

VR エンタテインメントシステムのための リアルタイムマンガ風画像生成シェーダーの開発

Development of real time cartoonization shader for VR entertainment system

藤村 航¹⁾, 小出 雄空明¹⁾, 奈良 優斗¹⁾, 白井 暁彦²⁾

Wataru FUJIMURA, Yukua KOIDE, Yuto NARA and Akihiko SHIRAI

1) 神奈川工科大学 大学院 工学研究科 博士前期課程 情報工学専攻

(〒243-0292 神奈川県厚木市下荻野 1030, manga@shirai.la)

2) 神奈川工科大学 情報学部 情報メディア学科

(〒243-0292 神奈川県厚木市下荻野 1030, shirai@ic.kanagawa-it.ac.jp)

概要: 本論文は, カメラで撮影した画像に, リアルタイムでマンガ風の画像処理を施すシェーダー開発について報告する. 第 20 回・国際学生対抗バーチャルリアリティコンテストにおいて発表した「瞬刊少年マルマル」では, カメラで取得した人物に, リアルタイムでマンガ風の画像処理を施すことで, マンガの世界を体験するシステムを実現できた. 本報告では, コンテスト以降に行われた作風の画質向上や, HD クラスの解像度に対応するための技術改善, Unity3D のシェーダーを用いることでスマートフォンなどに汎用化させるための手法について, 開発事例を挙げ報告する.

キーワード: Manga, Shader, Kinect, Cartoonization

1. はじめに

本論文では, 第 20 回国際学生対抗バーチャルリアリティコンテスト (IVRC2012) で発表された「瞬刊少年マルマル」を初めとする, マンガ風画像処理を行うシェーダー開発について報告する.

近年「マンガの世界に入る」といったインタラクティブシステムやスマートフォンアプリが開発され話題になっている. 大日本印刷が開発によるキャラクターと一緒に写真を撮影できる「ジョジョの奇妙なスタンド体験」[1] や, チームラボによるブランドショップ店内向けサイネージ「teamLabCamera」シリーズ [2], スマートフォンアプリでは「漫画カメラ」[3] など多数存在する.

しかし, これらの作品の多くは, カメラで撮影した画像をマンガ風に加工しており, 撮影から生成まで 1 秒近くの時間が必要である.

今後, 体験者が体験中もマンガの世界に入り込んでいるような演出を行う上で, マンガ風の画像処理をリアルタイムで行う必要があると考えられる.

2. 先行研究

松下は, 計算機によるコミックのコード化するために必要なマンガ構成要素を抽出する手法についてまとめている [4]. 松下は, マンガのコマの構造を (1) 知的内容 (2) 書誌記述 (3) 構造記述 (4) グラフィック要素の 4 要素をモ

デル化することで, 汎用化が行えると報告している.

梅田らは, Kinect で認識した体験者の動きや言葉から, マンガ効果を演出する手法について報告している [5][6]. 体験者の動きに合わせ, 指定したマンガ風強調表現を画像処理によるリアルタイム生成を報告されているが, 内部処理は公開されていない.

3. 瞬刊少年マルマルとは

「瞬刊少年マルマル」は, 体験者がマンガの世界に飛び込み主人公となって活躍し, オリジナルのマンガを生成できる体験者没入型 VR エンタテインメントシステムである.

以下に, 瞬刊少年マルマルのロードマップを示す (図 1).



図 1: 瞬刊少年マルマルの推移

3.1 IVRC2012

3.1.1 予選大会バージョン

2012 年 9 月 13, 14 日に行われた予選大会では, マンガ風の画像処理, ストーリー 1 本の構成で展示を行った. マンガのコマを生成する要素を抽象化し, コマの素材をレイヤー

構造（図2）にして表示することで、体験者を取り込んだコマを生成する手法を確立した。

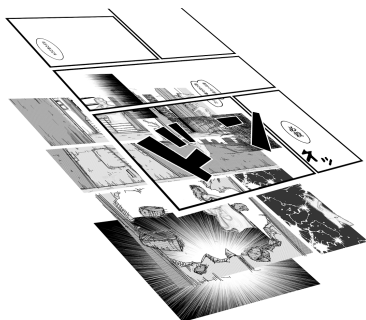


図 2: コマのレイヤー構成概念図

3.1.2 本選大会バージョン

2012 年 10 月 25 日～27 日に行われた本選大会では、予選大会で指摘された音の追加、ストーリーの追加を行った。ストーリー追加を行う上で、コマの生成する要素をプログラム上で記述すると、今後のストーリー追加が難しくなると考え、コマに使うテクスチャや体験者の画像の位置・大きさ調整のデータを外部ファイルに記述しスクリプト化を行った（図3）。スクリプトは、CSV 形式によるテキストファイル、PNG 画像ファイル名、画像配置データによって構成される。

コマNo	コマ位置X	コマ位置Y	体験者画像位置X	体験者画像位置Y	体験者画像拡大率
1	886	0	224	545	0.5
2	0	0	118	300	1.2
3	0	815	800	120	0.8
4	1136	1331	-157	384	1.3
5	0	1331	56	100	1.6

図 3: コマの構成に用いるスクリプトの一部（表形式）

3.2 国際版「Manga Generator」

IVRC2012 を終えてから、研究として用いるため、指導教員の元でシステムや KinEmotion 姿勢認識アルゴリズム [7] の再構成を行った。アルゴリズムを再構成したことで、姿勢認識の段階を増やすことができた。

また、国際展示用のオリジナルストーリー 1 本を書き起こし、英語とフランス語の 2ヶ国語にわけ、展示を行った [10]。

3.3 まんが王国とっとり「まんが博・乙」

2013 年 7 月 13 日～8 月 24 日の間、鳥取県で行われたとっとり王国とっとり「まんが博・乙」のイベントにて長期間展示をした。遠隔地からのメンテナンスできる環境の構築、起動・終了・エラー時にサーバーへメッセージを送信する機能の追加、シェーダーの改善を行った。

また、オリジナルストーリー 3 作に加え、すでに有名なマンガ作家とのコラボレーションしたストーリー 3 作を追加した。

4. マンガ風シェーダー

瞬刊少年マルマルでは、体験者をマンガの世界に没入させるため、カメラで取得した人物画像を「マンガ風」に画像処理を行う。

マンガ的な画像処理を行う上で必要な要素は、

- (1) 背景から人物の切り出し
- (2) エッジ強調
- (3) 減色処理

があげられる。画像処理にはグラフィックス・プロセッシング・ユニット（GPU）を用いた分散処理によって実装することで、画像処理を高速に行うことができ、リアルタイムで画像をマンガ風加工することが可能になる。

一般的にエッジの検出・強調処理は、Sobel フィルタ、laplacian フィルタ、Canny フィルタなどが用いられる。しかしこれらの画像処理は、近傍の参照回数や演算が多くなり、シェーダーモデル 2.0 の最大命令スロット数である 64 スロットを超えてしまうため、リアルタイム化が難しい。

予選大会バージョンでは、注目画素の斜め方向に 1 次微分を行うフィルタによってエッジ検出を行った。

今回システムを開発・実行した環境は、Windows 7、VisualStudio 2010、XNA Game Studio 4.0、Kinect for Windows SDK v1.7 である。

4.1 人物切り出し

人物の切り出しには、Kinect で取得した深度画像を用いる。人物認識した深度画像をマスクとして用いることで、カラー画像から人物のみを切り出す（図4）。Kinect では、深度情報を赤外線によるランダムドットパターンを用いて取得している [8]。この手法では、赤外線を吸収する黒色の物体、ガラスやプラスチックなどの反射物では認識しづらいといった特性がある。よって、人間の髪の毛や黒い服装の場合、認識人物像がちらつくことがある。浅見らは距離画像のノイズを削減するために、距離画像を蓄積することでノイズの低減が可能であることを報告している [9]。



図 4: カラー画像（左）、深度画像（中央）、切り出した人物画像（右）

4.2 エッジ強調

エッジ強調は、注目画素の斜め方向に 1 次微分を行うフィルタによって行った。

カラー画像だけでは、人物の輪郭線を強調することができなかったため、深度画像によるエッジ検出も行った。深度画像は白色と黒色のみで構成されているため、人物の境界を明確に検出することができた。この深度画像で検出したエッジを、カラー画像のエッジに乗算することで、輪郭線を強調することに成功した（図5）。

4.3 階調変化とエッジ強調

図5のエッジ強調のみでは、ライティングによって現れる人物の影を表現できない。よって人物の輪郭線の下層に、カラー画像をグレースケール化したのち、階調変換を行う



図 5: 認識人物画像にエッジ検出を行った

```
// 解像度からエッジのオフセットを算出
float2 edgeOffset = 1 / screenResolution;

n1 = tex2D(tsA, inp.uv + float2(-1,-1) * edgeOffset);
n2 = tex2D(tsA, inp.uv + float2( 1, 1) * edgeOffset);
n3 = tex2D(tsA, inp.uv + float2(-1, 1) * edgeOffset);
n4 = tex2D(tsA, inp.uv + float2( 1,-1) * edgeOffset);

// エッジ抽出した画像
colorImage = abs(n1 - n2) + abs(n3 - n4);

// モノクロ化処理
colorImage.rgb = dot(colorImage.rgb, float3( 0.298912f, 0.586611f, 0.114478f));
```

図 6: エッジ検出のシェーダーコード (HLSL)

ことで陰影をつける (図 7)。その後、エッジを乗算することで陰影の付いた線画を生成した (図 8)。

この処理では、階調数を増加した場合、ディティールやマンガとして重要な目鼻の部分にエッジを描画することができた。しかし階調数が増加するとエッジも増加し、顔が見づらくなる問題が発生する。階調数を減少した場合では、エッジにより顔が見えづらくなることはなくなったが、目鼻の部分が階調変化で埋没してしまい、認識しづらる。



図 7: カラー画像をグレースケール化し階調変換



図 8: 階調変換した画像にエッジを乗算

```
// 階調画像を微分
float fx = tex2D(tsA, input + float2( 0, 1) * offset) - tex2D(tsA, input + float2( 0,-1) * offset);
float fy = tex2D(tsA, input + float2( 1, 0) * offset) - tex2D(tsA, input + float2(-1, 0) * offset);

// 深度画像を微分 (これにより、プレイヤーの周りの輪郭線を目立たせる)
float bfx = tex2D(tsB, input + float2( 0, 1) * offset) - tex2D(tsB, input + float2( 0,-1) * offset);
float bfy = tex2D(tsB, input + float2( 1, 0) * offset) - tex2D(tsB, input + float2(-1, 0) * offset);

// 画素の勾配を計算
float centercolor = sqrt(fx * fx + fy * fy);
float outline = sqrt(bfx * bfx + bfy * bfy);

// 輪郭線情報を追加
centercolor += outline;

// 線画だけでなく、ボスタリゼーション効果を追加する
centercolor += (1.0f - (src.a * 0.114 + src.g * 0.587 + src.b * 0.299));

// 反転
centercolor = 1.0 - centercolor;
```

図 9: 階調変化とエッジ強調のシェーダーコード (HLSL)

4.4 色差による処理

この処理では、画像の色差をとることで、瞳や影などの黒い部分を強調する。また人の肌を、HLSL の組み込み関数「smoothstep」を用いて、指定した 2 つの閾値による色分け表現を実装する。if 文や for 文を用いず閾値のみを変更することで、強調部分の範囲を指定することができ、さらに短いコードで陰影をつけることが可能になった。



図 10: 色差を求め、閾値によって色分けを行った

```
// 深度画像を差分 (これにより、プレイヤーの周りの軸距離を算出させる)
float bfy = tex2D(tdb, input + float2(0, 1) * offset) - tex2D(tdb, input + float2(0, -1) * offset); // y方向の差分
float bfx = tex2D(tdb, input + float2(1, 0) * offset) - tex2D(tdb, input + float2(-1, 0) * offset); // x方向の差分

// 画像の勾配を計算
float outline = sqrt(bfx * bfx + bfy * bfy);

// 色相を求める
float irosa = src.r * src.g + src.b;

// 輝度によって色を区切る
float irosa = smoothstep(0.35f, 0.45f, irosa);

colorImage.rgb = irosa;
```

図 11: 色差を求めるためのシェーダーコード (HLSL)

4.5 展示セットアップでの発見

瞬刊少年マルマルは体験型エンタテインメントシステムであるため、人通りが多い場所や、目に留まる場所で展示することが多い。よって、カメラ・体験者・光源の位置関係によって陰影の付き方が変わりやすい。

このような場合は、マンガの画風に合わせた画像を得られるように、シェーダーのパラメータを調整し、ライティングを生かした処理を行う必要がある。図 12 は色差によるシェーダーを用いて、体験者の後ろ側から日が当たる展示セットアップ環境で、逆光を生かした演出を行った場合の結果であり、実験室では発見できない効果であった。



図 12: 逆光を用いたシルエット風の表現

5. Unity3D でのモジュール化

これまでに挙げたシェーダーは、XNA Game Studio 4.0 の HLSL を用いて開発を行った。しかし、今後様々な端末・環境でマンガ風シェーダーを用いる場合、シェーダーの汎用化が必要になった。そこで、総合ゲーム開発環境である Unity3D のシェーダーを用いることで、スマートフォンやタブレット端末にも対応したアプリ・システム開発が容易になる。

Unity3D には、ウェブカメラやタブレットのカメラで取得した画像をテクスチャとして扱う「webcamtexture」クラスが存在する。このクラスで取得したテクスチャにシェーダーを記述することで、各種端末でリアルタイムにマンガ風の画像を撮影することが可能になる (図 13)。

6. 結論・展望

本論文では、IVRC2012 で発表された「瞬刊少年マルマル」を初めとする、マンガ風画像処理を行うシェーダー開

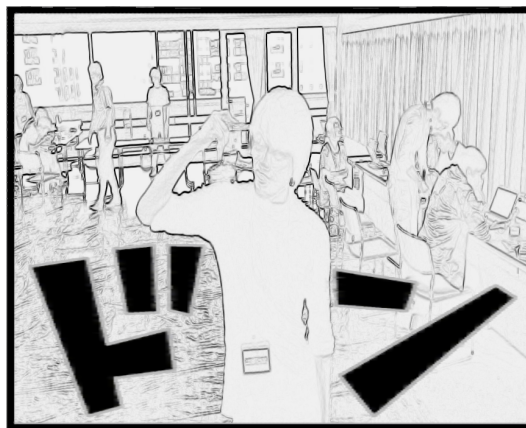


図 13: Unity3D のシェーダーを用いた実行例

発について報告した。

今後の展望としては、マンガの作風に合わせたシェーダー開発、深度画像を用いたトーンの表現、頂点シェーダーを用いた変形・パース表現などを実現したいと考える。

参考文献

- [1] 【ジョジョ展】AR 展示「ジョジョの奇妙なスタンド体験」のデモ映像!, 2012. <http://www.youtube.com/watch?v=fat81YDRshg>
- [2] TEAMLAB: teamlabcamera, 2010-2012. <http://www.team-lab.net/tag/teamlab-camera>.
- [3] スーパーソフトウェア: 漫画カメラ, 2012. <http://tokyo.supersoftware.co.jp/mangacamera/>.
- [4] 松下光範: コミック工学の可能性, 第 2 回 ARG WEB インテリジュンスとインタラクション研究会, pp. 63-68, 2013.
- [5] 梅田大樹, 森谷友昭, 高橋時市郎: Kinect を用いた漫画風表現のリアルタイム生成システム, VC シンポジウム 2012, 2012.
- [6] Daiki Umeda, Tomoaki Moriya, and Tokiichiro Takahashi. “Real-time manga-like depiction based on interpretation of bodily movements by using kinect,” SIGGRAPH Asia 2012 Technical Briefs, pp. 28:1-28:4, 2012.
- [7] Yuto Nara, Akihiko Shirai: KinEmotion: Context-Controllable Motion Analysis Method for Interactive Cartoon Generator, SIGGRAPH 2013 Posters, 2013.
- [8] 上田 智章: トランジスタ技術 2012 年 8 月号 特集 奥行カメラ Kinect で 3D 計測 第 2 章, pp. 62-72, 2012.
- [9] 浅見 典充, 河北 真宏, 白井 暁彦, 小林 希一, 滝沢 国治: 3 次元カメラの画像蓄積による距離検出分解能の向上と形状計測, 画像電子学会年次大会予稿集, Vol. 32nd, pp. 71-72, 2004.
- [10] Yuto Nara, Genki Kunitomi, Yukua Koide, Watar Fujimura, Akihiko Shirai: Manga Generator, Laval Virtual VRIC 2013, Article No. 29-7, 2013.
- [11] 長尾 智晴: C 言語による画像処理プログラミング入門, pp. 50-64, 2011.