

匠深 -TAKUMI- : 深度情報に基づく実時間 NPR システム

若間 弘典^{†1} 片寄 晴弘^{†1}

本稿ではカメラから得られた実画像に対して遠近感や立体感を残した NPR をリアルタイムに行うシステム「匠深-TAKUMI-」を提案する。これまで静止画に対してアニメ調表現を行う多くのシステムが提案されてきたが、画面全体に一律な輪郭線抽出やトーン単純化処理を実施するため、遠近感や立体感が損なわれる問題があった。本システムでは実画像と対応した深度情報を用いてこの問題を解決を図る。輪郭線の遠近の描き分けに深度情報に基づく平滑化フィルタを用い、トーン単純化に被写体の法線情報に基づく平滑化を行う。GPU を利用することでリアルタイムでの動作を実現した。本システムの利用により、実動画像からのエンタテインメントへの応用が期待される。

TAKUMI : Renderer for Animated Cartoon with Depth Map

HIRONORI WAKAMA^{†1} and HARUHIRO KATAYOSE^{†1}

This paper proposes a new non-photorealistic rendering method using a depth map, and introduces a real-time cartoon-like video converter called TAKUMI, based on the proposed method. here have been proposed many systems that convert an given image into non- photorealistic one. Images that these conventional methods renders lack in perspective, due to processing equally applied to the whole image. We propose a method that solve using a depth map. The proposed method generates cartoon-like jump edges based on depth information, and adjusts the tone color of region of the same plane considering the values of normal lines. TAKUMI, implemented the proposed method using Kinect, works at the frame rates or 30fps. We expect TAKUMI to be used in the fields of entertainment and content production.

^{†1} 関西学院大学 理工学研究科
School of Science and Technology, Kwansai Gakuin University

1. はじめに

カメラから得られた実画像や CG を用いて作成された写実的な画像を非写実的な画像に変換する技術はノンフォトリアリスティックレンダリング(以下 NPR と記述)と呼ばれている。NPR は従来人の手で行われてきた絵画的表現を半自動化することが出来るため、様々な分野でその技術応用が期待されている。これまでに水彩画や鉛筆画、イラスト画等の描画スタイルを表現する多くの NPR 手法が提案されてきた¹⁾ 本研究ではアニメ画を生成する NPR システムを取り扱う。なお、ここでいうアニメ画とは、輪郭や境界線をはっきりとした線で描き、色や影のグラデーションを単純化させて段階的に表現しているセルアニメ調のものを指す。

イラスト画の NPR 表現を行っている研究としては、浦浜ら²⁾ の実画像に対し自己商画像を用いる手法などが挙げられる。この手法によって生成された画像は、それぞれの描画スタイルを上手く表現できていたが被写体までの距離を考慮していないため、遠近感を再現することができなかった。アニメ画の NPR 表現を行っている研究としては Fisher ら³⁾ の研究がある。ここで実画像に対してバイラテラルフィルタ (以下 BF と記述) を用いることでトーンを簡略化する手法が提案されているが、実画像をアニメ画にする際に同色の異なる被写体が近接している時に色が混ざり合ってしまう問題があった。本研究では、実画像と対応した深度情報を利用することにより上記の 2 つの問題の解決を図る。深度情報を取得するデバイスとして Kinect を用い、負荷の大きい計算を GPU 内で処理することでリアルタイムで動作するアニメ表現システム「匠深-TAKUMI-」を提案する。

2. 手法の検討

本研究では、アニメ画らしさを表出するために以下のような表現の実施を目標として設定する。

- (1) 被写体までの距離に応じて描き分けを行う、はっきりとした輪郭線表現
- (2) 被写体の形状を考慮した、色や影のグラデーションの段階的な表現

以下、これらの目標を達成するために従来用いられてきた手法とその問題点を整理し、提案手法の概要について述べる。

2.1 NPR におけるアニメ画の生成手法と課題

(1) の目標に対して先述の浦浜らの自己商画像を用いた輪郭線抽出法を用いることで、アニメ画に適したはっきりとした輪郭線を得ることが出来る。しかし、彼らの提案手法は被写

体までの距離に関わらず画面全体に一律な処理を行うため、出力結果の遠近感が過度に失われてしまう問題がある。アニメ画の輪郭線に用いられる遠近の描き分け表現として、距離に応じた輪郭線の詳細さと濃淡の変化がある。一般に遠くの被写体ほど輪郭線が省略され、薄く描かれるようになる。これは大気遠近法と呼ばれる絵画技法を簡略化したものであり、輪郭線に遠近感を持たせるためにはこの二つの再現が重要であると考えられる。

アニメ画の段階的なトーン変化を実現するために従来様々な手法が提案されてきたが、多くの場合 BF が用いられてきた。BF は平滑化フィルタの一種であり、平滑化の際に輝度の変化量を考慮に入れた重み付けを行うことができるため、エッジを保持したままトーンを単純化する必要があるアニメ表現に適したフィルタである。しかし、BF は複数の面を持つ色変化の少ない被写体に対して、一つの面で被写体が構成されているかのような立体感の希薄な塗り分けとなってしまう問題がある。

2.2 本研究のアプローチ

本研究では、実画像と対応した現実環境の深度情報を用いることで、従来手法では失われがちな遠近感や立体感を残したアニメ表現を行うレンダリングシステムを開発する。本システムの処理概要を図 1 に示す。はじめに、カメラから得られた実画像と深度情報に対してノイズ低減処理を施す。次に、遠近感を残したアニメ画の輪郭線を生成するために、実画像に対して先に平滑化を行う。平滑化処理では、深度情報に基づいて重みづけを行った適応的平滑化フィルタを利用する。最後に、深度情報から求めた被写体の法線情報に基づく BF を適用することで立体感を残した色表現を行う。この処理により、従来手法の課題であった過度な平滑化を抑制することができる。

深度情報を得るデバイスとして Kinect を用い、計算負荷の大きい処理を GPU 内で行うことで、NPR 画像を実時間で生成することができる。本システムを用いることで従来手法より優れた NPR 表現を実時間で行うことができる。

3. 入力情報のノイズ低減処理

本章ではカメラから得られた実画像と深度情報に含まれるノイズの低減処理について述べる。実画像に対しては中央値フィルタを用いて平滑化を行い、深度情報に対しては動画像の時間軸を利用した平滑化フィルタを用いてノイズの影響を低減する。なお、本研究における目標の一つであるリアルタイムでの動作を実現するため、各ハードウェアの特性を考慮した実装を行っている。以下に具体的な処理手順を示す。

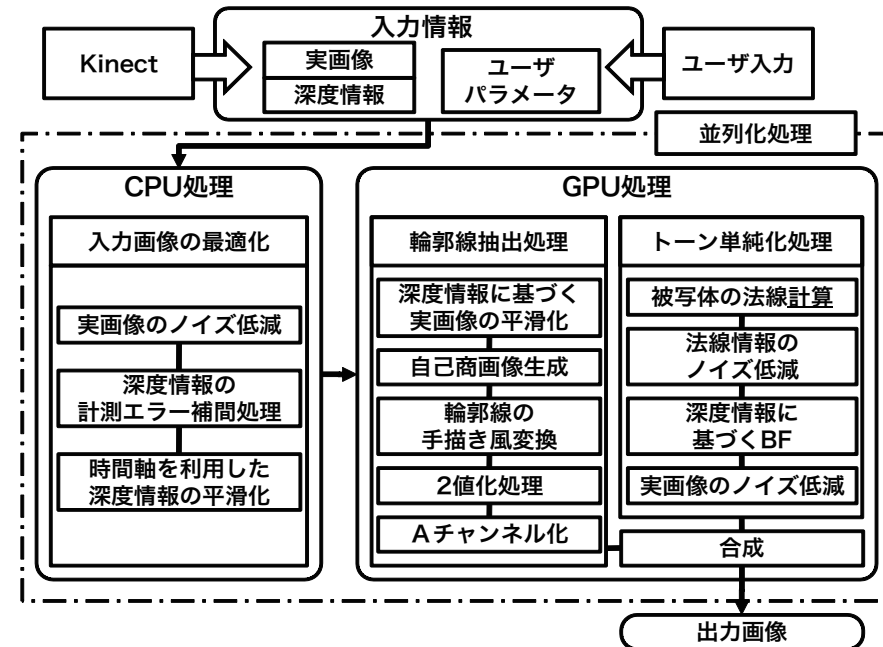
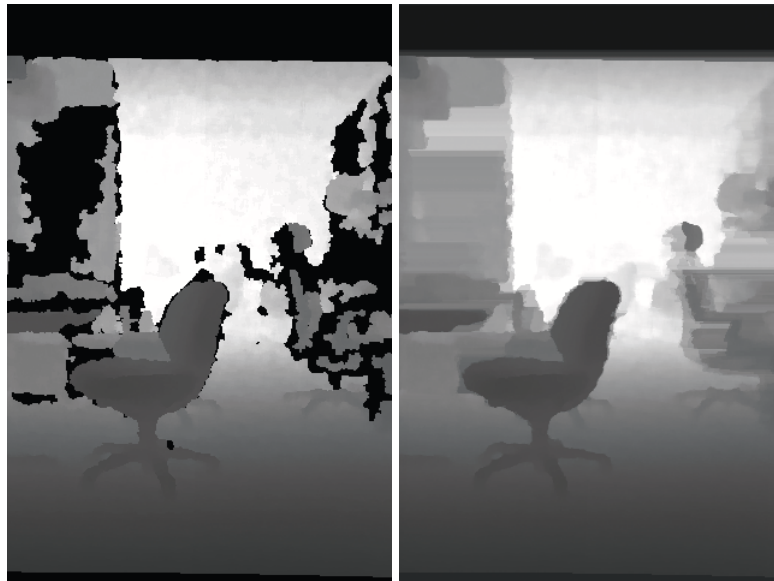


図 1 本システムの処理概要

3.1 実画像のノイズ低減処理

本システムではアニメ画特有の輪郭線、色表現を得るために BF を用いている。BF は先述の通りアニメ表現に適したフィルタであるが、エッジを保持する特性からカメラのセンサー等に起因するスパイク状のノイズを取り除くことができない。これらのノイズは後の処理工程において出力画像のちらつきや平滑化不良の原因となるため、可能な限り取り除く必要がある。そこで、まず実画像の平滑化を行う。本手法では平滑化フィルタとしてスパイクノイズに有効な中央値フィルタを用いた。

中央値フィルタは注目画素とその近傍の画素をソートする必要があり計算負荷が大きい処理である。一般に、全ての画素に対して同一の処理を行う場合 CPU よりも GPU を用いたほうが高速である。しかし、GPU では条件分岐が比較的高価な処理であるためフレームレートの低下を招く恐れがある。そこで、本研究では CPU を用いて中央値フィルタを適用



(a) Kinect から得られる深度情報

(b) 平滑化処理による補正後

図 2 深度情報に対する平滑化処理の比較

した。リアルタイム性を維持するため、後述の GPU 内の処理と並列してフィルタ処理を行う。処理を施した実画像は次フレームの GPU 処理を通じて画面に表示する。1 フレームの遅延が生じるが実用上問題はないと考えられる。

3.2 深度情報のノイズ低減処理

Kinect から得られた深度情報を伸長表示したものを図 2(a) に示す。この図から分かるように深度情報には計測誤差が多く含まれている。

このまま輪郭線抽出処理やトーンの簡略化処理に用いると深度情報の急激な変化によるちらつきが発生する可能性がある。特に Kinect の深度計測範囲から外れた領域は 0 が出力されるため、はじめにこの領域の補間を行う。本手法では、該当領域の周辺画素の平均を画素の値として適用した。次に、連続フレーム間において深度値の変化は小さいと仮定し、時

間軸を利用した BF を用いて平滑化を行う。平滑化に利用するフレーム数を多くするほどノイズを小さくすることが出来るが、視点の変化が大きい場合に残像の影響が大きくなるため、違和感の少ない値を求める必要がある。本研究ではフレーム数を 4、BF のガウシアン係数を 1、色差分係数を 0.1 とした。ノイズの低減を行った図を図 2(b) に示す。図 2(a) に比べてノイズが減少していることが確認できる。

4. 遠近感を残したアニメ画の輪郭線表現

本章では輪郭線の遠近の描き分けを行うための輪郭線抽出法とその実装について述べる。まず、深度情報に基づいた平滑化フィルタを用いて輪郭線を抽出する。次に、手描きに近いなめらかな輪郭線を得るために非等方性 BF を適用する。本手法は全て GPU 内で実装を行っている。以下に具体的な処理手順を示す。

4.1 深度情報に基づく輪郭線生成

遠近感を伴ったアニメ画の輪郭線を生成するためには、距離に応じて輪郭線の詳細さや太さ、濃淡を変化させる必要がある。つまり、遠くの被写体であるほど輪郭線を少なく薄く描画する。最初に細部の描き込みを変化させる手法について述べる。浦浜らの提案手法である自己商画像を用いた輪郭線抽出法は、実画像の画素の値を平滑化を行った画像の画素の値で割り、画素値の変化の激しいエッジ部の値を大きくすることで輪郭線の抽出を行っている。そこで、実画像に対して先に深度値を重み係数として掛け合わせた平滑化処理を行い、遠くの被写体であるほど詳細な輪郭線が抽出されないようにする。その後、浦浜らの提案手法を用いて輪郭抽出を行う。次に、距離に応じて線の太さを変更する手法について述べる。自己商画像に対して閾値処理を行うことで検出される輪郭線の太さは、自己商画像の強調されたエッジ部の幅に比例する。そのため、線の太さを変更するには浦浜らの提案手法で用いる平滑化画像を生成する際に、深度情報に基づいた平滑化フィルタを利用することで実現できる。本研究では、平滑化フィルタとしてガウシアンフィルタを用いて自己商画像を作成した。最後に輪郭線の濃淡の違いを再現する手法について述べる。これは深度情報を透過度を示すチャンネルとして用いることで、遠くに存在する被写体の輪郭線は薄く、近くに存在する被写体の輪郭線は濃く描くことができる。

4.2 輪郭線の手描き風変換処理

前節の手法で生成された輪郭線は不要な線や孤立点が多数存在しており、また輪郭線の角のエッジが立っているためアニメ画にはあまり適していない。そのため、よりアニメ画に適した輪郭線を得るために BF を用いて平滑化を行う。BF は以下の式で表される。

$$h(x, y) = \frac{\sum_{i=-W}^W \sum_{j=-W}^W d_2 \exp\left(-\frac{(i^2 + j^2)}{2\sigma_1^2}\right) \exp\left(-\frac{(d_1 - d_2)^2}{2\sigma_2^2}\right)}{\sum_{i=-W}^W \sum_{j=-W}^W \exp\left(-\frac{(i^2 + j^2)}{2\sigma_1^2}\right) \exp\left(-\frac{(d_1 - d_2)^2}{2\sigma_2^2}\right)} \quad (1)$$

BF を複数回用いることでノイズの少ないなめらかな輪郭線を得ることができる。しかし、BF はカーネルサイズに比例して計算量が増加するため、実時間で処理を行うためには計算量を減らす必要がある。そこで、本研究では BF による平滑化を行う前に微分フィルタを用いて領域ごとに最適な平滑化方向を求めることで計算量の削減を試みる。以下が改良した非等方性 BF の式である。

$$h(x, y) = \frac{\sum_{i=-W}^W d_2 \exp\left(-\frac{((iv_x)^2 + (iv_y)^2)}{2\sigma_1^2}\right) \exp\left(-\frac{(d_1 - d_2)^2}{2\sigma_2^2}\right)}{\sum_{i=-W}^W \exp\left(-\frac{((iv_x)^2 + (iv_y)^2)}{2\sigma_1^2}\right) \exp\left(-\frac{(d_1 - d_2)^2}{2\sigma_2^2}\right)} \quad (2)$$

自己商画像に対して閾値処理を行う前に Sobel フィルタを用いて画素ごとの勾配ベクトルを求め、この勾配ベクトルの直角方向に対して 1 次元の BF を用いることで効率の良い平滑化を行うことができる。また、2 つの輪郭線が交差する箇所においては、なす角に対して直角に平滑化を行うので輪郭線のエッジの角をとる働きがある。図 3 に平滑化前後の輪郭線を示す。孤立点が減少し、丸みのある手描きに近いタッチの輪郭線が生成されている。

5. 立体感を残したアニメ画の色表現

本章では被写体形状に合わせた色表現を行うための平滑化フィルタとその実装方法について述べる。まずはじめに現実環境の深度情報から被写体の法線を求め、次に実画像に対してその法線情報に基づいた平滑化を施す。本処理についても前章と同様、リアルタイム性を維持するために処理の大半を GPU 内で行っている。以下に具体的な処理手順を示す。

5.1 被写体表面の法線計算

深度カメラから得られた深度情報を基に被写体表面の法線の計算を行う。カメラの射影変換のモデルとして正射影を用いた場合、現実環境の視体積の X, Y, Z 軸はそれぞれ実画像の X, Y 軸と深度値に対応付けることができる。この時、ある注目画素の法線方向は注目画素と近傍 2 画素の 3 次元座標から外積を用いることで求められる。本研究では、ノイズ



(a) 平滑化処理を行う前

(b) 非等方性 BF による修正後

図 3 輪郭線に対する平滑化処理の比較

の影響を小さくするため近傍 8 画素を用いて法線の計算を行っている。

以上により計算された法線には計測誤差が含まれる。これを解消するため、以下の式による平滑化を実施する。

$$\vec{h}(x, y) = \frac{\sum_{i=-W}^W \sum_{j=-W}^W \vec{v}_2 \exp\left(-\frac{(i^2 + j^2)}{2\sigma^2}\right) \exp\left(-\frac{(\vec{v}_1 \cdot \vec{v}_2)^2}{2\sigma^2}\right)}{\sum_{i=-W}^W \sum_{j=-W}^W \exp\left(-\frac{(i^2 + j^2)}{2\sigma^2}\right) \exp\left(-\frac{(\vec{v}_1 \cdot \vec{v}_2)^2}{2\sigma^2}\right)} \quad (3)$$

ここで、 \vec{v}_1 は注目画素の法線ベクトル、 \vec{v}_2 は参照画素の法線ベクトルである。この式により、法線の向きが近い画素間では大きい重みが与えられ、その逆では、小さな重みが与えら

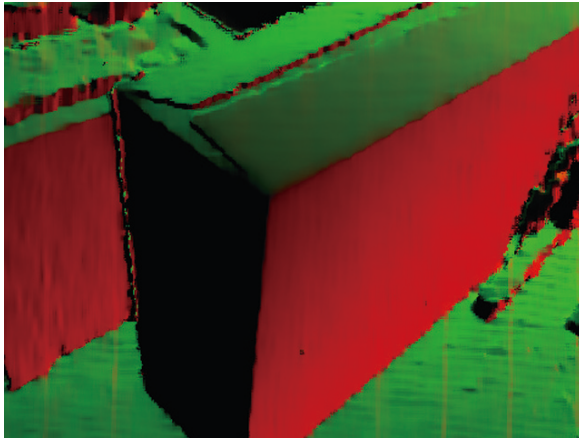


図4 法線の平滑化処理

れる．この式を用いて法線の平滑化を行なった様子を図4に示す．法線の向きが揃えられているにもかかわらず，その向きが大きく変化する被写体のエッジ部において変化が抑えられていることがわかる．

5.2 法線情報を用いた平滑化フィルタ

前節で求めた法線を用いて平滑化を行う．BF に対して注目画素と参照画素間での法線の向きが大きく異なれば平滑化時の重みが小さくなる項を追加する．以下に用いた式を示す．

$$h(x, y) = \frac{\sum_{i=-W}^W \sum_{j=-W}^W d_2 w_1(i, j) w_2(d_1, d_2) w_3(\vec{v}_1, \vec{v}_2)}{\sum_{i=-W}^W \sum_{j=-W}^W w_1(i, j) w_2(d_1, d_2) w_3(\vec{v}_1, \vec{v}_2)} \quad (4)$$

$$w_1(m, n) = \exp\left(-\frac{(m^2 + n^2)}{2\sigma_1^2}\right) \quad (5)$$

$$w_2(d_1, d_2) = \exp\left(-\frac{(d_1 - d_2)^2}{2\sigma_2^2}\right) \quad (6)$$

$$w_3(\vec{v}_1, \vec{v}_2) = \exp\left(-\frac{(\vec{v}_1 \cdot \vec{v}_2)^2}{2\sigma_3^2}\right) \quad (7)$$

v_1 は注目画素の法線ベクトル， v_2 は参照画素の法線ベクトル， σ_3 はそれら二つの法線の成す角度の分散であり，その他の値は BF と同様である．BF では異なる被写体であっても輝度値が近ければ同一領域として扱われてしまう問題点があった．しかし，法線の向きを重みとして組み込んだこの式を用いた場合，輝度値が近い場合においても法線の向きが異なれば平滑化を抑え，エッジとして保持することができる．

6. 実施例

本研究で提案したシステムがリアルタイムで動作していることを述べ，その効果をスナップショットを用いて検証する．本検証で用いたハードウェア構成は Intel CPU Core i7 2.93Ghz，メモリ 3GB，GPU は Nvidia Quadro CX である．上記の構成で本手法を用いたシステムはフレームレート 30fps で動作することを確認した．以下に図5を入力画像として用いた結果を示す．

6.1 輪郭線抽出法の比較

図6(a)はアニメ画の輪郭線抽出手法の比較である．図6(a)左の従来手法を用いた画像では被写体までの距離によらず画像全体に一律な輪郭線が現れており，変化に乏しく遠近感をあまり感じさせない結果となっている．図6(b)右の提案手法を用いた画像では，被写体までの距離に応じて輪郭線の密度と濃淡に変化があり，従来手法に比べ遠近感を感じさせる結果が得られている．濃淡の変化が遠近感の獲得に特に大きな影響を与えていると思われる．

6.2 トーン単純化手法の比較

図6(b)はアニメ画におけるトーンの単純化手法の比較である．図6(b)左の従来手法を用いた画像では，本棚の異なる面が平滑化によって繋がってしまっており，立体感に欠ける結果となっている．図6(b)右の法線情報を重みに加えた提案手法では，エッジ部分が消失せず立体感のある結果が得られている．このことは面における法線の向きの異なりによって平滑化が抑制されることを示しており，本研究の提案手法が正しく機能していることがわかる．

なお，これらの出力を合成した最終結果を図7に示す．

7. まとめ

本研究では，実画像と対応した深度情報を利用することにより上記の2つの問題の解決手法を提案した．深度情報を取得するデバイスとして Kinect を使い，負荷の大きい計算を GPU 内で処理することでリアルタイムで動作するアニメ表現システム「匠深-TAKUMI-」

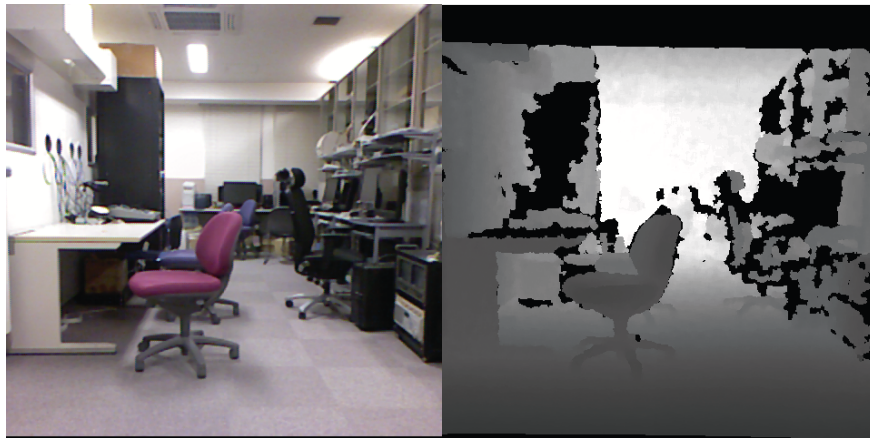
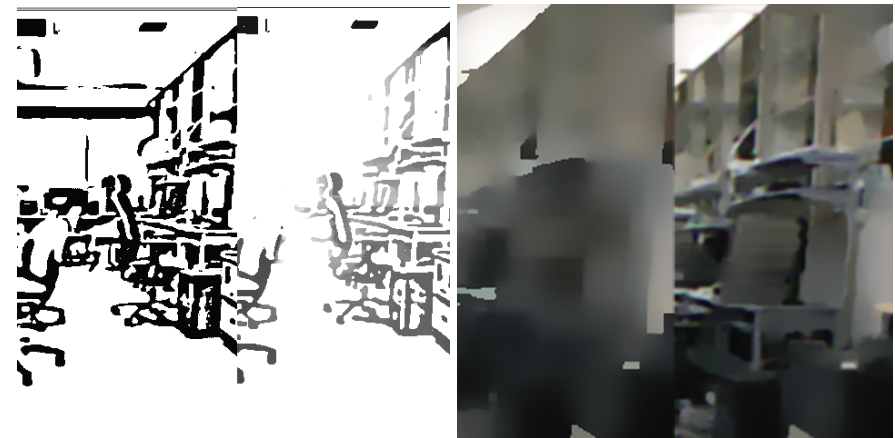


図 5 入力として用いた実画像と深度情報



(a) 輪郭線抽出手法の比較

(b) 色表現手法の比較

図 6 従来手法と提案手法の比較

を開発した。NPR 表現における従来のアニメ画生成手法に対して、深度情報を用いることで遠近感や立体感を付加することが可能であることを示した。深度情報を用いた輪郭線の抽出手法と深度情報から得られた法線情報に基づくトーンの単純化手法が有用であることを確認した。本手法は極めて汎用性が高く他の絵画モデルを用いた手法に対しても応用可能である。今後は鉛筆画や水彩画、油彩画等の表現技法において遠近感や立体感を獲得する手法を模索していきたい。

参 考 文 献

- 1) 茅 暁陽, 長坂 好恭, 山本 茂文, 今宮 淳美: LIC 法を利用した鉛筆画の自動生成, 芸術科学会論文誌, Vol. 1, No. 3, pp.147-159, 2002 .
- 2) 景, 井上, 浦浜: 自己商画像の非等方平滑化に基づくイラスト風画像の生成, 電子情報通信学会論文誌, J89-A, 5, pp.385-386, 2006 .
- 3) Jan Fischer, Dirk Bartz: Real-time cartoon-like stylization of AR video stream on the GPU. Technical Report WSI-2005-18, Wilhelm Schickard Institute, 2005

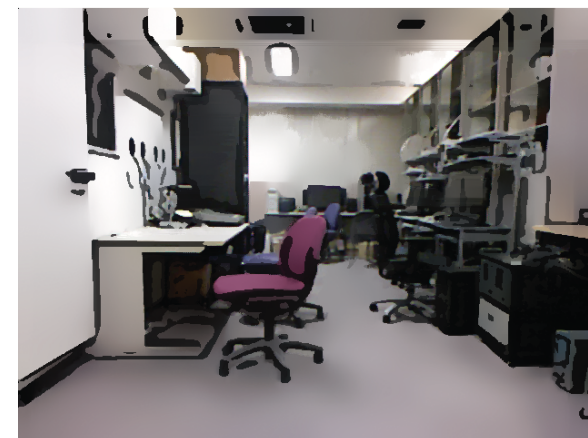


図 7 提案手法の出力結果