンド – 22

リアルタイムシェーダ更新情報

- XSI v.5.11 で CgFX に対応
 - XSI v.5.1、v.5.0 では、CgFX 利用不可
 - XSI v.5.11 と Cg 1.5 Beta 1 か Cg 1.5 Beta 2 が必要

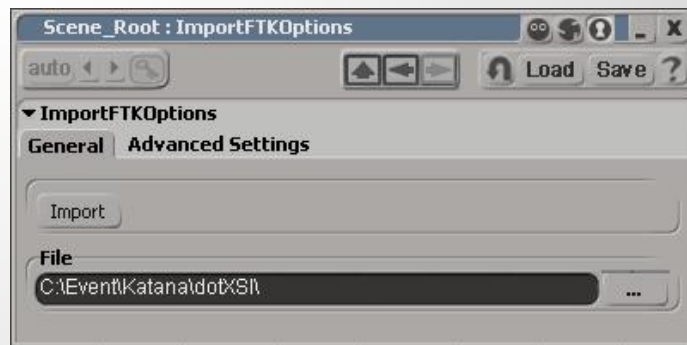
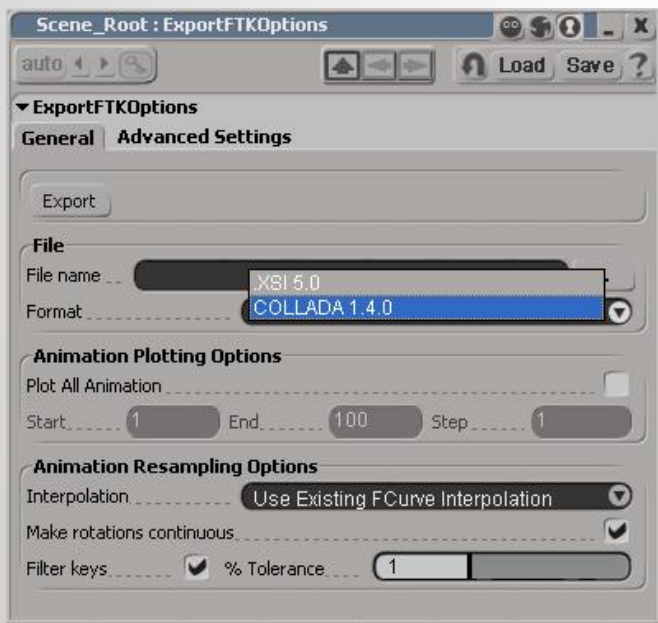
<http://developer.nvidia.com/object/cg-toolkit-15beta.html>

<http://developer.nvidia.com/object/cg-toolkit-15beta2.html>



COLLADA 1.4 の入出力

- SOFTIMAGE|XSI v.5.1 で COLLADA 1.4 の入出力に正式対応
 - COLLADA v.1.3.x の入出力はプラグインで提供していたものを、COLLADA v.1.4 では製品の中に組み込む

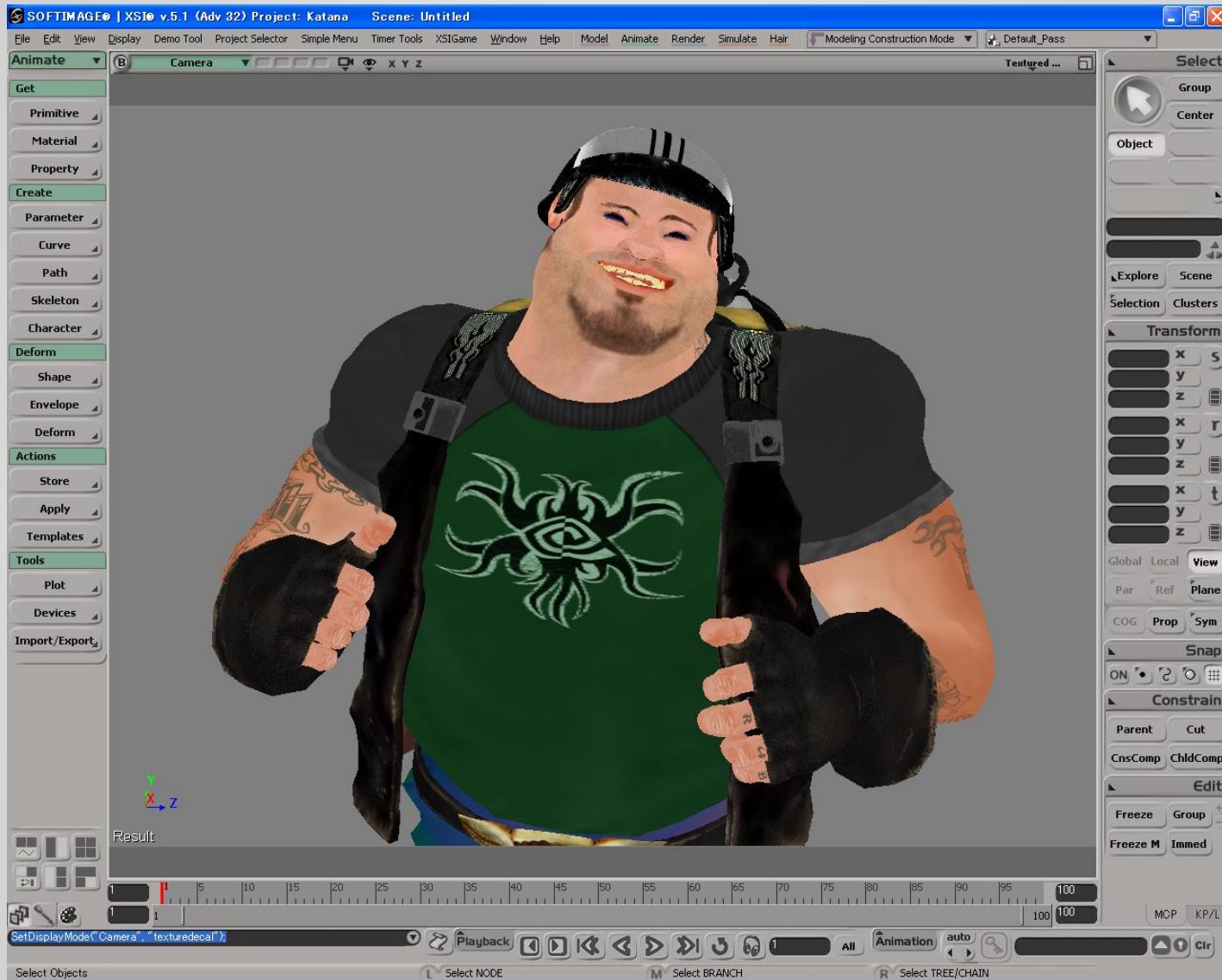


COLLADA 1.4 の入出力

- SOFTIMAGE|FTK で COLLADA v.1.4 に対応、
 - dotXSI ファイル用のファイル入出力ライブラリ、SOFTIMAGE|FTK で COLLADA の直接の入出力に対応
 - ファイルパーサ、セマンティックレイヤーの機能を提供し、データの扱いが容易
- COLLADA 1.4 では、ソースコードを一から書き直しより安定、高速化
 - COLLADA 1.3 では、dotXSI に吐き出しそれを変換していたが、FTK が COLLADA に対応したことにより一から開発、より高速、安定
 - ソースコードも提供
- COLLADA FX にも対応
 - XSI で作成した CgFX のエフェクトの入出力にも対応

COLLADA 1.4 の入出力

- ジオメトリにサブディビジョンを適用
 - XSI の中で設定したサブディビジョンされたメッシュのデータを自動的に出力
- タンジェントを頂点カラーとして書き出し
 - XSI の中で使われるタンジェントデータを、頂点カラー(0~1の間)で出力するか、タンジェント(-1~1)として出力するかのオプション
- シェイプアニメーションの出力
 - シェイプキーのみか、全てのプロットの出力
- XSI のエキストラデータを書き出し
 - COLLADA で対応していないデータを、追加データとして出力、<extra> テンプレートとして出力



Thank you.
