

エッジ検出のための画像のブロック分割によるグレースケール変換

松村 将太郎[†] 熊澤 逸夫[†]

[†] 東京工業大学 像情報工学研究施設 〒 226-8503 神奈川県横浜市緑区長津田町 4259 R2 棟 330
E-mail: †{matsu-sho,kumazawa}@isl.titech.ac.jp

あらまし FA などの画像処理では、グレースケール画像を用いて処理を行っている。しかし、グレースケール変換は 3 次元のカラー情報を 1 次元の輝度情報にする次元圧縮問題であるため、カラー画像では存在するはずのエッジ情報がグレースケール画像では失われてしまうことがある。本研究では、カラー画像を微小ブロックに分割し、ブロック毎にグレースケール変換係数を求めることで、グレースケール画像においてもエッジ情報を保存するためのグレースケール変換手法を提案する。さらにブロック別に不連続なルールを適用する際に生じるブロック歪みを抑制する方式についても示す。

キーワード グレースケール変換, 次元圧縮, ブロック分割, エッジ検出, ブロック歪み

Grayscale Conversion by Block Segmentation with Continuous Boundary Conditions

Shotarou MATSUMURA[†] and Itsuo KUMAZAWA[†]

[†] Imaging Science and Engineering Laboratory, Tokyo Institute of Technology
R2-330, 4259 Nagatsuta-tyo, Midori-ku, Yokohama-shi, Kanagawa, 226-8503 Japan
E-mail: †{matsu-sho,kumazawa}@isl.titech.ac.jp

Abstract The grayscale image is used by image processing such as FA. However, grayscale conversion is dimensionality reduction problem that color information is converted to luminance information. Therefore edge information might be lost in grayscale image. This research proposes the grayscale conversion technique to preserve edge information by dividing an image into small blocks and obtains coefficient of grayscale conversion at every block. In addition, a method which restrains the block distortion by preserving continuity along the block boundaries, is shown.

Key words Grayscale Conversion, Dimensionality Reduction, Block Segmentation, Edge Detection, Block Distortion

1. まえがき

現在、FA (Factory Automation) など使われる業務用画像処理ソフトウェアの多くは、グレースケール画像で処理を行っている。またそれらの画像処理では、特徴量としてエッジを用いている場合が多い。しかし、既存のグレースケール変換では、色で判断すると境界を判断できる画像 (図 1(a)) でも、従来のグレースケール化手法では、その境界の情報が失われることがある (図 1(b))。これは、グレースケール変換が、三次元の色情報を一次元の情報に圧縮するために生じる問題である。つまり、カラー画像をグレースケール画像に変換することは、次元圧縮問題と考えることができる。ここで、効率よく次元圧縮する手法として、PCA や LLE [1] などが知られている。

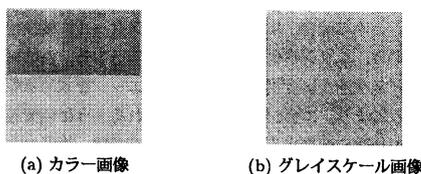


図 1 グレースケール変換による情報の欠落
Fig. 1 The lack of information by grayscale conversion

グレースケール変換に関する研究として、単色性色覚者に対して、カラー画像の細部の情報まで提示するための研究 [2] が行われている。また最近では、人間の知覚に基づいて、グレースケール変換を行う "Color2Gray" というアルゴリズムが提

案された [3]。この手法は、人間の知覚特性に最も近いとされる $L^*a^*b^*$ 色空間において、ピクセル間の輝度差を色空間の輝度と色差をもとに算出し、実際のグレースケール画像においても同じ輝度差となるようにグレースケール画像を求めている。この手法を用いると、カラー画像の細部の情報を正確に保存できる。しかし、この手法の問題点として、多量の計算時間が必要なが挙げられる。このアルゴリズムは、 S を画像サイズとする時に、 $O(S^4)$ という計算量であるため、高速な処理が要求される画像処理では使用することができない。そこで本研究では、カラー画像の細部の情報を保存しつつ、高速なグレースケール変換を求めめることを目的とする。

本研究では、グレースケール変換の高速化を図るため、グレースケール変換式に着目した。NTSC 規格におけるグレースケール変換式は、次式で与えられ

$$L = 0.299R + 0.587G + 0.114B \quad (1)$$

画像全体で同じ変換式を用いている。本来多彩な色を含むカラー画像に均一ルールを適用することには無理があり、均一ルールを用いる限り改善できないと考えられる。そこで本研究では、画像を微小ブロックに分割し、ブロック毎にグレースケールへの変換係数を求めることで、カラー画像の情報がなるべく失われないようなグレースケール画像を求める方法を提案する。またこのようにブロック別に不連続なルールを適用する際に生じるブロック歪みを抑制する方式も示す。

2. 色情報の利用

2.1 FA における画像処理

FA において現在実用化されている技術の多くはグレースケール画像を用いたものである。しかしながら、カラー画像を利用し、新たに色情報を加えることで精度や速度の向上を期待することができ、これまでも多くの研究がなされてきた。

画像の照合、検索への色情報の用い方としては、色を直接的に用いるか、間接的に用いるかの2つの方法が考えられる。色情報を直接的に用いる方法としては色ヒストグラムを用いた方法が代表的である。しかし、色情報を数値として扱うコンピュータでは、照明や影による小さな色の変化も異なる色として判断してしまうため、照合、検索精度の悪化を招いてしまい、精度を向上させるためにはテンプレートとして多くの登録画像を用いる必要があった。一方、色情報を間接的に用いる方法では、画像の照合、検索自体はグレースケールに変換後の画像あるいは、エッジ情報を特徴量として実行する。色情報はこのグレースケールに変換する、あるいはエッジを求める際に利用する。つまり、エッジが明確に検出されるように色情報をモノクロの情報に変換するのである。本研究では、後者の方法により新たなグレースケール変換手法を提案する。

2.2 本研究における色情報の利用

グレースケール変換は、情報の損失を伴う変換手法であるため、色情報を直接用いる手法以上の精度にすることは不可能である。よって、グレースケール変換を用いる場合、どのような情報が必要なのかを決め、その情報だけを抽出するような変換

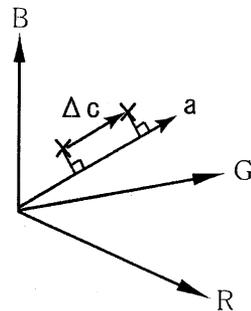


図2 色空間における射影
Fig.2 The projection in color space

ルールを求めることが望ましい。そこで本手法では、エッジ情報をグレースケール画像に残すような変換ルールを求める方法を提案する。

グレースケール画像におけるエッジとは、隣り合う画素間での輝度値の変化が大きい部分であり、カラー画像におけるエッジとは、色相または明度の変化の大きい部分と見ることができる。また、式(1)におけるグレースケール変換式は、係数ベクトルを $\mathbf{a} = (a_r, a_g, a_b)^T$ 、色ベクトルを $\mathbf{c} = (R, G, B)^T$ とおくと、次のようにベクトルの内積で表される。

$$L = a_r R + a_g G + a_b B \quad (2)$$

$$= \mathbf{a}^T \mathbf{c} \quad (3)$$

さらに、エッジ境界付近における色ベクトルの変化量を $\Delta \mathbf{c} = (R_1 - R_2, G_1 - G_2, B_1 - B_2)^T$ とすると、

$$L = \mathbf{a}^T \Delta \mathbf{c} \quad (4)$$

と書ける。つまり、2つのベクトルの方向が等しい、または逆方向であれば内積の値が最大になることから、カラー画像におけるエッジをグレースケール画像にうまく反映させるには、色空間において、色相または明度の変化の大きい方向に係数ベクトルの方向を決めればよい(図2)。しかし、カラー画像全体で見ると、いくつものエッジによって領域が分割されているため、その係数ベクトルを一意に決めることはできない。

そこで本研究では、画像を微小ブロックに分割することで、それぞれのブロック毎に、最適な係数ベクトルを求める。画像を微小ブロックに分割することで、エッジの存在するブロックでは、高々2つのクラスに分けられると仮定できる。従って、ブロック毎にそのブロックのエッジを明確にするための係数ベクトルを一意に求めることができる(図3)。

射影する軸を決める方法として、本手法では、k-means 法により2クラスの重心を求め、その重心を通る直線を射影軸としている(図4)。

3. 提案手法

以上の考えに基づいた新たなグレースケール変換、およびエッジ検出までの流れを図5に示す。3.1では、各ブロック毎

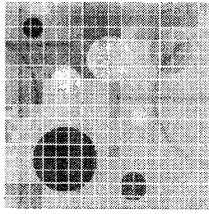


図3 画像の分割
Fig. 3 Image segmentation

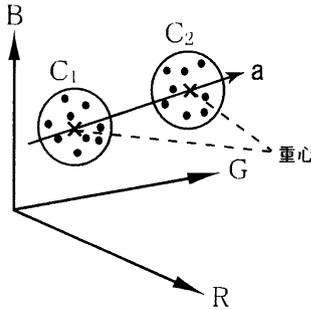


図4 微小ブロックの色空間
Fig. 4 Color space in blocks

に新たなグレースケールの変換ルールを求める方法について述べる。3.2では、前半部で生じるブロック歪みを抑制方法について述べる。最後に3.3では、ブロック境界を移動することによりブロック境界のエッジを除去する方法について述べる。

3.1 前半部

前半部におけるアルゴリズムを以下に示す。

- Step1 カラー画像を微小ブロックに分割する
- Step2 k-means 法で2クラスに分ける
- Step3 射影する軸の方向を決め変換ベクトルを求める

まず、画像を微小分割することにより、色空間内の点の分布を2クラスに分けられると仮定できるほどのブロックサイズに分割し、k-means法を用いて2クラスの重心を求める。次に、2つの重心を通る直線を決める際に、2通りの直線の方向が考えられるため、2クラスの重心を式(1)で、一度輝度値に変換し、輝度値の低いクラスから高いクラスの方に射影する軸を決める(図6)。最後に、エッジの存在しないようなブロックでは、適切な変換ベクトルを求めることができないため、ある閾値を設け、クラス間の距離が閾値以下ならば、そのブロックの変換ベクトルは求めず、係数ベクトルは式(1)のままとする。

前半部までの処理を行った画像を図7(h)に示す。このように、ブロック別に不連続な変換ルールを適用する弊害として、ブロック歪みが生じるという問題がある。そのため、ブロック歪みを抑制する方法が必要である。

3.2 後半部

前半部で生じるブロック歪みの問題を解決するため、後半部

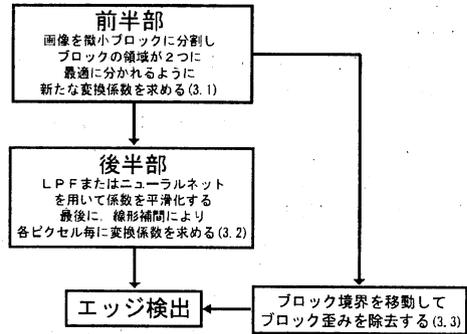


図5 提案手法の流れ図
Fig. 5 Flow chart

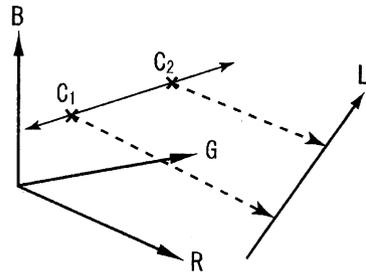


図6 射影軸の軸方向
Fig. 6 Direction of projection axis

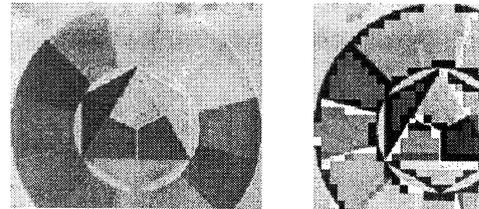


図7 ブロック歪みの影響
Fig. 7 The problem of block distortion

ではブロック歪みを抑制するための処理を行っている。ブロック歪みが生じるのは、ブロック間で変換係数が不連続に変化するためである。そこでブロック歪みを抑制する方法として、変換係数を局所的なLPFによって平滑化する方法とニューラルネットワーク(以下NN)による平滑化を提案する。局所的なLPFとして、8近傍ブロックの平均値フィルタを用いた。またNNは、ブロック間で係数が連続になるように次式のエネルギー関数を定義し、最適解を求めている。

$$E = \frac{A}{2} \sum_i \sum_{j \in \mathcal{L}} (a_i - a_j)^2 + \frac{B}{2} \sum_{k \in \mathcal{K}} (a_k - a_{ok})^2 \quad (5)$$

ただし、 N :ブロック数、 \mathcal{L} :注目ブロックの近傍ブロック、 \mathcal{K} :新

たに変換係数を求めたブロック, a_{ok} :新たに求めたブロックの係数である。

式(5)の第一項は、近傍ブロック間の係数が滑らかにするための項あり、第二項は、新たに求めた変換係数を平滑化後も残すための項である。しかし、上述の方法により平滑化を行っても、完全にブロック歪みを抑制することはできない。そこで、ブロック間の境界においてもなるべく変換係数が連続になるように、ブロックの変換係数をブロックの中心のピクセルの変換係数とし、バイリニアの線形補間により、最終的に各ピクセル毎に変換係数を求め、グレースケール画像を生成する。

3.3 ブロック境界を考慮したエッジ検出

ここでは、前半部で求めたグレースケール画像から直接エッジ検出を行う方法について述べる。ブロック歪みが生じるのはブロック毎に変換ルールが異なるためである。よって、各ブロック毎にエッジを検出すればブロック歪みによるエッジを考慮しなくてもよい。

また、ブロック境界にエッジが生じることを利用し、ブロック境界を移動すれば境界のエッジもその分移動することから、移動前と移動後の結果を利用してエッジ検出を行うことが出来る。まず、図8(a), (b)に示す移動前と移動後のエッジパターン、さらに、境界に生じるエッジパターン図8(c), (d)を用意する。ブロック境界のエッジを除去するためには、図8(a), (b)の画像に対して、境界部分のエッジを図8(c), (d)のブロック境界に生じるエッジパターンによってマスキングし、残ったエッジを合成すればよい。したがって、境界移動前のエッジパターン、移動後のエッジパターンをそれぞれ、 x_1, x_3 、移動前後のブロック境界のエッジパターンをそれぞれ、 x_2, x_4 、 $x_i \in \{0, 1\}$ (0:エッジなし, 1:エッジあり)、とすると、ブロック境界のエッジを除いた後の画像は、次式の論理演算により求めることが出来る。

$$f = (x_1 \wedge \bar{x}_2) \vee (x_3 \wedge \bar{x}_4) \quad (6)$$

4. 実 験

実験では、従来手法、提案手法、Color2Grayアルゴリズムについて、それぞれグレースケール変換を行った結果について述べる。さらに、それぞれのグレースケール画像に対しエッジ検出を行った結果について述べる。実験には、Intel(R)Pentium(R)4 processor 2.60GHzのPCを使用した。

4.1 グレースケール変換

図9(a)の画像(画像サイズ256*256pixel)に対して、それぞれの手法を適用した結果を図9(b)~(e)に示す。提案手法でのブロックサイズは、8*8pixelとした。

従来手法のグレースケール変換では、境界部分が欠落しているのに対し、提案手法では、境界部分が保存されている。ただし、後半部でLPFを用いた場合は、ブロック間の平滑化がうまく行われていないが、NNを用いた場合は、平滑化がうまく行われていることが分かる。LPFでの平滑化がうまく行われなかったのは、変換係数を求めることができなかったブロックの割合が多かったため、求めることができたブロックとできな

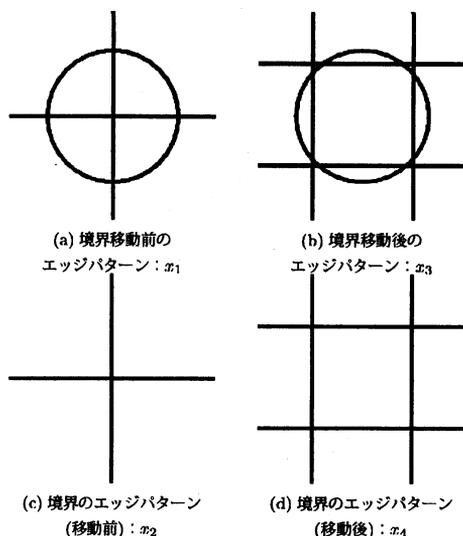


図8 エッジパターン

Fig. 8 Edge pattern

表1 計算時間

Table 1 The calculation cost		
画像サイズ [pixel]	提案手法	Color2Gray
64*64	15[msec]	1.2 [sec]
128*128	61[msec]	214 [sec]
256*256	249[msec]	6970 [sec]

かったブロックの係数差が非常に大きくなったためだと考えられる。また、提案手法とColor2Grayアルゴリズムを比較すると、Color2Grayの方が、カラー画像の細部の情報を正確に保存できているが提案手法とそれほど大きな違いは見られなかった。

画像サイズ毎にグレースケール変換に要した計算時間を表1に示す。提案手法は、前半部と後半部の処理を合わせた計算時間である。Color2Grayは、画像サイズが小さければ短時間で結果を得られるが、大きなサイズになるほど、多量の時間を要することが分かる。それに対し、提案手法ではColor2Grayよりも短時間で結果を得られている。

4.2 エッジ検出

図10のグレースケール画像に対して、エッジ検出を行った結果を示す。エッジ検出にはCannyエッジ検出法を用いた。従来手法によるグレースケール画像に対しては、カラー画像では違う色で分かれている領域が、同じ輝度値になっている領域があるために、大部分のエッジが検出できていないが、提案手法に対するエッジ検出では多くの境界エッジを検出することができている。また、Color2Grayでの画像でも提案手法と同等のエッジ検出結果が得られている。

図11では、前半部でブロック歪みが生じている画像に対して、ブロック境界のエッジを考慮してエッジ検出を行った結果を示す。ブロック毎にエッジ検出を行う方法、境界を移動させ

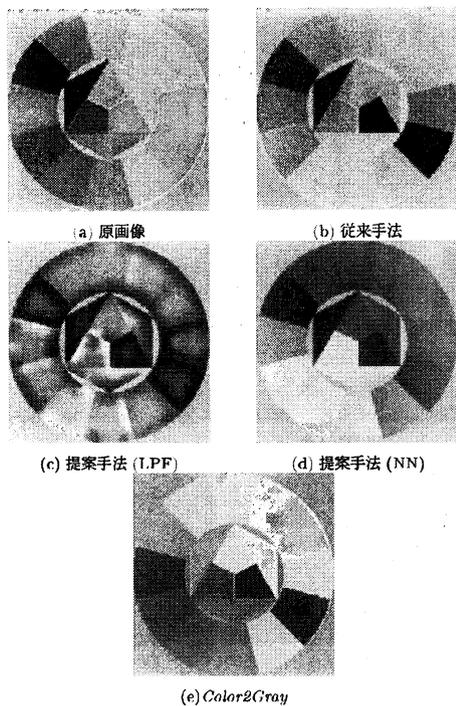


図9 グレイスケール変換

Fig.9 The result of grayscale conversion

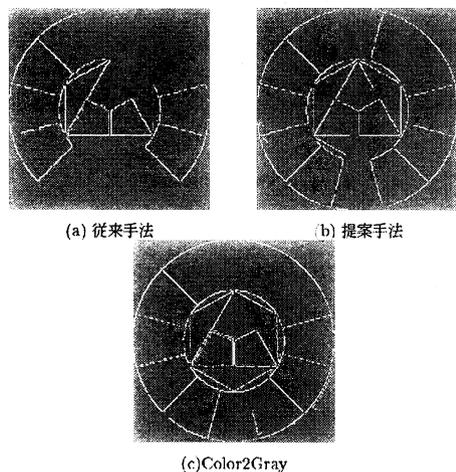


図10 エッジ検出

Fig.10 The result of edge detection

エッジ検出を行う方法のどちらでも、ブロック歪みによるエッジの誤検出は見られなかった。しかし、ブロック毎にエッジ検出を行う方法は、ブロック境界上にブロック歪みによるエッジ以外があったとしても、そのエッジは検出されないが、プロッ

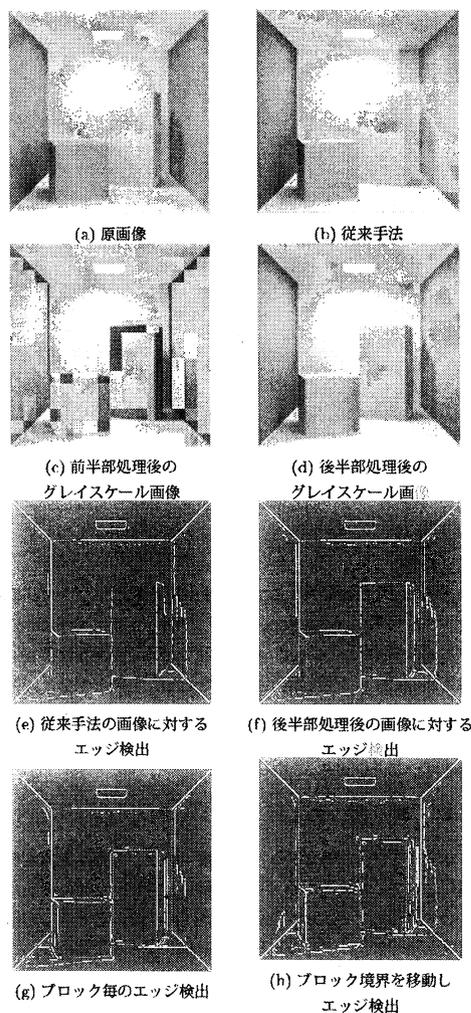
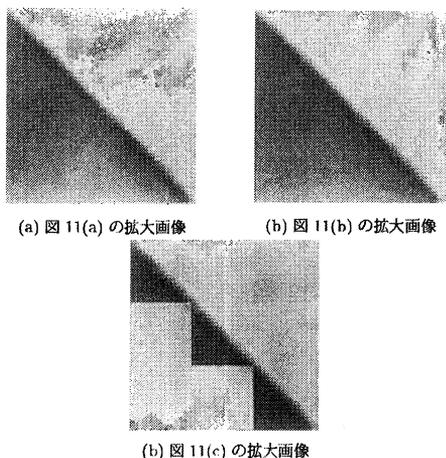


図11 ブロック歪みの生じた画像に対するエッジ検出

Fig.11 Edge detection for block distortion image

ク境界を移動させエッジを検出を行う方法は、そのようなエッジも検出できる。しかし、後者は二度グレイスケール変換を行うので、多くの計算時間を必要とする。また、従来手法および後半部の処理後にエッジ検出を行った画像と比較すると、前半部の処理後にエッジ検出を行った方が、細かいエッジも検出できている。これは、平滑化を行うことでブロックの領域を2つに最適に分かれるように選んだ軸がずれてしまうためである。さらに、図11(e), (h)をよく見ると、逆に提案手法では、消えてしまうエッジがある。原画像を拡大した図12(a)を見ると、実際は3つの領域に分けられるが部分が、提案手法(図12(c))では、2つの領域に分けるように最適な軸を求めているため、2つのクラスを1つのクラスと見なしてしまうためであると考



(a) 図 11(a) の拡大画像 (b) 図 11(b) の拡大画像



(c) 図 11(c) の拡大画像

図 12 3 クラスに分かれるブロック

Fig. 12 The block which separates into three classes

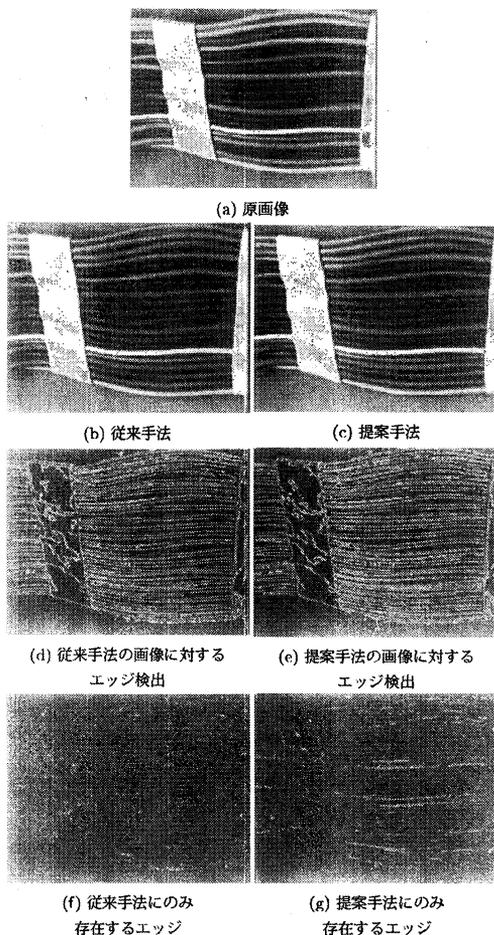
えられる。

最後に、実画像に対するエッジ検出結果を図 13 に示す。画像は VGA サイズ (640*480pixel) で、ブロックサイズは 8*8pixel である。図 13(f), (g) は、提案手法でのエッジ検出結果から従来手法でのエッジ検出結果の差分をとったものである。両方を比較すると、提案手法のグレイスケール画像は、従来手法のグレイスケール画像では検出できない連結性のあるエッジを検出できているといえる。

5. まとめ

本研究では、グレイスケール変換でエッジ情報が欠落する問題について、画像をブロックに分割し、ブロック毎に変換ルールを求めることでエッジ情報の欠落を最小限にとどめる方法を提案した。また、ブロック毎に不連続なルールを適用することによるブロック歪みの影響を、係数の平滑化によって抑制した。実際に、本手法を適用した画像に対しエッジ検出を行うと従来手法では、検出されなかったエッジを検出することができることを確認した。さらに、ブロック歪みの生じた画像に対し、ブロック境界を移動させることでブロック歪みによるエッジを除去する方法を提案した。この手法では、変換係数を平滑化し、エッジを検出した結果よりも細かいエッジを検出できることを確認した。変換に要する時間については、画像サイズが大きいとその分多くの計算時間を要するが、Color2Gray アルゴリズムと比較すると大幅に計算コストを抑えることができる。また、提案手法に要する計算時間の多くは、k-means 法による計算コストである。しかし、本手法では、ブロック毎に独立に変換ルールを求めており、並列処理が容易であることから、さらに計算時間を短縮することが可能であると考えられる。以上から、グレイスケール画像を用いる画像処理などに、本手法を適用することで、精度を向上できる可能性があると考えられる。

本手法の問題点として、微小ブロック内の領域が 3 つに分か



(f) 従来手法にのみ存在するエッジ (g) 提案手法にのみ存在するエッジ

図 13 実画像に対するエッジ検出

Fig. 13 Edge detection for real image

れる場合、検出されなくなってしまうエッジがある。この問題は、ブロックサイズを小さくすることで解決できる場合があるが、ブロックサイズが小さすぎると、ブロック歪みによるエッジによって、エッジ検出自体が難しくなる。そこで、今後の課題として、ブロック内の領域が 3 つに分けられる場合でも、正しくエッジを検出する方法について検討したいと考えている。

文 献

- [1] Sam T. Roweis and Lawrence K. Saul: "Nonlinear Dimensionality Reduction by Locally Linear Embedding", Science, Vol.290, pp2323-2326, 2000
- [2] Karl Rasche, Robert Geist and James Westall: "Detail Preserving Reproduction of Color Image for Monochromats and Dichromats", IEEE Comput. Graph., pp22-30, 2005
- [3] Amy A. Gooch, Sven C. Olsen, Jack Tumblin and Bruce Gooch: "Color2Gray:Saliency-Preserving Color Removal", In Proc. of ACM SIGGRAPH, pp.634-639, 2005