

# 低コスト画像圧縮ノイズ低減技術

谷田部祐介<sup>†1</sup> 小味弘典<sup>†1</sup>

概要：デジタル TV などのコンシューマ製品向けにモスキートノイズやブロックノイズなどの画像圧縮で発生するノイズに対して、低コストで低減する技術を開発した。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事でメモリ量の削減を実現した。  
キーワード：画像圧縮 モスキートノイズ ブロックノイズ ラインメモリ

## Low Cost Noise Reduction Technology For Compressed Video

YUSUKE YATABE<sup>†1</sup> HIRONORI KOMI<sup>†1</sup>

Abstract: a low cost noise reduction technology for compressed video such as mosquito noise and block noise has been developed. The authors characterized the length of filter taps for block and mosquito noise and proposed the memory sharing architecture for both types of noise detection. The architecture contributes to the reduction of line memories required for buffering filtering data in a noise reduction pipeline.

Keywords: Image Compression, Mosquito Noise, Block Noise, Line Memory.

### 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毎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事により、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する必要が生じ、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事が求められる。

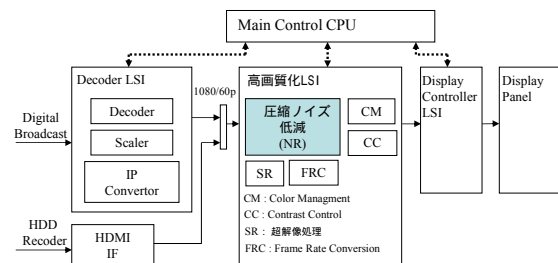


図 1 ターゲット TV システム

Figure 1 The Target TV System

我々は、低コストに 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す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、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、常に表示画像サイズに拡大して

<sup>†1</sup> (株)日立製作所 横浜研究所  
Yokohama Research Laboratory, Hitachi Ltd.

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為、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する技術を提案する。

本稿では、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。

## 2. ノイズ低減アルゴリズム

本章では、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、画質の劣化を生じさせる。また、超解像処理やエッジ強調処理において、エッジと誤判定されてしまう可能性もあり低減が必須である。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、ブロック毎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2 削減対象とするブロックノイズの例  
 Figure 2 An Example of Block Noise

図3に、ブロックノイズを低減す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。

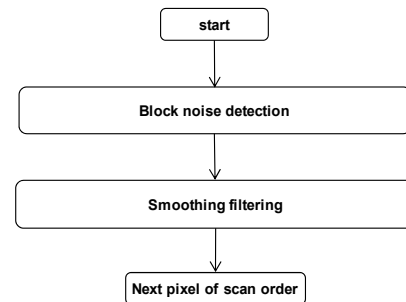


図3 ブロックノイズ低減処理フロー

Figure 3 A Flow Chart of Our Block Noise Reduction

ブロックノイズ検出について述べる。ブロックノイズ検出は、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。

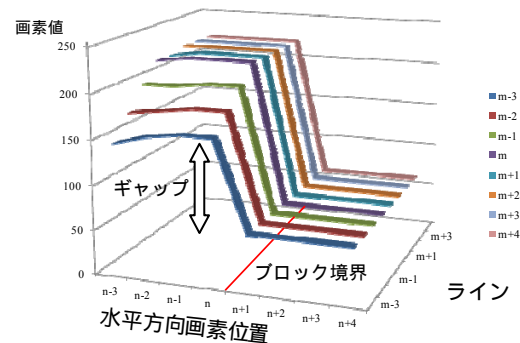


図4 ブロックノイズの特徴

Figure 4 Characteristics of Block Noise

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。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事が確認できる。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事が多い。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事が確認できる。



図 5 モスキートノイズの例

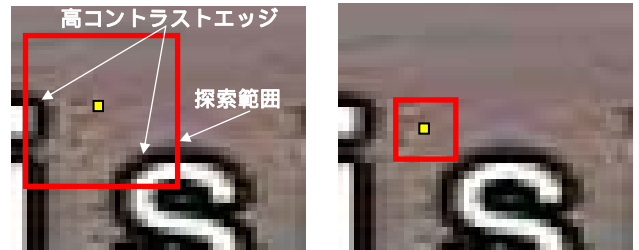
Figure 5 An Example of Mosquito Noise

モスキートノイズ低減方針を示す。画像圧縮時の量子化幅はブロック毎に変更可能であり、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易いため [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事が一般的である。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事が望まれる。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 [8]。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、ノイズ低減を行う事とする。

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事ができ、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が含まないため、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は、広範囲でのエッジ検出処理を行う事が重要である。

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 8x8 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 から常にプログレッシブ形式の 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 (854x480 画素) の放送も考えられ、それらを FHD に拡大した場合に

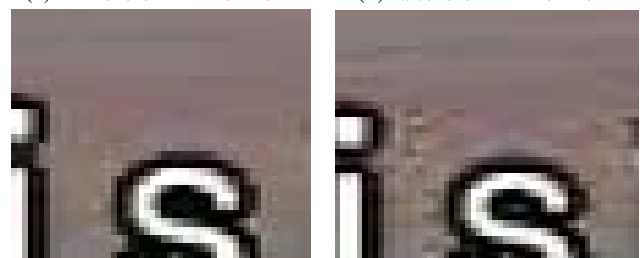
は、8x8 画素のブロック・サイズは、18x18 画素のブロックサイズに拡大される事になる。ビットストリーム非参照型では、ブロック境界位置が不明のため、上記範囲すべてを検索する場合、中心画素から縦横方向に 17x17 画素を検索する必要があり、合計で 35x35 画素の範囲において、エッジの存在を検出する必要がある。



■ モスキートノイズ低減対象の画素

(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

Figure 6 An Examples of Mosquito Noise Reduction

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事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 画素程度である事が確認できる。

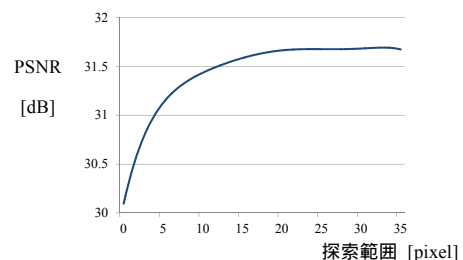


図 7 エッジ検出範囲とノイズ低減の関係

Figure 7 The Relation Between Search Range and Quality

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。

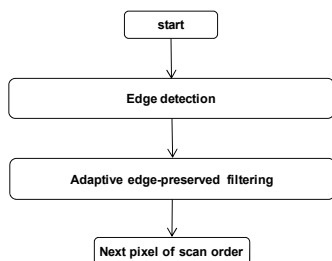


図 8 モスキートノイズ低減処理のフローチャート  
 Figure 8 A Flow Chart of Our Mosquito Noise Reduction



1: edge, 0: no edge

図 9 エッジ検出結果フラグのマッ  
 Figure 9 Edge Flag Map

### 2.3 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は、従来から多くの研究が行われている。ブロックノイズ低減は、画素値の不連続性を元にブロックノイズの境界を検出し、ロー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[9]、各種平滑化フィルタを施す方式が研究されている[3][4][8]。一方、ラインメモリ共有化の技術については、イメージセンサからの画素混合読み出しに用いる手法[10]、CPU のキャッシュ効率改善[11]の研究はなされているが、複数信号処理接続時のフィルタタップ数の違いによるラインメモリの低減技術については報告されていない。本論文では、従来の圧縮ノイズ低減アルゴリズムを元にリアルタイムハードウェア化した際のラインメモリ低減技術について述べる。

### 3. 低コスト圧縮ノイズ低減アーキテクチャ

本章では、我々が提案するラインメモリ削減技術を説明する。図 10 は、従来方式であるパイプライン化された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、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リアルタイムでの処理を可能とするため、ラインメモリを用いてパイプライン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、画素毎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。

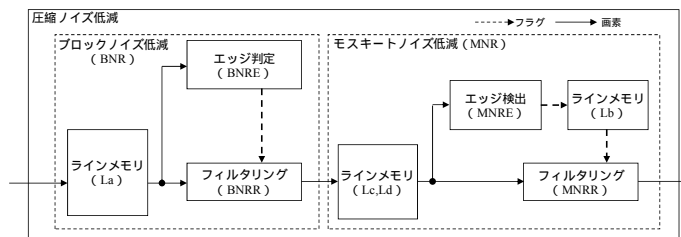
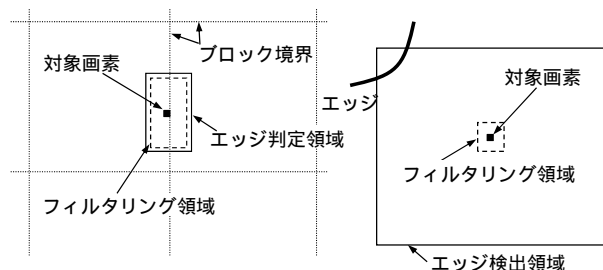


図 10 従来のノイズ低減パイプライン処理

Figure 10 Conventional Architecture of Noise Reduction

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理の必要領域に比べて広い範囲が必要となる。



(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

Figure 11 The Reference Area for Noise Reduction

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事が出来ず、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事である。

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

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せるケースを考慮して表 1 に記載した数値とする。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 (a) において、 $n$  ライン目が入力された時に 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  $L_b$  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  $L_b$  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時に、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一つである。

表 1 各処理の理想的な処理ライン数

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 ( $L_a$ )  | 4       |
| BNRR | 9 ( $L_a$ )  | 4       |
| MNRE | 35 ( $L_b$ ) | 17      |
| MNRR | 3 ( $L_c$ )  | 1       |

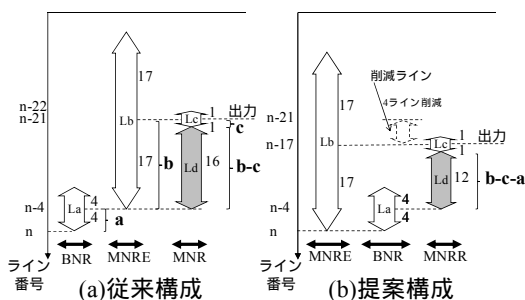


図 12 処理ライン遅延量

Figure 12 The Line Delay of Noise Reduction

そこで、我々は、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、 $L_d$  を低減する手法を提案する。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事に基づき共有する。しかし、本手法を用いる事により、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事となる。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事で、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  $n-17$  ラインに早める事ができ、画素を保持しておくラインメモリ  $L_d$  を、従来方式よりも 4 ライン削減する事ができる。実際の信号処理では、YUV の 3 成分に対して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。

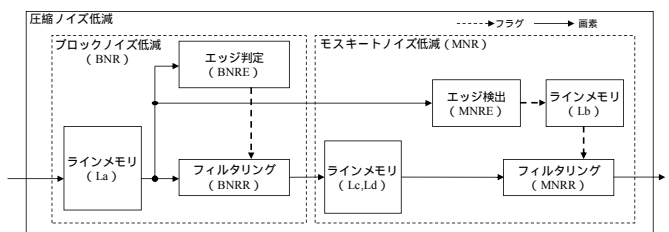


図 13 提案するパイプライン構成

Figure 13 The Proposed Architecture of Noise Reduction

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  $d$  (3章の説明では簡単化の為  $d=0$  としている) とした場合、 $L_d$  のラインメモリ数  $L_{dn}$  は以下の式で表わす事ができる、 $L_{dn} = b + d - c$

数式 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、 $(L_{dn} - 0)$

を条件として以下として表わす事ができる．

$$LMR = \min ( a , Ldn )$$

数式 2

#### 4. 評価

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

##### 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 250kG、SRAM は 1,000kbit を目標とした．

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                 |

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する．なお、ハードウェア規模は、2way-NAND を 1Gate として換算した値である．モスキートノイズ低減部は、輝度だけにエッジ保存型のメディアンフィルタを施す事で、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．

本ハードウェアにおける、ラインメモリである SRAM のビット量の削減結果について説明する．表 3 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．

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 |
|------|---------|---------|
| BNRE | 12 (La) | 9 (a)   |
| BNRR | 12 (La) | 9 (a)   |
| MNRE | 23 (Lb) | 11 (b)  |
| MNRR | 3 (Lc)  | 1 (c)   |

BNR の処理は、局所的なエッジ検出を複数ライン行なった後に、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施す方式とし、処理ライン遅延 9 ライン (a=9) の処理とした．MNRE は、5 タップのエッジ検出フィルタを用い処理遅延 2 ライン (d=2) とし、図 7 で示した性能維持に必要な 23 画素のエッジ検出を行い (処理遅延 b=11)、MNRR は、3 タップのフィルタ (処理遅延 c=1) を用いた．本条下での SRAM の Bit 量を図 14 に示す．従来方式は、ラインメモリの共有技術を用いない方式である．提案方式は、従来方式に比較して、数式 2 より、9 ラインのラインメモリの削減となる．これは、1 画素 10bit、1 ライン 1920 画素、YUV 3 成分では、約 500kbit の削減となり非常に大き

な削減効果であると考えられる．因みに、回路規模に換算すると約 500kG (1bit を 1Gate 換算) の削減となる．

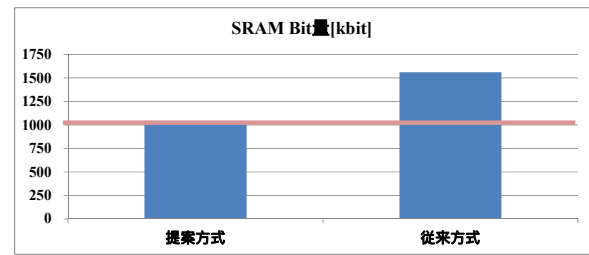


図 14 SRAM Bit 量

Figure 14 The Number of SRAM Bits

##### 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 15 に示す評価画像に対して、MPEG2 TM5[12]を用いてQP45 固定により画像圧縮を施した画像に対して、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 16、図 17 はそれぞれ、3 つ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)、本提案の低コストノイズ低減処理 (Proposed method)、ラインメモリの共有化処理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) の PSNR と、ノイズ低減後の再生画像を示している．

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ノイズ低減を行う効果が確認でき、また、Proposed Method と Pipelined Method を比較することで、ラインメモリの共有化による性能劣化は、微小である事が確認できた．また、Low Pass Filter、Median Filter 方式と比較して、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、再生画像の比較により、図 16 ではモスキートノイズ低減、図 17 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、ラインメモリの共有化技術を用いる事の画質への影響は微小である事が確認できる．

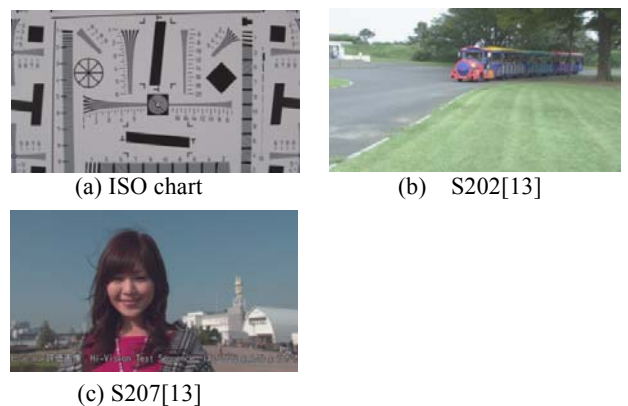


図 15 評価画像

Figure 15 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 |

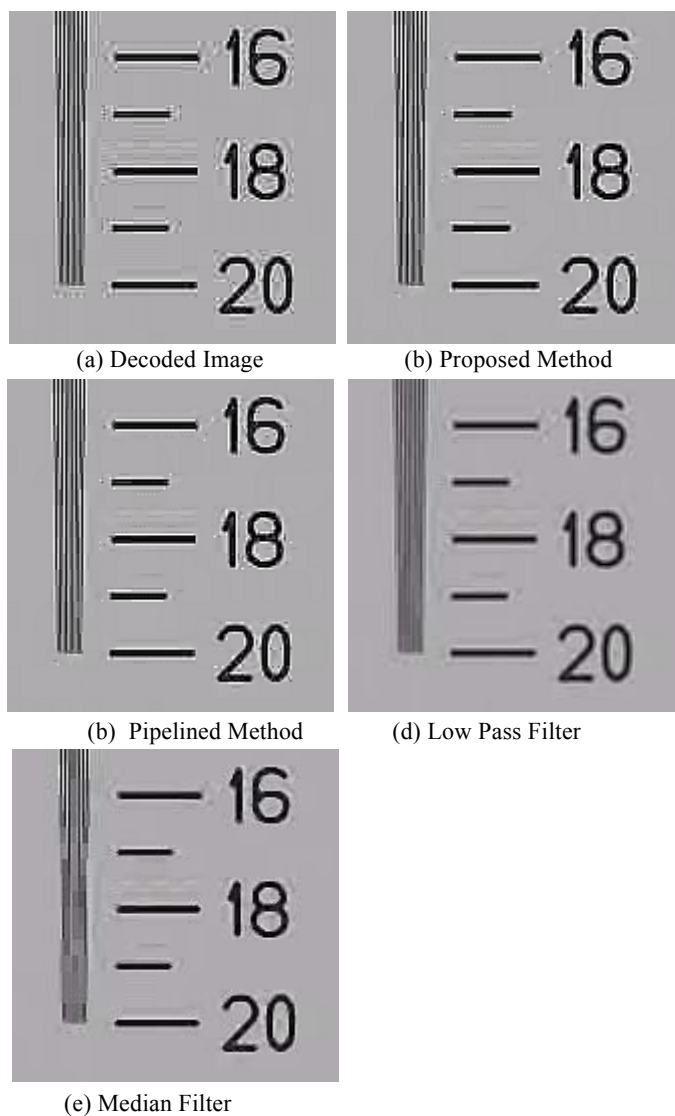


図 16 Chart 画像の再生画像

Figure 16 Simulation Result of the ISO Chart

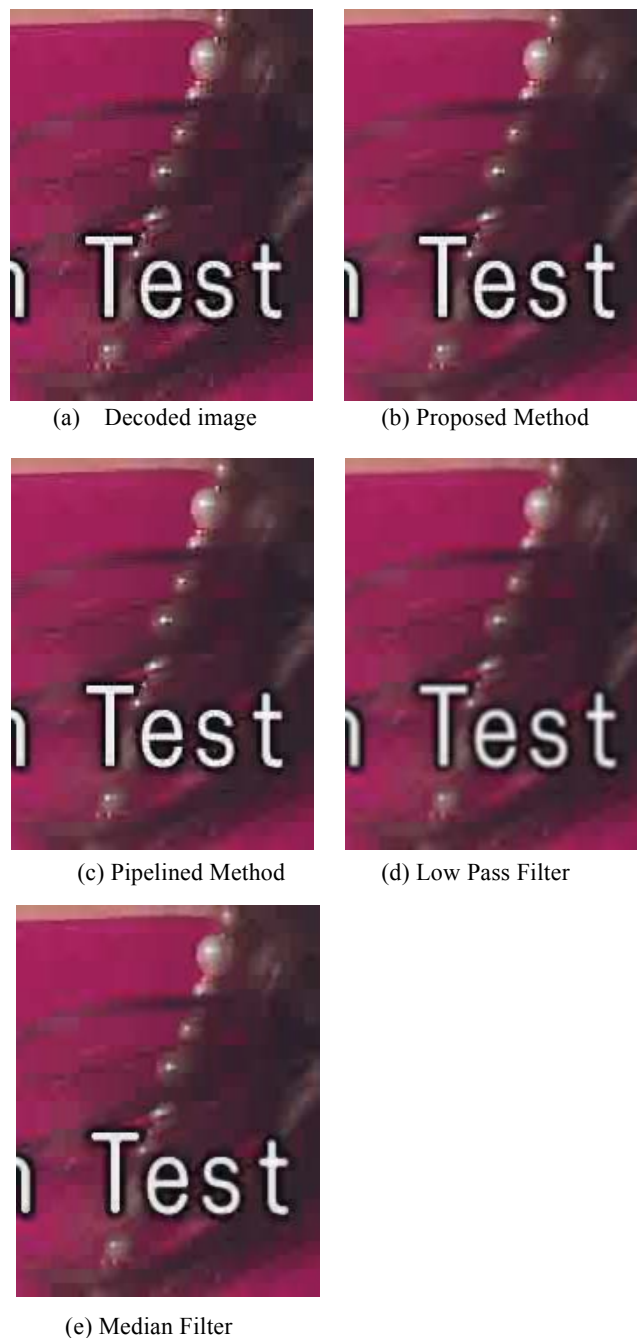


図 17 S207 の再生画像

Figure 17 Simulation Result of S207

#### 4.3 考察

従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事となる。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事無しに、ラインメモリを削減できる1つの技術を提案した。このラインメモリ削減技術は、後段ブロックの処理範囲を縮小させずに  $L_d$  の削減数を最大化するには、数式 2 より、前段ブロックの処理遅延  $a$  を可能な限り  $L_{dn}$  とする事が有効であることがわかる。本回路開発におけるブロックノイズ低減処理は、複数ラインに渡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事で処理遅延を大きくし、効果の最大化を図った。この様に、複数処理のパイプライン接続時には、数式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事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低コスト化を可能と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、センサ画像の高画質化や、擬似3D変換処理などで用いられる広範囲に周辺の状況を解析する必要がある他信号処理に適用する事も可能である。

## 参考文献

- 1) Peter, L. et al.: Adaptive Deblocking Filter, IEEE TRANSACTIONS ON 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, Image Processing, 1999. ICIP 99. Proceedings, Vol.1, pp94-98 (1999)
- 3) Abbas, H. and Karam L. J.: Suppression of Mosquito Noise by Recursive Epsilon-Filters, Acoustics, Speech and Signal Processing 2007. ICASSP 2007. IEEE International Conference on, pp.773-776 (2007)
- 4) Westen, S.J.P. et al.: Adaptive spatial noise shaping for DCT based image compression, Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on, 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 Interframe/intraframe Objective Parameters, World Multiconference on Systemics, Cybernetics and Informatics (SCI) 2001, Orland, Vol.13, pp.296-305 (2001)
- 7) Z.Wang A.C. et al.: Blind Measurement of Blocking Artifacts in Images, International Conference on Image Processing, Vancouver, Canada, Vol.3, pp981-984, (2000)
- 8) 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, pp59-64 (2008)
- 9) 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, pp81-86 (2005)
- 10) 三沢岳志, 他: 30fps VGA 相当動画像読み出し対応 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, Vol126(26), pp.65-70 (2002)
- 11) 岡慶太郎, 他: 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)

- 12) ISO/IEC 13818-5 Moving Picture Experts Group Test Model 5
- 13) ITE/ARIB ハイビジョン・システム評価用標準動画像 第2版