

トゥーンシェーディング技術の発展 ——アニメから始まりアニメを超えて——

The Toon Shader for Anime and Beyond

安生健一

Abstract

トゥーンシェーディングとは、三次元 CG (Computer Graphics) で表される表示対象に対して手描きアニメ風の陰影付けをすることを意味する。手描きアニメの中に三次元モデルを違和感なく登場させるために必須となる表現である。本稿では、まずトゥーンシェーディングに関する基本的な考え方と歴史、更には実用上の問題点と解決事例について述べる。トゥーンシェーディング及び関連する技術の進化や多様化についても、昨今の GPU 及び深層学習的アプローチを用いた 2, 3 の事例とともに考察する。

キーワード：三次元 CG, トゥーンシェーディング, GPU, デジタルアニメーション

1. はじめに

トゥーンシェーディング (Toon Shading) とは、三次元形状モデルに対して、あたかも手描きアニメで描かれたような陰影を与えること、またはその技術や手法のことを意味する。一般的には、映画・テレビ・ゲーム等のエンターテインメント分野で用いられる技術として知られている。しかしそもそものニーズは、それに限るものではない。また、昨今の GPU の急速な性能向上と用途拡大の潮流の中で、トゥーンシェーディングのリアルタイム応用や深層学習を援用した進化も見られる。

このような状況を鑑み、本稿では、トゥーンシェーディングあるいはスタイライズドシェーディング (様式化された陰影付け: Stylized Shading) と呼ばれる技術がどのように生まれ、進化してきたかについて、2, 3 の産業応用とともに解説する。

なお一般的な CG 用語として、シェーディングは表示される物体の各頂点での色情報を求めることである。レンダリングは、表示される三次元シーンの情報から表示スクリーンの各画素の色情報を求める一連の処理のことを指している。

2. トゥーンシェーディングの発展

2.1 原義

三次元物体上の点 p での輝度値を求める方法はいくつかあるが、拡散反射、つまりその点からの反射光が一律に広がる場合を考える。シンプルな計算方法としてランバートの拡散モデルがある。すなわち、点 p の輝度値を、点 p から光源に向かう単位ベクトルと点 p における外向きの単位法線ベクトルの内積値に比例する、として求める。図 1(a) は、三次元 CG キャラクタの顔の

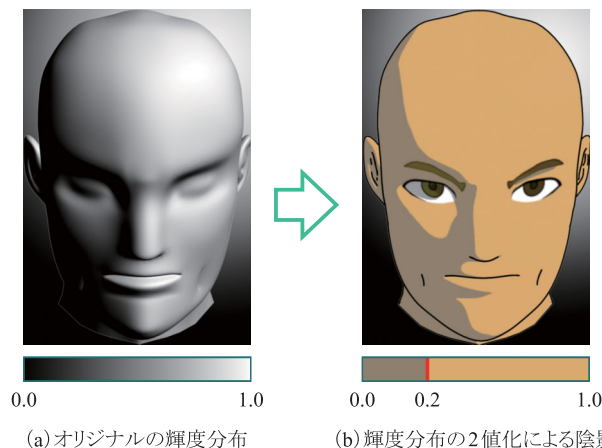


図 1 トゥーンシェーディングの原理 物体表面の輝度分布を 2 値化して明部と暗部を定め、手描きアニメ風の陰影付けを行う。

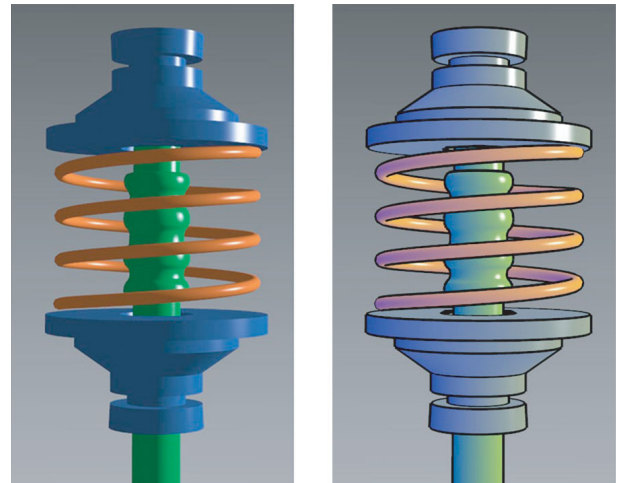
上を p が動いたときのこの輝度値の分布を表している。この図では、輝度分布を 0 から 1 に至る連続的な値として表し、0 に近いほど光源の影響が少ない、すなわち「暗い」ということを意味し、同図では黒っぽく表示されている。1.0 に近い値を取るところでは白っぽく表示している。物体表面の陰影は本来連続的に変化するが、輝度値を明部と暗部の二通りに分ける、すなわち“2 値化”することを考える。そのために輝度値を基にして得られる 2 値化した陰影画像が図 1(b) である。更にしきい値を複数導入すればよりきめの細かい陰影付けができる。このようにして得られた画像は手描きアニメでよく見られる陰影付け（シェーディング：Shading）に対応する効果を生み出す。このようにしきい値を用いた陰影付けはトゥーンシェーディングの典型的な手法である。

なおトゥーンシェーディングという言葉の定義に定説はなく、現在ではデジタルアニメ作品に使われるシェーディング全般のことを意味している。

2.2 トゥーンシェーディングからスタイライズドレンダリングへ

ところで図 1(b) をよく見るとキャラクタの輪郭線や頬のえくぼ（？）の位置に黒い線が描かれている。特に日本の手描きアニメではこのような輪郭線・特徴線の描画が好まれている。図 1(b) の三次元キャラクタは、トゥーンシェーディングにより平たんな顔立ちになるが、これらの特徴線を追加することで顔をより詳しくあるいはより魅力的に表現できるようになる。このような表現は様式化されたレンダリング（Stylized Rendering）、あるいはより一般的に、非写実的レンダリング（Non-photorealistic Rendering）⁽¹⁾ と呼ばれている。

アニメの世界とは異なるが、工業製品の説明図でも、配色は単純化し、輪郭線や特徴は強調して描画することがある。これは CAD（Computer Aided Design）の実用化やデジタルアニメよりずっと前から、テクニカルイラストの分野でよく用いられてきた⁽²⁾。この分野においても三次元 CG を用いた表現手法が有効である。先駆的研究は幾つかあるが、特に G バッファ法（G-buffer）⁽³⁾ がその後の研究をけん引した。三次元 CG の表示技術としては、 z バッファ法が基本である。 z バッファ法では、表示スクリーンの各画素に投影される各表示物体の ID と深さ情報を更新しながら蓄える。この深さ情報に加えて、当該物体の法線情報などの幾何学的情報を蓄えたものが G バッファである。これによってシェーディング処理を画素ごとの計算として分離し、効率的に行えるようになる。例えば輪郭線抽出も併せて行えば、テクニカルイラストで見られるような強調表示が可能となる。その後の技術発展の中で、テクニカルイラストにより特化した手法⁽⁴⁾ も登場した。適用例を図 2 に示す。この手法では、言わばフェイクなライティング効



(a) フォンシェーディングによる画像

(b) 輪郭線の付与、及び色相を暖色系から冷色系に展開して見やすくした

図 2 三次元モデルのテクニカルイラストレーション (a) がオリジナルの三次元形状モデルを通常のフォンシェーディングを用いて表示したもの。(b) は輝度分布の広がりを単なる明暗とせず、暖色系から冷色系に変更して見やすくした。ハイライトは (a) と (b) で共通して三次元的なリアリティを残す。

果としてメタリックな陰影や暖色系から寒色系への色相変化を実現し、形状の把握しやすさを増すことに成功している。

なお G バッファは、GPU を駆使した遅延レンダリング⁽⁵⁾（Deferred Rendering）等のよりリアルな表現体系にも威力を発揮する。

3. デジタルアニメとトゥーンシェーダ

1990 年代のアニメ制作の観点から言うと、トゥーンシェーダはアニメ表現の多様化と効率化の両方に大きく貢献するだろうと期待されていた。ただし、トゥーンシェーディングによって三次元モデルが見た目は手書き風になったとしても、二次元の手描きアニメとうまくなじませる必要があった。このなじませ方が実用化に向けた最初の課題であった。以下では三次元 CG がどのようにしてアニメ作品の制作へ取り入れられ、現在の多様な作品作りにつながったかを振り返る。

3.1 2D/3D のハイブリッドアニメ

先述のように、トゥーンシェーディングによって手描きアニメ風の三次元キャラクタが作れる。そうすると見た目は手描き風でも、動きやカメラは三次元 CG の世界で制御できるので、自由度は格段に高くなり、手描きアニメでは困難な表現が（比較的）容易に得られる。例えば、カメラがキャラクタに焦点を合わせながら、キャラクタの回りを一周する、といったことは手描きではかな

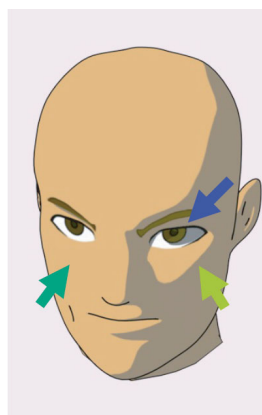
り困難だが、三次元の世界では容易である。しかし1990年代の多くの制作会社は、全てのキャラクタや背景を三次元CGで作ることは想定せず、手描き(2D)と三次元CG(3D)とを組み合わせるハイブリッドアニメ制作を目標としていた。あくまでルックは手描き風で作るのがアニメ作品だったわけである。一方、ピクサは独自の路線を歩み、フル(3D)CG映画「トイストーリー」^(注1)を1995年に劇場公開した。

1999年は本格的な劇場版ハイブリッドアニメ作品が公開されたという意味でエポックメイキングな年であった。具体的には、ディズニーの映画「Tarzan(邦題:ターザン)」^(注2)、ワーナー・ブラザーズの映画「The Iron Giant(アイアンジャイアント)」^(注3)、そして東宝/ワーナー・ブラザーズの映画「Pokémon the First Movie」^(注4)の3作品である。それぞれの作品で、ハイブリッドアニメを実現するための独自の技術的工夫があった。例えば「Pokémon」では一部の背景を三次元CGを用いて表現し、手描き表現ではなし得ないダイナミックなカメラワークを実現した。「アイアンジャイアント」はロボットが主役であり、ソリッドな動きを表すために三次元ポリゴンモデルで表された。他の人物キャラクタや背景はほぼ手描きであった。3作品ともハイブリッドとはいえず手描きアニメを補強する形で三次元CGを取り入れていた。実は先のなじみの問題も含め試行錯誤を繰り返しての作品制作であった。その後の変遷を経て、今日のデジタルアニメ作品は、ディズニー映画のようなスタイライズドレンダリングによるフルCG作品と、日本の作品に見られるように手描き主体のハイブリッドアニメを両極とし、様々なバリエーションを迎えることとなった。

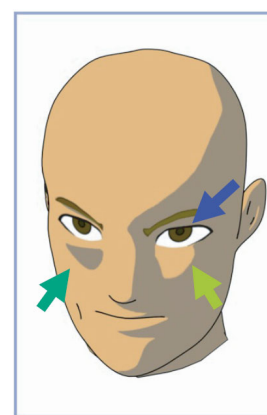
3.2 デジタルアニメーションの表現技術

上述のように僅か20年ほどの間にアニメ作品は多様化した。それにはCG技術の高度化が大きく貢献している。高度化にあたっての技術課題は大きく分けて表現技法に関する課題と、ワークフローの効率化に関するものに分けられる。もちろん相互に深く関係する場合もあるので、課題解決にはどちらの視点も重要である。

本節では主にハイブリッドアニメと手描きのデジタルアニメ表現に関する具体的な技術課題と解決事例を示す。



(a) しきい値処理による陰影



(b) (a)に加筆修正後の陰影

図3 トゥーンシェーディングの問題と解決例 (a)ではしきい値処理のみで陰影付けしているの、アニメの演出には不十分である。(b)はShade Painterによる編集結果を示す。頬の丸みを強調するような暗部の修正(図中の黄緑矢印)やフェイクシャドウ(緑矢印)がペイント処理で追加され、かつアニメーションに反映される。

3.2.1 ハイブリッドアニメの陰影表現

図1ではトゥーンシェーディングの原理を示したが、これをハイブリッドアニメに適用するには実は幾つか問題がある。例えば単純にしきい値を用いただけでは、図3(a)のような陰影ができてしまう。右目の一部(同図の青矢印で示されるエリア)が暗くなったり、右の頬の辺り(同図黄緑矢印)の陰影の境界線がストレートになり頬のふくら感が出ていない。これらは三次元モデルの幾何的な情報から「正確に」しきい値処理をした結果である。いくら物理的に正しくとも、アニメには不適切であると言える。アニメの陰影にはフェイクが多く用いられ、それが演出効果としてとても重要なのである。つまり陰影をアーティストのセンスで実現できることが望ましい。このような柔軟な編集作業を実現する方法の一つにShade Painterがある⁽⁶⁾。図3(b)はShade Painterによって編集したキーフレームだが、目の部分は明るく、左頬は盛り上がって見えるように明暗を修正、更に右頬(同図緑矢印)にはフェイクな暗部を足している。本手法では、デフォルトとして図3(a)に示したような従来のトゥーンシェーディングによるアニメーションを入力とし、キーフレームと呼ばれる重要なフレームでペイント処理して編集し、それを基にキーフレーム補間で最終的なアニメーションを得る。その場合の補間は、編集後の輝度分布に関するものでRBF(動径基底関数)を用いている。

ところでCG研究においてはいわゆる ill posed な問題を扱うことが多い。ゆえに後述の深層学習アプローチも有効だが、様々な数学的発想や手法も問題解決に貢献している⁽⁷⁾。Shade PainterにおけるRBF補間はその一例である。

(注1) <https://www.disney.co.jp/fc/toystory> (2023年12月確認)

(注2) <https://www.disney.co.jp/studio/animation/0612> (2023年12月確認)

(注3) <https://eiga.com/movie/85543/> (2023年12月確認)

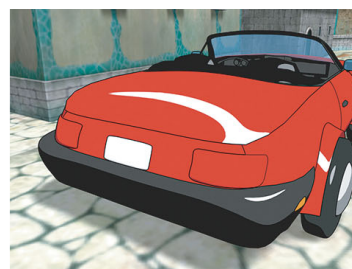
(注4) https://youtu.be/hX-NHafvY5I?si=wCGiZ_5MQQGvYrcN (2023年12月確認)



(a) しきい値処理による単純な
ハイライト



(b) 望ましいハイライト



(c) 複雑なハイライトの生成例

図4 ハイライトアニメーションの作成 手続き型手法によって、アニメ風のハイライトの形と動き(b)(c)がデザインできる。

3.2.2 ハイブリッドアニメのハイライト表現

ハイブリッドアニメのハイライトにも工夫が必要である。図4では、三次元モデルで作られた自動車の上にアニメ風のハイライトが付加されている。図4(a)，(b)では、車の正面の窓にグレーで示されているのがハイライト領域である。図4(a)では2.1の場合と同様に、プリンのモデルと呼ばれるハイライト計算式を導入し、そのしきい値処理によりハイライト領域を定めている。しかしこれはアニメらしくない。窓の大部分が、単に薄い青色に塗り込まれたように見えている。アニメに親しんでいる方ならお気づきと思うが、アニメにおける窓のハイライトとは厳密にはハイライトではなく、窓に何か映り込んでいる様子を「記号化」して表してハイライトと呼んでいるのである。例えば図4(b)のように、ストライプ型の(記号的な)ハイライトで二つの帯が並行に並んで描かれたりするのである。しかも、これは視点の変化に合わせて動かなければならない。1フレームずつテクスチャとして描き込む方法もあるが、労力がかかりすぎてしまう。アニメらしいハイライトを時間方向にも違和感なく作れるかが課題である。

この問題を解決する手段の一つに Highlight Shader⁽⁸⁾がある。本手法は関数型モデルを採用しているが、図4(b)のハイライトも簡単なパラメータ設定で容易に構成できる。また Shade Painter のときと同様に、各キーフレームでアーティストがハイライト形状を編集し、ハイライトのキーフレームアニメーションが作れる。Highlight Shader は表示対象の幾何情報も考慮した関数型モデルなので、図4(c)のような複雑なハイライトも生成できる。

Highlight Shader も Shade Painter も GPU を活用した編集機能を持ち、商業作品の中で使用された。しかしアーティストの作業を支援するツールとしてはまだ多くの課題が残っている。例えば、ゲーム応用を念頭に、高精細高解像度のリアルタイムキャラクタをどのように表現・編集すべきかという問題がある。これに対する解決策も模索され始めている⁽⁹⁾。

3.2.3 手描きアニメのデジタル化

前項まではハイブリッドアニメを念頭に述べてきたが、伝統的な手描きアニメも多く制作会社で制作工程がデジタル化され、全フレームがデジタル画像として処理されている。デジタル化による表現力の向上も期待されているが、特にワークフローの効率化が切望されている。

本項では、制作効率を上げるための研究例として、手描きアニメの自動彩色処理を紹介する。これは既述のアプローチと異なり、完全に二次元の手描きアニメに関する研究である。

手描きアニメの彩色処理は、(1)色付けしたいフレームのキャラクタの線画とカラーパレットが与えられ、(2)与えられた線画で囲まれるキャラクタの各部分領域に、カラーパレットにある色を割り振る、という作業である。ここでカラーパレットとは、キャラクタや小物などの物体ごとに定義される色見本から定義される。つまりキャラクタの線画は、キャラクタを単色で塗られる部分領域の境界線の集まりとして表されていることになる。なお映像業界では、単色で塗られた領域のことをベタ塗りされた領域ということがある。例えば図5の点線枠内において、上段が指定されたキャラクタの線画、下段はその線画に基づいた人手による彩色結果が示されている。

この作業は煩雑な手作業であるため、従来から効率化が切望されていた。彩色処理は理想的には100%自動化したいが、作業者の修正作業がなるべく少なくなるような半自動化を目指すのが現実的なゴールであると考えられる。ここでは産学協同研究プロジェクトとして推進されている Few-shot Learning に基づくアプローチ⁽¹⁰⁾を概観する。このアプローチの技術的なポイントは、実写ビデオ映像のスタイル変換のためのパッチベース学習法⁽¹¹⁾をアニメのようなベタ塗り彩色手法へと拡張した点である。技術の詳細は原論文⁽¹²⁾を参照されたい。

図5では、2種類のキャラクタについての彩色結果を例示している。この例では、各キャラクタに対し僅か二

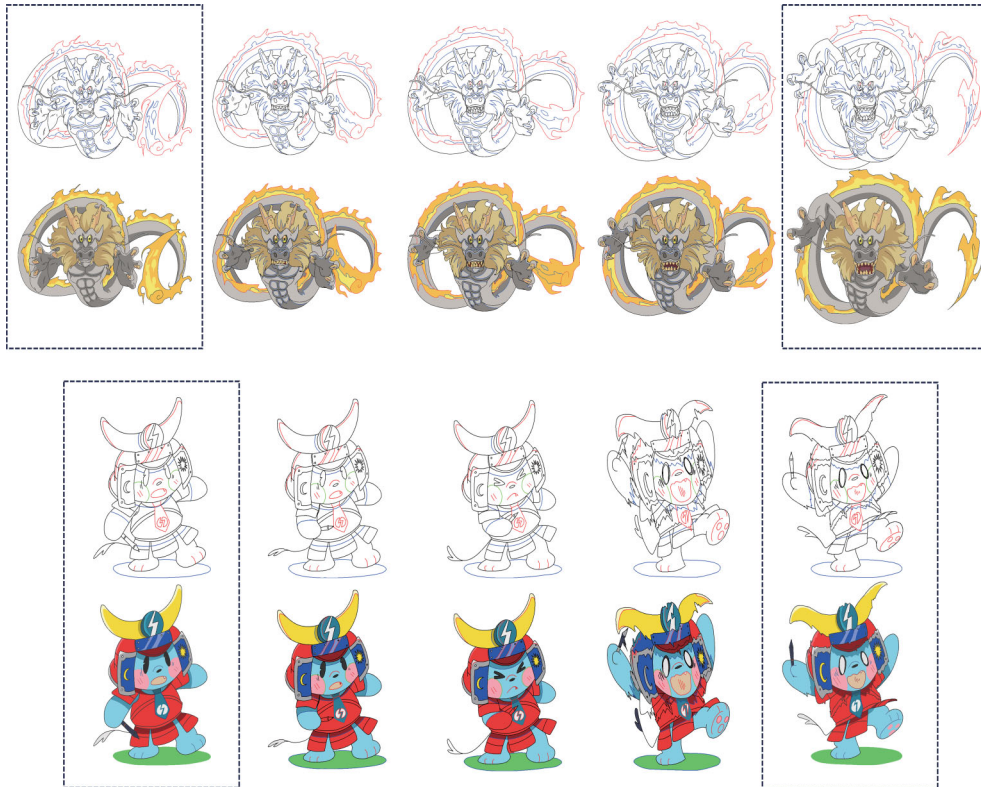


図5 手描きアニメの彩色処理に関する深層学習アプローチ 上下にタイプの違うキャラクターを例示した。点線枠内では、入力される線画と手作業によって得られた彩色結果を示している。上段と下段それぞれの場合に、二つの学習データ（2種類の点線枠内の事例）のみから学習している。点線枠なしの線画は新たな入力線画、その下の彩色画像はそれに対する本アプローチの結果である。おおむね良好な結果であるが、局所的に不自然な結果もある。例えば下段の点線枠内のキャラクターが右手に持っている黒いペンは新たな線画入力では背景のボディ色である赤になる場合がある。（©OLM Asia SDN BHD）

つのサンプル（点線枠で囲まれた線画と彩色画像）から点線枠外にある線画を彩色するために必要なモデルを1分半程度で学習し、彩色結果（点線枠外の線画の下段の画像）を得る。図5に示された学習データとは異なる3種類の線画（点線枠のない線画）に対し、本アプローチにより得られる彩色結果を、それぞれの線画の下に示す。一見すると良好な結果のようだが、局所的には色塗りの結果が望ましくないものがある。

他の多数の事例についてもこのアプローチを試したが、手作業で作成した彩色結果と比較して求めた正解率は平均して6割から7割となった。この正解率が妥当かどうかは評価の分かれるところであるが、本研究プロジェクトに参画した映像制作会社では、本手法から得られる画像を「初期値」として適宜修正するというワークフローにより約20%の効率化が得られた。

3.2.4 デジタルアニメの多様性と可能性

再び三次元モデルを用いたデジタルアニメに戻る。北米やヨーロッパのデジタルアニメも多様化している。とりわけアメリカンコミックの表現をアニメーションの世界にまで押し広げたソニー・ピクチャーズの映画

「スパイダーバース（Spider-Verse）」シリーズ（2019年、2023年）は、特筆すべき作品である。

制作にあたっては様々なツール開発があった。例えば2019年の映画第一作「Spider-Man: Into the Spider-Verse」^(注5)において、アメリカンコミックで重要視される輪郭線の強調表現がアニメの世界に引き継がれた。3.で述べたことと同様に、輪郭線の時間方向の変化をどう制御するかが技術的課題となった。「スパイダーバース」シリーズの場合は、深層学習を基にした「初期値」がアーティストに提示され、必要に応じてアーティストが三次元モデルに直接加筆するというワークフローで実現された。学習にあたっては各キャラクターに対し、その表情・動きを想定して600フレームほどの学習データを用意し、scikit-learnのGradient-Boosting Regressorを用いて学習した⁽¹³⁾。なお輪郭線データはUV空間（三次元キャラクターの幾何情報を局所的に平面上に展開したもの）上で扱っている。以上述べた輪郭線制御手法は、ワークフローの効率化を図るという意味で有効であった。

（注5） <https://youtu.be/g4Hbz2jLxvQ?si=oDkqswdoTHrJV7l5>（2023年12月確認）

4. おわりに—アニメを超えて—

三次元キャラクタの今後の活用を考えてみる。映画やゲームはもちろんであるが、ライブエンターテイメントでの本物の人間との共演や、リアルなデジタル環境空間、例えばデジタルツインと呼ばれる世界に登場するアバターとしてのニーズは増加するであろう。

トゥーンシェーディングは三次元キャラクタの表示手法として活用されているが、上記のニーズや応用を想定すると更に進化すると思われる。本稿では、深層学習がデジタルアニメーションのワークフローの「効率化」に有効である事例を述べた。これからの深層学習は、キャラクタやアニメの「新しい表現」を引き出せるだろうか？ より基本的なこととして、今日の細分化した技術群を考えると、技術者とアーティストによる本格的な協働が切望される。

文 献

- (1) B. Gooch and A. Gooch, Non-photorealistic rendering, A K Peters, 2001.
- (2) J. Martin, Technical Illustration : Materials, Methods, and Techniques, Time Warner Books, UK, 1989.
- (3) T. Saito and T. Takahashi, "Comprehensible rendering of 3D shapes," Proc. SIGGRAPH '90, pp. 99-106, 1990.
- (4) A. Gooch, B. Gooch, P. Shirley, and E. Cohen, "A non-photorealistic lighting model for automatic technical illustration," Proc. SIGGRAPH '98, pp. 447-452, 1998.
- (5) S. Hargreaves and M. Harris, Deferred shading (slides).

http://download.nvidia.com/developer/presentations/2004/6800_Leagues/6800_Leagues_Deferred_Shading.pdf

- (6) H. Todo, K. Anjyo, W. Baxter, and T. Igarashi, "Locally controllable stylized shading," ACM Trans. Graph., vol. 26, no. 3, Article 17, 2007.
- (7) 安生健一, CG は数学でできている, 日本評論社, 2023.
- (8) K. Anjyo and K. Hiramitsu, "Stylized highlights for cartoon animation and rendering," IEEE Comput. Graph. Appl., vol. 23, no. 4, pp. 54-61, 2003.
- (9) L. Peticam, K. Anjyo, and T. Rhee, "Shading rig : dynamic art-directable stylised shading for 3D characters," ACM Trans. Graph., vol. 40, no. 5, Article 189, 2021.
- (10) ARG Tech Blog, "アニメスタイルの自動彩色手法のご紹介," <https://tech.imagicagroup.co.jp/res/5733/>
- (11) O. Texler, D. Futschik, M. Kučera, O. Jamriška, Š. Sochorová, M. Chai, S. Tulyakov, and D. Sýkora, "Interactive video stylization using few-shot patch-based training," ACM Trans. Graph., vol. 39, no. 4, Article 73, 2020.
- (12) A. Maejima, H. Kubo, S. Shinagawa, T. Funatomi, T. Yotsukura, S. Nakamura, and Y. Mukaigawa, "Anime character colorization using few-shot learning," SIGGRAPH Asia 2021 Technical Communications, ACM, New York, USA, Article 8, 2021.
- (13) M. Seymour, Ink lines and machine learning. <https://www.fxguide.com/fxfeatured/ink-lines-and-machine-learning/>

(2023 年 8 月 31 日受付)



あんじょう けんいち
安生 健一

オー・エル・エム・デジタル技術顧問,
IMAGICA GROUP プリンシパルリサー
チャー, Victoria University of Wellington 非常
勤教授. CG の研究開発と実用化に従事. 博士
(工学). 平 26 年度文部科学大臣表彰科学技術
賞受賞. 著書「CG は数学でできている」
(著), 「3DCG アニメーション」(共著) など.