

画像圧縮ノイズ低減におけるラインメモリ削減技術

谷田部 祐介^{1,a)} 小味 弘典¹

受付日 2013年12月19日, 採録日 2014年6月24日

概要: 画像圧縮により発生するモスキートノイズとブロックノイズを低減させる機能をハードウェアで実装する際の合理化技術を開発した。我々は、これらの画像圧縮ノイズ低減に適用するフィルタタップ数の違いに着目し、フィルタ対象画像を保持するラインメモリの共有化を行うことでSRAMのbit量削減を実現した。これにより、本機能を実装するLSIの単価軽減や、メモリ削減分を他信号処理に割り当てることによるLSIの性能向上を可能とした。

キーワード: ラインメモリ, 画像圧縮, モスキートノイズ, ブロックノイズ

Line Memory Reduction Technology Suitable for Compression Artifacts Suppression

YUSUKE YATABE^{1,a)} HIRONORI KOMI¹

Received: December 19, 2013, Accepted: June 24, 2014

Abstract: An efficient hardware implementation method suitable for reducing mosquito and block noise caused by image compression has been developed. After analyzing the length of the filter taps necessary for noise reduction, we optimized line memories used for buffering pixels in the filtering process, and achieved a reduction in the amount of SRAM bits. Applying our method, we can reduce LSI costs and enhance the performance of the LSI by utilizing the reduced memories.

Keywords: line memory, image compression, mosquito noise, block noise

1. はじめに

近年、デジタルTVでは、画像の精細度を向上させる超解像処理や、画像のフレームレートを向上させるFRC (Frame Rate Conversion) 処理など、画像処理による高画質化が進んでいる。通常デジタル放送では、伝送帯域の有効利用を目的に画像圧縮が施されており、それに起因する圧縮ノイズにより高画質化処理の性能劣化が生じるため、高画質化の実現には圧縮ノイズの低減が必須である。

画像圧縮ノイズの推定や低減方法は、ビットストリーム参照型 [1], [2], [3], [4] と、ビットストリーム非参照型に分類することができる [5], [6], [7]。

ビットストリーム参照型について説明する。圧縮ノイズ

発生の原因は、画像の矩形ブロックごとに離散コサイン変換 (DCT) により周波数変換した画像データを量子化することによる周波数成分の欠落である。よって、画像圧縮ノイズの発生量は、画像圧縮を行った際の量子化ステップサイズと相関が高い。そのため、圧縮ストリームをデコードする際に、この量子化を行ったブロック境界位置と量子化ステップサイズを分析することにより、圧縮ノイズ低減のためのフィルタの適用位置と強度を調整でき、高精度なノイズ低減を行うことができる。

一方、ビットストリーム非参照型は、デコードの詳細情報が入手できないため、圧縮ストリームの解析が行えず、デコード後の画像情報のみから圧縮ノイズ低減を行う方式である。画像情報のみで圧縮ノイズを検出し低減させる必要が生じ、ストリーム参照型に比べて圧縮ノイズ低減の難易度は高くなるのが一般的である。

図1に我々が目標とするデジタルTVシステムのブロッ

¹ 株式会社日立製作所横浜研究所
Yokohama Research Laboratory, Hitachi, Ltd., Yokohama,
Kanagawa 244-0817, Japan

^{a)} yusuke.yatabe.wj@hitachi.com

ク図を示す。放送波を受信し画像ストリームをデコードする Decoder LSI と、Decoder LSI の出力画像に対して、圧縮ノイズ低減，超解像処理，コントラスト調整，カラーマネージメント，FRC などの高画質化処理を行う高画質化 LSI と，その出力画像をパネルに出力するためのパネルコントローラ LSI から構成される。本システムにおける高画質化 LSI は，Decoder LSI と HDMI I/F が出力する 1080/60p の映像をリアルタイムに処理することが求められる。

我々は，低コストで TV システムを構築するために，画像圧縮ストリームをデコードする機能は，すでに安価に入手可能な汎用 LSI を用い，高画質化を行う機能を新規追加するシステムをターゲットとした。この場合，圧縮ストリームをデコードする機能と，画像圧縮ノイズ低減を行う機能は分離され，画像圧縮ノイズ低減機能では，量子化ステップサイズなどの詳細なパラメータを取得できず，デコード画像のみから画像圧縮ノイズを検出し低減させるビットストリーム非参照型での機能開発が必要となる。また，汎用デコーダは，画像の拡大機能（スケーラ）を有し，様々な画像サイズで画像圧縮されたストリームを，つねに表示画像サイズに拡大して出力するものが一般的である。この場合，量子化を行ったブロックも拡大されてしまい，圧縮ノイズ低減を行うためには，広範囲な圧縮ノイズの検出が必要となる。

この広範囲の圧縮ノイズ検出をハードウェアで実現する際には，広範囲の画像を保持するためのメモリが必要となるが，メモリコストは LSI コストに大きな影響を与えてしまうため，コスト削減手法の開発が必須である。

本研究における我々の目標は，低コスト画像圧縮ノイズ低減機能を開発することであり，ビットストリーム非参照型での圧縮ノイズ低減を前提に，画像圧縮ノイズ低減処理に用いるメモリを共有することにより，低コストで圧縮ノイズを低減させる技術を提案する。

本論文では，圧縮ノイズ低減処理の低コスト化技術について述べる。2 章では，圧縮ノイズ低減アルゴリズムの概要と，広範囲に圧縮ノイズの検出を行う必要性について示す。3 章では，提案方式であるラインメモリ削減技術について説明する。4 章では，ラインメモリ削減の結果を示す。

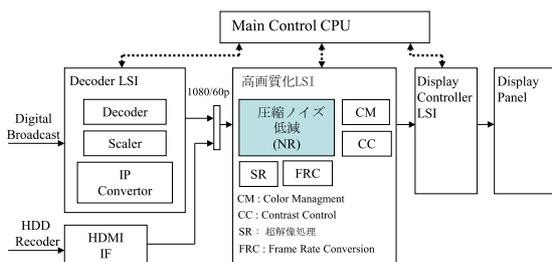


図 1 ターゲット TV システム
Fig. 1 The target TV system.

2. ノイズ低減アルゴリズム

本章では，従来手法である画像圧縮ノイズ低減アルゴリズムを説明し，圧縮ノイズ低減処理において画素を保持するためのメモリが多く必要とされる理由について示す。

2.1 ブロックノイズ低減処理

図 2 に本論文で低減対象とするブロックノイズの例を示す。ブロックノイズは，圧縮レートが低い場合や，多くの高コントラストな物体が画面内で激しく移動する場合に表示画像の固定位置にブロック状のノイズとして発生し，視覚的に非常に目立つ。さらに，超解像処理やエッジ強調処理において，ブロックノイズがエッジと誤判定されてしまいブロックノイズが強調されてしまう可能性もあり低減が必須である [8]。ブロックノイズが発生する原因は，画像圧縮時に個々のブロック独立に量子化を施すために，ブロックごとに圧縮精度が異なり，ブロック境界において，画素値の連続的なギャップが生じるためである。

本システムでは，Decoder LSI から量子化ブロック位置を取得できないため，画素値情報のみからブロック境界を検出する必要がある。従来研究では，画素値のギャップの検出と，その連続性を検出してブロックノイズ境界を検知する手法が提案されている [5]。本研究でも同様の方針でブロックノイズの検出を行う。

図 3 に，ブロックノイズを低減させるためのアルゴリズムのフローチャートを示す。ラスタスキャン順に入力される画素に対して，ブロックノイズの検出を行い，ブロックノイズと判定した部分のみに平滑化フィルタリングを施す。ブロックノイズ検出について述べる。ブロックノイズ検



図 2 削減対象とするブロックノイズの例
Fig. 2 An example of block noise.

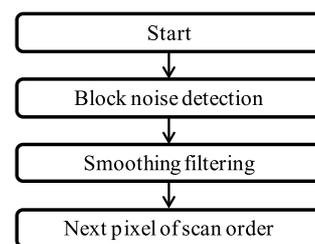


図 3 ブロックノイズ低減処理フロー
Fig. 3 A flow chart of block noise reduction.

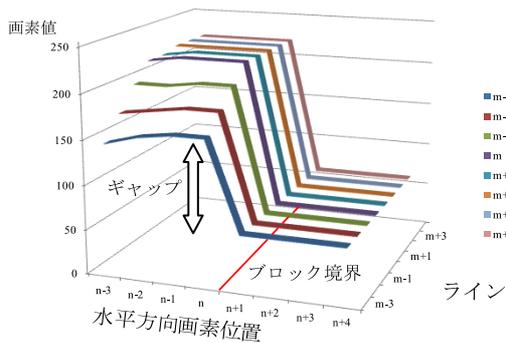


図 4 ブロックノイズの特徴
Fig. 4 Characteristics of block noise.



図 5 モスキートノイズの例
Fig. 5 An example of mosquito noise.

出は、画像情報のみからブロックノイズの特徴に基づいてブロック境界を検出する。ブロックノイズの特徴について実際の画素値の例を図 4 に示す。

本図の x 軸は画像の水平方向の画素位置であり、 n 番目と $n-1$ 番目の画素間にブロックノイズの境界が存在する例を示し、y 軸は、ブロック境界の方向となるラインを示し、z 軸はそれぞれの位置の画素値を示している。本図より、ブロック境界に沿って画素値の差が大きなギャップがブロック境界方向に連続する特性を有していることが確認できる。これにより、ブロックノイズ検出は、ブロック境界方向に複数のラインのギャップを算出し、すべてのギャップの連続性を確認する。上記条件を満たす場合、ブロックノイズと判定する。

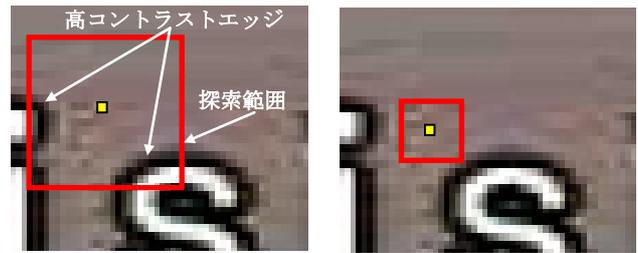
ブロックノイズと判定された境界に対しては、ローパスフィルタを施す。この際、高精度のフィルタリングを行うためには、多タップのフィルタリングが必要となる [1]。

上記説明したように、ビットストリーム非参照型において、高精度のブロックノイズの検出、低減を行うためには、ブロック境界周辺の広い領域の画素を用いてノイズ検出、ノイズ低減処理を行う必要がある。

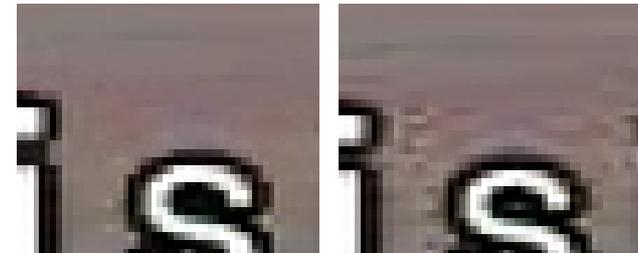
2.2 モスキートノイズ低減処理

モスキートノイズは、量子化による画像の空間周波数成分の欠落により発生する。画像圧縮では、高周波成分を多く含む高コントラストのエッジ周辺に発生することが多い。図 5 にモスキートノイズの典型的な例を示す。字幕の周辺にモスキートノイズが発生していることが確認できる。

モスキートノイズ低減方針を示す。画像圧縮時の量子化



(a) 広範囲エッジ検出 (b) 狭範囲エッジ検出



(c) (a)のノイズ低減結果 (d) (b)のノイズ低減結果

図 6 モスキートノイズ低減の例
Fig. 6 Examples of mosquito noise reduction.

幅はブロックごとに変更可能であり、比較的画素値が平坦な領域での画像劣化は視覚的に目立ちやすいため [4]、そのようなブロックでは量子化幅を小さな値とし、圧縮による画質劣化を抑えることが一般的である。そのような制御を行ったブロックでは、視覚的に目立つモスキートノイズが発生する可能性は低くなるため、ノイズ誤判定による画質劣化を防止する観点で、ノイズ低減処理を実施しないようにすることが望まれる。従来研究でも、ノイズが多く発生するエッジ周辺に着目してノイズ発生量を抑制する技術が報告されている [9]。よって我々は、高コントラストなエッジ周辺のみをモスキートノイズ低減の領域として判別し、その領域のみに対して、ノイズ低減を行うこととする。

図 6 にエッジ検出の探索範囲の違いによるノイズ低減効果の例について示す。図の (a) は、エッジ検出の探索範囲を量子化したブロックのサイズに設定した例で、高コントラストなエッジが探索範囲に含まれるため、モスキートノイズの発生を予測することができ、ノイズ低減を良好に行うことができる (図の (c))。一方、図の (b) は、エッジ検出の探索範囲を、量子化したブロックのサイズより狭くした例であり、注目画素は、探索範囲に高コントラストなエッジを含まないため、強度の強いノイズ低減を行えず、ノイズ低減は不十分になってしまう (図の (d))。よって、モスキートノイズ低減のためには、広範囲でのエッジ検出処理を行うことが重要である。

エッジ検出範囲について述べる。デジタル TV 放送を対象とする場合、入力画像は、MPEG2 ビデオ規格によって符号化され、それらは 8×8 画素のブロック単位に DCT と量子化が行われ、そのブロック内にモスキートノイズが発生する。本システムでは、Decoder LSI からつねにプロ

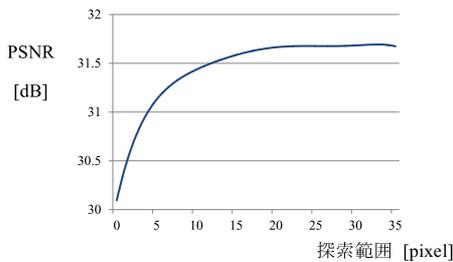


図 7 エッジ検出範囲とノイズ低減の関係

Fig. 7 The relation between search range and quality.

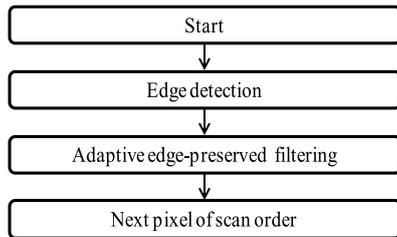


図 8 モスキートノイズ低減処理のフローチャート

Fig. 8 A flow chart of mosquito noise reduction.

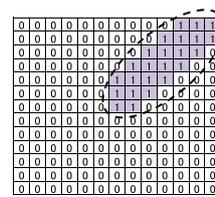
グレース形式の FHD サイズに拡大されて出力されることを想定している。放送では、画像サイズは 480p サイズ (854 × 480 画素) の放送も考えられ、それらを FHD に拡大した場合には、8 × 8 画素のブロック・サイズは、18 × 18 画素のブロックサイズに拡大されることになる。ピットストリーム非参照型では、ブロック境界位置が不明のため、上記範囲すべてを検索する場合、中心画素から縦横方向に 17 × 17 画素を検索する必要があり、合計で 35 × 35 画素の範囲において、エッジの存在を検出する必要がある。

エッジ検出範囲とノイズ低減の関係を調査した結果を、図 7 に示す。図は、480p の画像を FHD に拡大した場合のノイズ低減処理に対して、横軸にエッジ検出範囲、縦軸にノイズ低減後の PSNR を示す。PSNR の測定は、平坦部と急峻なエッジを多く含むチャート画像を用い、モスキートノイズ低減量が PSNR に反映されやすい環境で行った。図より、探索範囲を広くすることで高精度なノイズ低減が可能であることが、また、本拡大率の条件下では必要な探索範囲は 23 画素程度であることが確認できる。

図 8 に、モスキートノイズ低減処理のアルゴリズムフローチャートを示す。入力画素に対して、広範囲のエッジ検出処理を行い、周辺にエッジを含む場合にのみ、エッジ保存型のメディアンフィルタリングを施す。この際、広範囲の周辺のエッジ検出には、図 9 に示すエッジ検出を行った結果の 1bit のフラグのみを保持する。

2.3 ラインメモリ削減手法 (従来研究)

上記述べた画像圧縮ノイズ低減アルゴリズムについて従来研究を改めてまとめる。ブロックノイズ低減は、画素値の不連続性をもとにブロックノイズの境界を検出し、ロー



1: edge, 0: no edge

図 9 エッジ検出結果フラグのマップ

Fig. 9 Edge flag map.

パスフィルタを施す方式が研究されている [1], [2], [5]. モスキートノイズは、エッジ周辺に多く発生し [10], 各種平滑化フィルタを施す方式が研究されている [3], [4], [9].

次に、本ノイズ低減をハードウェアで実装する際のアーキテクチャについて従来研究を示す。画像の画素データを走査線 (ラスタスキャン) 順に伝送させるリアルタイム画像処理では、ラスタスキャン順に入力される画素を SRAM により構成されるラインメモリで複数ライン分保持し、そしてラインメモリから複数ラインの画素を同時に読み出すことにより、複数ラインの画素を用いた画像処理を実現している [11], [12]. そして、これらのブロックを複数ブロック接続する場合には、カスケード接続しパイプライン処理するのが一般的である [13], [14]. このアーキテクチャは、画像を扱う幅広い分野で使用されている。本構成におけるラインメモリ削減の技術については、画像処理ブロック内の画像処理アルゴリズムを最適化することにより処理ライン数を削減する手法が数多く報告されている。これは、ラインメモリは高回路コストであるため、画質を維持したうえでのラインメモリ数削減が求められるためである。具体的には、顔検出における画像入力方法を工夫することによりラインメモリ量を削減する手法 [15], 画像精鋭化フィルタやランダムノイズ低減のアルゴリズムを最適化することにより必要ライン数を削減する手法 [16], [17], イメージセンサの複数ある読み出しモードを考慮してラインメモリ本数を最適化した手法 [18] などが報告されている。また、画像信号処理以外のラインメモリの共有化手法としては、CPU のキャッシュメモリの共有化手法 [19] などが報告されている。

本論文では、圧縮ノイズ低減アルゴリズムを例に複数の画像信号処理ブロックをカスケード接続した際に、各処理ブロックのフィルタタップ数の違いに着目しラインメモリを削減する手法について報告する。

3. 低コスト圧縮ノイズ低減アーキテクチャ

本章では、従来方式と提案方式のそれぞれのアーキテクチャについて簡略化した例を用いて説明を行う。

3.1 従来アーキテクチャ

2.3 節で説明した画像圧縮ノイズ低減処理技術を、リアルタイム処理を可能とするハードウェアで実装する際の従

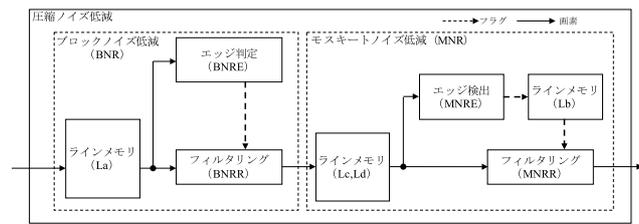
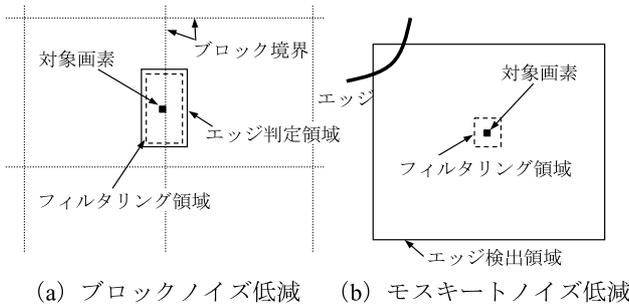


図 10 従来のノイズ低減パイプライン処理
Fig. 10 Conventional architecture of noise reduction.



(a) ブロックノイズ低減 (b) モスキートノイズ低減
図 11 ノイズ低減処理範囲
Fig. 11 The reference area for noise reduction.

来アーキテクチャについて説明する. 図 10 は, 従来方式であるノイズ低減処理のアーキテクチャを示している. このアーキテクチャは, 2.3 節で説明したようにラインメモリを用いてブロックノイズ低減処理と, モスキートノイズ低減処理をカスケード接続した構成である.

ラスタスキャン順に入力される画素は, ブロックノイズ低減処理のラインメモリ (La) で必要ライン数分保持され, 複数ライン分の画素データに対してブロックノイズのエッジ判定 (BNRE) を行い, ブロックノイズと判定されたエッジに対してのみ, ブロックノイズ低減用の平滑化フィルタリング (BNRR) を実施し, ラスタスキャン順に画素を出力する. 次に, その出力画素はモスキートノイズ低減処理用のラインメモリ (Lc) に入力され, 周辺のエッジ検出 (MNRE) が実施され, 検出結果のフラグがエッジ検出結果保持用ラインメモリ (Lb) に保持される. そして, モスキートノイズ低減処理の平滑化フィルタリング (MNRR) は, 画素ごとに周辺のエッジ情報をラインメモリ Lb から読み出し, 周辺にエッジを含むと判定された場合のみ, エッジ保存型の平滑化フィルタリングを行う.

この構成における圧縮ノイズ低減処理に必要とされる画素範囲について説明する. 図 11 (a), (b) はそれぞれ, ブロックノイズ低減処理におけるブロックノイズ検出領域とフィルタリング領域, モスキートノイズ低減処理におけるエッジ検出領域とフィルタリング領域を示している. ブロックノイズ低減処理においては, エッジ検出とフィルタリングの範囲はほぼ同程度の領域であるが, モスキートノイズ低減処理では, フィルタリングの領域に比べてエッジ検出は広範囲に判定を行う必要がある (2 章参照). また, モスキートノイズのエッジ検出は, ブロックノイズ低減処

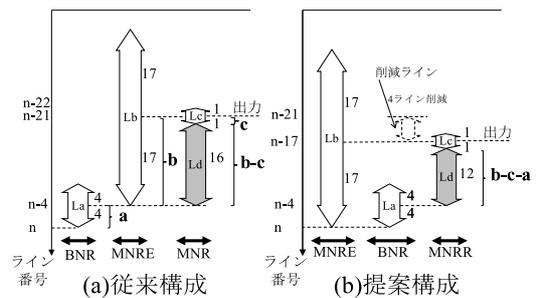


図 12 処理ライン遅延量
Fig. 12 The line delay of noise reduction.

表 1 各処理の処理ライン数
Table 1 A number of lines for each function.

ブロック	必要ライン数	処理ライン遅延
BNRE	9 (La)	4
BNRR	9 (La)	4
MNRE	35 (Lb)	17
MNRR	3 (Lc)	1

理の必要領域に比べて広い範囲が必要となる.

このような特徴を持つ両者のブロックをパイプライン接続した場合の問題点を示す. それは, モスキートノイズ低減処理における MNRR 処理は, 広範囲のエッジ検出処理が終了するまで処理を開始することができず, 入力画素を MNRR 処理開始ラインまで保持しておくための追加のラインメモリが必要となることである.

追加のラインメモリについて述べる. 図 12 は, 縦軸に入力される画像のライン数を示し, ラスタスキャン順に n ライン目が入力された場合に, 各ブロックがフィルタリングを行うライン数と, フィルタリング後の出力ラインを示している. 各ブロックの必要ライン数, 処理ライン遅延は, 2 章で説明した 480p の映像が FHD に拡大されて入力されるケースを考慮して表 1 に記載した数値と仮定した. なお本表においては, 説明の簡略化を目的に追加のラインメモリの説明に最低限必要なライン数のみを表記している. 表 1 において, 必要ライン数は, 各処理に必要な処理ライン数を示し, 括弧内に保持するラインメモリブロックを, 処理ライン遅延は, 各処理のライン遅延数を示している. 図 12 (a) において, n ライン目が入力されたときに BNRE と BNRR は, 9 ラインのブロック境界検出, フィルタリングを行い, フィルタリング結果を $n-4$ ライン目に出力する. そして, モスキートノイズ低減部に入力され, MNRE でエッジ検出を行い結果のフラグデータを Lb に保持する. そして, MNRR では, $n-21$ ライン目を中心として 35 ライン分のエッジ情報を Lb から読み出し, 領域内にエッジを含むかの判定を行い, エッジを含む場合には, $n-21$ ライン目に対して平滑化フィルタリングを行う. この際, モスキートノイズ低減処理は, $n-4$ ライン目が入力されたときに, 周辺のエッジ検出が終了するのは $n-21$ ラインとなり, その結果を取得しないと MNRR を実施するか否かが

判定できないため、ブロックノイズ低減処理終了後の画素を、MNRRを行う $n - 21$ ラインまで保持させておく追加のラインメモリが必要となる (図 12 のラインメモリ L_d)。本ラインメモリ L_d は、ノイズ低減の効果には寄与しないライン遅延のみを目的とするラインメモリであり、コストや消費電力の増加を引き起こすリアルタイム処理を前提としたハードウェアパイプライン処理の課題の 1 つである。

3.2 提案アーキテクチャ

我々は、前節で示した従来アーキテクチャに対して両者のノイズ低減に用いるラインメモリを共有することで、 L_d を低減させる手法を提案する。提案するアーキテクチャを図 13 に示す。図 10 に示す従来方式の構成と比較して、モスキートノイズ低減処理のエッジ検出処理の入力を、ブロックノイズ低減のためのラインメモリの出力とし、ブロックノイズ低減処理とモスキートノイズ低減処理におけるエッジ検出のためのラインメモリを、MNR と BNR の処理目的が共通であることに基づき共有する。しかし、本手法を用いることにより、モスキートノイズ低減処理におけるエッジ検出処理を、従来はブロックノイズ低減処理後の画像に対して実施していたが、処理前の画像で実施することとなる。これにより、モスキートノイズ低減処理のエッジ検出がブロックノイズを検出することによる性能低下が考えられるが、モスキートノイズ低減処理のエッジ検出は、高コントラストなエッジを検出対象としているため、それに対してギャップの小さなブロックノイズは検出対象外であり、また、誤検出が懸念される高コントラストなブロックノイズ (画素値のギャップが大きなブロックノイズ) が発生する領域は、高コントラストな物体を含む領域であり、そもそもモスキートノイズ低減対象領域である場合が多いため、性能への影響は少ないと考える。

提案方式によるラインメモリの削減効果を、図 12 (b) に示す。モスキートノイズ低減のエッジ検出の入力画素をブロックノイズ低減処理の入力画素を用いて、モスキートノイズ低減処理のエッジ検出を先行して行うことで、モスキートノイズ低減のためのフィルタリングを実施するラインを $n - 17$ ラインに早めることができ、画素を保持しておくラインメモリ L_d を、表 1 に示すラインメモリの条件においては、従来方式よりも 4 ライン削減することができ

る。一般的な TV システムでは、YUV 3 成分に対して本処理を行う必要があるため、合計 12 ラインの削減となり、削減の効果は大きいと考えられる。本技術を適用することで、ラインメモリの削減を可能とし、低コスト化、低電力化が可能となる。

我々は、他信号処理へ本技術を適用するために、ラインメモリ削減手法を一般化する。図 12 において、前段ブロックの処理遅延数 a 、後段処理の広範囲のエッジ検出処理遅延 b 、後段処理の狭範囲のフィルタリング処理遅延 c 、 b 検出のためのフィルタリングにおける処理ライン遅延を e (3 章の説明では簡単化のため e を 0 としている) とした場合、 L_d のラインメモリ数 L_{dn} は以下の式で表すことができる。

$$L_{dn} = b + e - c \tag{1}$$

そして、本提案手法であるラインメモリの共有化手法を用いることで、ラインメモリの削減量 LMR は、($L_{dn} \geq 0$) を条件として以下として表すことができる。

$$LMR = \min(a, L_{dn}) \tag{2}$$

この式より、ラインメモリの削減量 LMR を最大化するためには、BNR 処理における処理ライン遅延量 a を L_{dn} まで大きくすることが必要であることが分かる。

4. 評価

我々は、提案方式のハードウェア開発を行い、本ラインメモリ共有化技術の妥当性確認を行った。

4.1 ラインメモリ共有化技術の効果

開発したハードウェアの仕様と開発結果を表 2 に示す。本開発では、他高画質化回路との兼ね合いもあり、回路規模は 250 kG, SRAM は 1,000 kbit を目標とした。

本ハードウェアは、ラスタスキャンで入力される画像を、リアルタイムにノイズ低減させる。なお、ハードウェア規模は、2way-NAND を 1 Gate として換算した値である。モスキートノイズ低減部は、輝度のみでエッジ保存型のメディアンフィルタを施すことで、低回路規模を実現している。ブロックノイズは、YUV の 3 成分について、ブロック境界に対してローパスフィルタによる平滑化を行っている。

表 2 ハード仕様および開発結果

Table 2 Hardware specifications and development results.

項目	仕様および開発結果
画像サイズ	1920x1080 60fpsプログレッシブ
画像フォーマット	YUV 4:4:4 各10bit
入出力順番	ラスタスキャン順 (YUV同期入出力)
動作クロック	150MHz
LSIプロセス	90nm
ハードウェア規模	
MNR	40 kG
BNR	190 kG

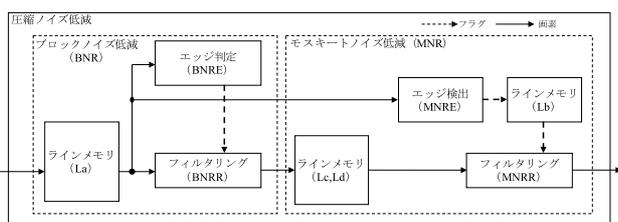


図 13 提案するパイプライン構成

Fig. 13 The proposed architecture of noise reduction.

表 3 開発したハードウェアのラインメモリ数

Table 3 The number of line memories of our hardware.

ブロック	必要ライン数	処理ライン遅延	成分数	bit数
BNRE	12 (La)	9 (a)	YUV 3成分	10bit
BNRR	12 (La)	9 (a)	YUV 3成分	10bit
MNRE	23 (Lb)	11 (b)	Yのみ1成分	1bit
MNRR	3 (Lc)	1 (c)	YUV 3成分	10bit

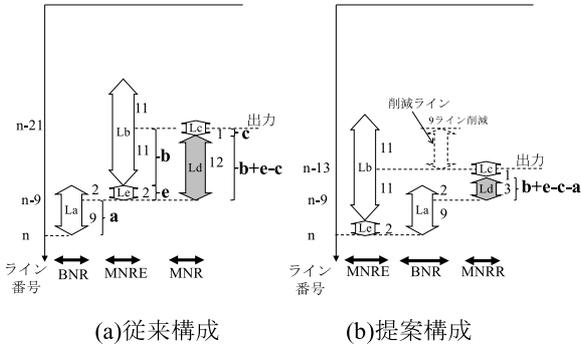


図 14 実開発における処理ライン遅延量
Fig. 14 The line delay of our hardware.

表 2 のスペックや目標回路規模, SRAM 量の条件で開発したハードウェアにおける SRAM の bit 量の削減結果について説明する. 表 3 に実際に開発したハードウェアにおける各処理のラインメモリ数を示す.

BNR の処理は, 表 1 では説明の簡単化のため省略していたエッジ検出のためのライン数も含めて, 必要ライン数を 12 ラインとしている. そして, BNR の処理ライン遅延は, 式 (2) よりラインメモリ削減量の最大化を図るため, 局所的なエッジ検出を複数ライン分行った後にギャップの連続性確認を行い, 境界にローパスフィルタを実施する方式として 9 ライン (処理遅延 $a = 9$) とした. MNRE は, 5 タップで処理ライン遅延が 2 ライン (処理遅延 $e = 2$) であるエッジ検出フィルタにより図 9 に示す 1 bit のエッジ検出結果フラグを作成し, そのフラグに対して図 7 で示した性能維持に必要な 23 ラインでのエッジ判定を行い (処理遅延 $b = 11$), MNRR 処理として 3 タップのフィルタ (処理遅延 $c = 1$) を用いた.

図 14 に本条件における各ブロックのライン遅延量と, ラインメモリの削減量を示す. なお図の L_e は, MNRE におけるエッジ検出のためのライン遅延量を示している. 図に示す従来構成は, 表 3 に示すライン数において図 10 に示すラインメモリの共有を行わない方式, 提案方式は図 13 に示すラインメモリの共有を行う方式である. 図より, ラインメモリの共有により, 提案構成は従来構成に比べて L_d を 9 ライン削減可能であることが確認できる.

次に, 両方式の SRAM の合計 Bit 量を図 15 に示す. Bit 量は, 表 3 に示すライン数と YUV 成分数と 1 画素あたりの bit 数と横ライン数の乗算により算出した. 上記提案方式は, 従来方式に比較して L_d を 9 ライン削減可能であり, YUV 3 成分, 1 ライン 1920 画素, 1 画素 10 bit で

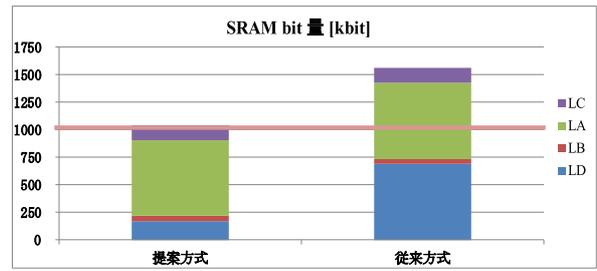


図 15 SRAM Bit 量

Fig. 15 The number of SRAM bits of our hardware.

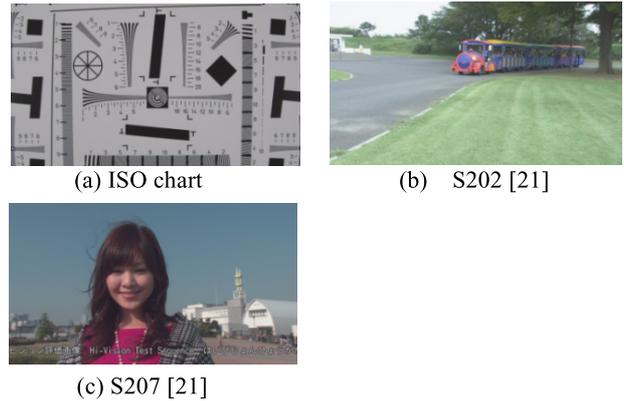


図 16 評価画像

Fig. 16 Original images.

表 4 ノイズ低減による PSNR の比較

Table 4 Comparison of PSNR.

Image	Chart	S202	S207	
Decoded Image	31.00 dB	32.53 dB	34.55 dB	
Proposed Method	31.62 dB	32.59 dB	34.61 dB	
Pipelined Method	31.62 dB	32.60 dB	34.63 dB	
Reference Methods	Low Pass Filter	21.58 dB	31.47 dB	34.15 dB
	Median Filter	22.84 dB	31.97 dB	32.00 dB

合計約 500 kbit を削減できる. この削減量は, 提案方式の全 SRAM 量の約 50% に相当する. また, 回路規模に換算すると約 500 kG (1 bit を 1 Gate 換算), 消費電力では約 30 mW の削減となる. これにより, たとえば図 1 に示す高画質化 LSI においては約 5% の SRAM 量の削減となり, それによるチップ面積削減による LSI 単価低減を可能とする. また, SRAM の削減分を他信号処理に割り振ることによる高機能化や, 新規ブロックの追加の組み込みによる多機能化を実現することも可能となる.

4.2 画質確認

本ラインメモリの共有化技術を用いることによる画質への影響を確認するために, 実際にノイズ低減処理適用後の画質の比較を行った. 図 16 に示す 3 種類の評価画像それぞれに, MPEG2 TM5 [20] を用いて QP45 固定により画像圧縮を施した画像に対して, 画像圧縮ノイズ低減処理を行った場合の PSNR と主観評価を行った. 表 4

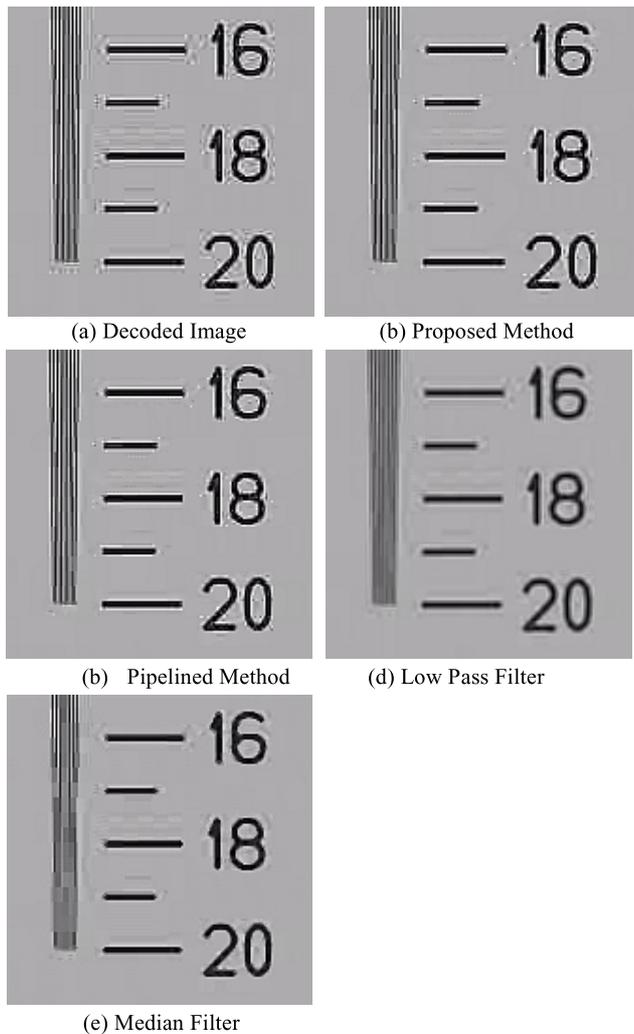


図 17 Chart 画像の再生画像

Fig. 17 Simulation results of the ISO chart.

は、3 種類の評価画像に対して、ノイズ低減処理を実施する前のノイズ発生画像 (Decoded Image)、表 3 に示すライン数の条件で図 13 に示すラインメモリの共有を行う提案方式 (Proposed Method)、図 10 に示すラインメモリの共有を実施しない従来方式 (Pipelined Method)、3 タップのローパスフィルタを画像全面に施した方式 (Low Pass Filter)、メディアンフィルタを画像全面に施した方式 (Median Filter) における画像圧縮前の原画像と、各処理後画像の PSNR を示している。

表 4 より、Decoded Image と Proposed Method を比較することで、デコード画像に対して圧縮ノイズ低減を行う効果が確認でき、Proposed Method と Pipelined Method を比較することで、ラインメモリの共有による性能劣化は微小であることが確認できた。また、Proposed Method を Low Pass Filter, Median Filter と比較することで、ノイズを検出して除去する方式の優位性も確認できた。

次に、再生画像の比較を行う。図 17 と図 18 は各方式の再生画像を示しており、図 17 は図 16 (a) に示す画像の中央部を、図 18 は図 16 (c) に示す画像の首元部分を拡大

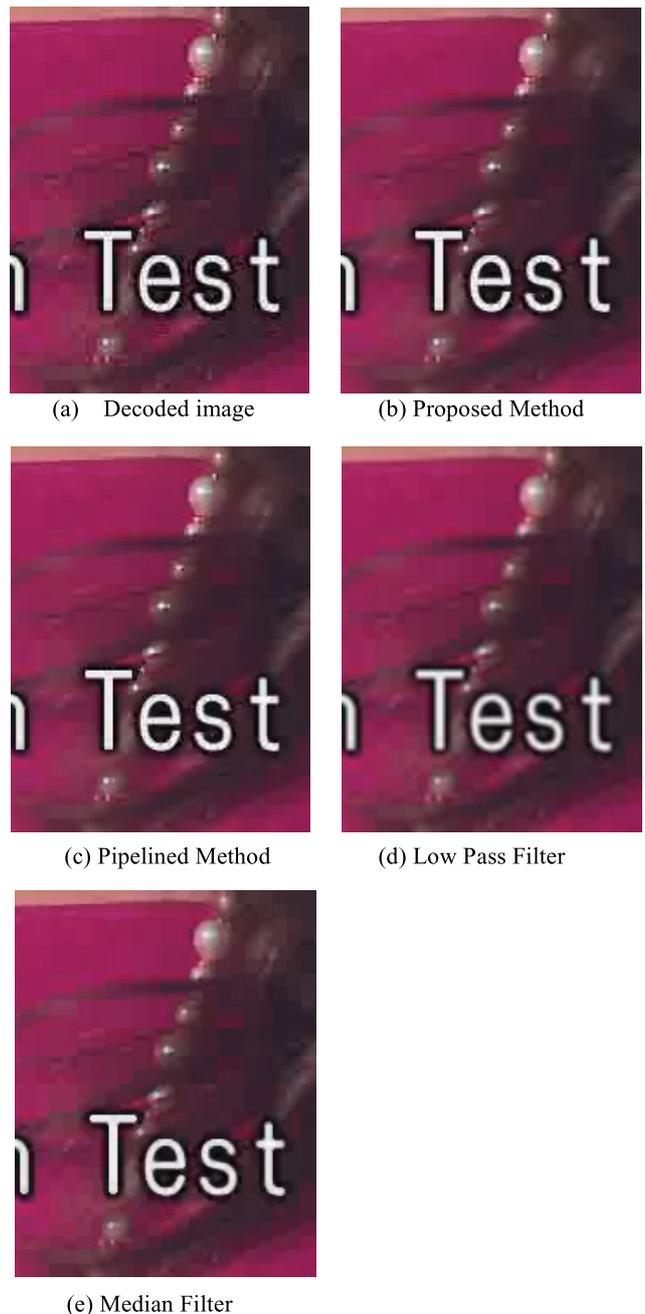


図 18 S207 の再生画像

Fig. 18 Simulation results of S207.

した画像である。それぞれの図において Decoded Image と Proposed Method を比較することで図 17 ではモスキートノイズ低減の効果が、図 18 ではブロックノイズ低減の効果が確認できる。また、Proposed Method と Pipelined Method を比較することで、ラインメモリを共有することでの画質への影響は微小であることが確認できる。さらに、Proposed Method を Low Pass Filter や Median Filter と比較することで、ノイズを検出して除去する方式の優位性も確認できた。

4.3 考察

一般的な回路開発においては、ラインメモリ量が目標を超える場合には、そのブロックの処理性能を犠牲として必要ライン数を減らすこととなる。しかし、本研究では、性能維持に必要なラインメモリを削減することなしに、ラインメモリを削減できる技術を提案した。本技術を一般化した式 (2) より、このラインメモリ削減技術は、後段ブロックの処理ライン数を縮小させずに L_d の削減数を最大化するには、前段ブロックの処理遅延 a を可能な限り L_{dn} とすることが有効であることが分かる。表 3 に示す BNR ブロックにおいては、複数ラインにわたる局所的なエッジの検出が終了した後にギャップの連続性の判定を行うことで処理遅延を大きくし、効果の最大化を図った。このように、複数処理のパイプライン接続時には、式 (2) をもとに、ラインメモリの共有化を図り、複数ブロックでの性能最適化を図ることが有効であると考えられる。

5. 結論

我々は、画像圧縮により発生するブロックノイズやモスキートノイズに対して、両者のノイズ低減処理に用いるラインメモリを共有することにより、画質に影響を与えずに SRAM を削減する技術を開発した。

本技術は、近年拡大化が進んでいる 4k2k, 8k4k などの次世代 TV への現行放送の拡大表示、量子化を行うブロックサイズが従来規格より大きくなる H.265 などの次世代の画像圧縮規格での圧縮ノイズ低減に対して有効である。また、本ラインメモリの共有技術は、ラストスキャン入力をラインメモリにより保持し、そのブロックを複数カスケード接続する信号処理に適用可能であり、たとえば、センサからの読み出し画像の高画質化や、擬似 3D 変換処理などで用いられる広範囲に周辺の状態を解析する必要がある処理に適用することが可能であると考えられる。

参考文献

- [1] Peter, L. et al.: Adaptive Deblocking Filter, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.13, No.7, pp.614-619 (1991).
- [2] Lee, Y.L. and Park, H.W.: Loop filtering and post-filtering for low-bit-rates moving picture coding, *Proc. Image Processing, 1999, ICIP 99*, Vol.1, pp.94-98 (1999).
- [3] Abbas, H. and Karam, L.J.: Suppression of Mosquito Noise by Recursive Epsilon-Filters, *IEEE International Conference on Acoustics, Speech and Signal Processing 2007, ICASSP 2007*, pp.773-776 (2007).
- [4] Westen, S.J.P. et al.: Adaptive spatial noise shaping for DCT based image compression, *Proc. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996, ICASSP-96 Conference*, Vol.4, pp.2124-2127 (1996).
- [5] Wu, H.R. and Yuen, M.: A generalized Block-Edge Impairment Metric for Video Coding, *IEEE Signal Processing Letters*, Vol.4, pp.317-320 (1997).
- [6] Hamada, T.: Digital Video/Audio NR+RR Monitoring System based on Motion Compensated Inter-frame/intraframe Objective Parameters, *World Multi-conference on Systemics, Cybernetics and Informatics (SCI) 2001*, Orland, Vol.13, pp.296-305 (2001).
- [7] Wang, A.C.Z. et al.: Blind Measurement of Blocking Artifacts in Images, *International Conference on Image Processing*, Vancouver, Canada, Vol.3, pp.981-984 (2000).
- [8] 柴田剛志ほか：摂動事例によって構成される事例ベース超解像、電子情報通信学会論文誌 D, 情報・システム, Vol.J96-D, No.8, pp.1790-1800 (2013).
- [9] 新井悠祐, 甲藤二郎: JPEG 圧縮における画質改善方法の比較検討, 情報処理学会研究報告オーディオビジュアル複合情報処理 (AVM), Vol.22, pp.59-64 (2008).
- [10] 河村 圭, 山本勇樹, 渡辺 裕: 動画におけるベクター表現の符号化に関する検討, 情報処理学会研究報告オーディオビジュアル複合情報処理 (AVM), Vol.124, pp.81-86 (2005).
- [11] 荒川尚久ほか: 画像処理用ラインバッファの高位合成系向けライブラリ化設計, 電子情報通信学会技術研究報告 CPSY, コンピュータシステム, Vol.112, No.376, pp.129-134 (2013).
- [12] 伊佐周平ほか: Bilateral Filter のハードウェア化による高速化, 電気学会論文誌 D, 産業応用部門誌, *IEEJ transactions on industry applications*, Vol.133, No.2, pp.132-138 (2013).
- [13] Seidner, D. et al.: Improved low-cost FPGA image processor architecture with external line memory, *2013 IEEE International Conference on Industrial Technology (ICIT)*, pp.1128-1133 (2013).
- [14] Maatta, J.-M. et al.: Generic software framework for a line-buffer-based image processing pipeline, *IEEE Trans. Consumer Electronics*, Vol.57, Issue 3, pp.1442-1449 (2011).
- [15] Kim, J-S. et al.: A novel architecture for low bandwidth and high utilization in face detection with Haar-like features, *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp.1-4 (2011).
- [16] Chen, S-L. et al.: VLSI Implementation of a Low-Cost High-Quality Image Scaling Processor, *IEEE Trans. Circuits and Systems II: Express Briefs*, Vol.60, Issue 1, pp.31-35 (2013).
- [17] Lien, C-Y. et al.: An Efficient Denoising Architecture for Removal of Impulse Noise in Images, *IEEE Trans. Comput.*, Vol.62, Issue 4, pp.631-643 (2012).
- [18] 三沢岳志ほか: 30 fps VGA 相当動画読み出し対応 330 万画素 CCD イメージセンサとその応用, 映像情報メディア学会技術報告, Vol.26, No.26, pp.65-70 (2002).
- [19] 岡慶太郎ほか: データ値の局所性を利用したライン共有キャッシュの提案, 研究報告計算機アーキテクチャ (ARC), Vol.24, pp.1-9 (2011).
- [20] ISO/IEC 13818-5 Moving Picture Experts Group Test Model 5.
- [21] ITE/ARIB ハイビジョン・システム評価用標準動画第 2 版.



谷田部 祐介 (正会員)

1999年東京理科大学理工学部電気工学科卒業。2001年同大学大学院修士課程修了。同年(株)日立製作所に入社。画像圧縮伸張技術, 画像高質化技術の研究開発に従事。映像メディア学会会員。



小味 弘典

1969年生。1992年大阪府立大学工学部電気工学科卒業。1994年同大学大学院修士課程修了。同年(株)日立製作所に入社。画像圧縮伸張技術の研究開発に従事。1999~2000年南カリフォルニア大学客員研究員。映像情報

メディア学会会員。