

E-13

遠隔協働支援システム t-Room における 映像エコー・キャンセリング法の実装と実験的評価

Implementation and Experimental Evaluation of Image Echo Cancellation in Remote Collaboration Support System t-Room

中村 譲†
Yuzuru Nakamura

小寺 晋平†
Shinpei Kotera

片桐 滋†
Shigeru Katagiri

大崎 美穂†
Miho Osaki

1. はじめに

遠隔協働支援システムの実現形態の一つとして「t-Room」が提案され、その性能向上を目指した研究開発が精力的に行われている¹⁾。t-Room は、遠隔地点間における視覚情報対称性の実現することによって、その利用者に同室感を提供することを目指している。しかし、その対称性の実現のために映像データの取得と再生に用いるカメラとディスプレイが対峙し、結果的に映像のハウリング、すなわちエコーを伴うと言う問題を持っている。筆者らは、この問題を解決することを目指し、カメラ映像とディスプレイ映像の差分を用いて映像エコーの除去を行う「映像エコー・キャンセリング法」の基本的動作を確認するためのプロトタイプを行ってきた²⁾。本稿は、そのプロトタイプ・システムを実際に t-Room 上に搭載し、その動作に関する評価結果を報告するものである。

2. t-Room と映像エコーの問題

2.1 t-Room

図 1 に t-Room の実現例を示す。t-Room は、大型のディスプレイとカメラ、マイク、スピーカのメディア機器から構成される壁面（モノリス）によって囲まれたマルチメディア空間である。それぞれのメディア機器は、例えばディスプレイ用のディスプレイサーバのような、対応するサーバコンピュータによって制御され、それらのサーバは LAN によって接続されている。



図 1 t-Room の構成事例。

接続される t-Room 同士は、モノリスに関する同一の空間的配置を採ることによって、メディアに関する対称性、即ち異なる t-Room 内にいる利用者同士が同一の視覚的あるいは聴覚的なメディアを知覚できる性質を実現することを

目指している。

2.2 フィードバックに起因する映像エコー

映像メディアの対称性を実現するためには、図 2 に図解するように、カメラとディスプレイを対峙する位置に置く必要がある。すると、明らかなように、カメラ映像を再生したディスプレイ映像は別のカメラ映像として取り込まれ、それが再び別のディスプレイ映像になり、それが再び初めのカメラ映像となるという映像のフィードバックが生じ、結果的に映像エコーを引き起こす。

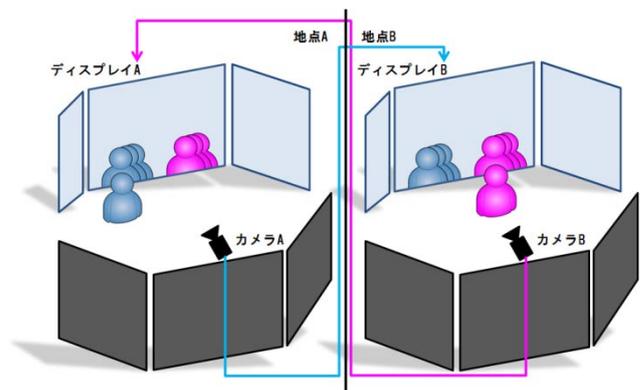


図 2 映像のハウリング。

これまでの t-Room では、この映像フィードバックはカメラに偏光フィルタを装着することによって抑制されてきた。この手法は非常に手軽で安価であるが、カメラに届く光が削減され、映像が暗くなる欠点をもつ。さらに、ディスプレイの種類は液晶型でなければならず、プロジェクタを用いる映像表示や他種のディスプレイの活用が困難となる。従って、この簡便ではあるが制約も大きな手法に代わる新しいエコー防止策の開発が望まれてきた。

3. カメラ映像とディスプレイ映像の差分を用いる映像エコー・キャンセリング法

3.1 方式の概要

映像差分を用いる映像エコー・キャンセリング法とは、映像エコーを生み出す元凶となるカメラ映像とディスプレイ映像とのフィードバック・ループを逆にとり、カメラ映像とディスプレイ映像との差分を求め、その差分映像から撮影すべき対象映像を抽出する手法である。要点は t-Room はディスプレイ上に表示されている映像を知っている点にある。取得したカメラ映像中からディスプレイ映像を差分

† 同志社大学大学院, Graduate School of Engineering, Doshisha University

することにより、カメラ映像中でディスプレイ前にある撮影対象部分のみを選択的に抽出する。図3に映像エコー・キャンセリング法の処理の概要を示す。

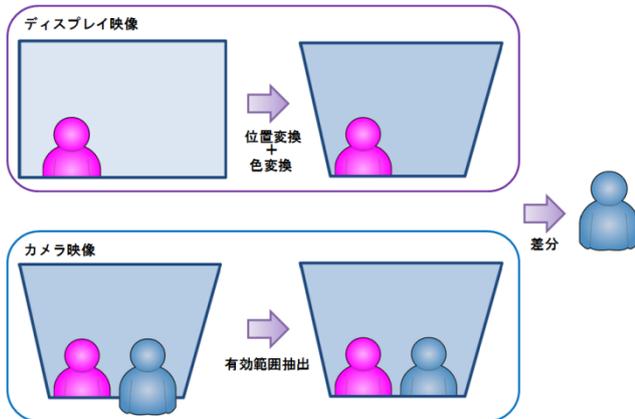


図3 映像エコー・キャンセリング法.

ここで注意することは、カメラ映像は一般にディスプレイの外側の被写体までも含んでいることや、カメラ映像中のディスプレイ領域は撮影角度の設定などの影響で歪を含むこと、さらにはディスプレイ上の色情報とカメラ映像上の色情報は必ずしも一致しないことである。従ってこの手法では、まずカメラ映像の中からディスプレイが映っている“有効範囲”を特定し(図中下側)、ディスプレイ上の表示映像をカメラ映像中のディスプレイ部分の領域に合わせるように変換し(図中上側)、さらにディスプレイ映像の色情報をカメラ映像の色情報に変換する(図中上側)必要がある。

図4に映像エコー・キャンセリング法全体の流れを図解する。本手法は、実際に映像間差分を計算して背景映像の削除を行うキャンセリング部とその処理に必要な事前情報を作成する前処理部から構成される。以下に各部の詳細を解説する。

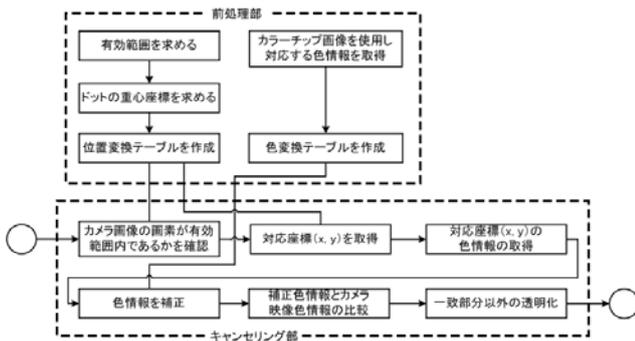


図4 映像エコー・キャンセリング法の全体の流れ.

3.2 前処理部

映像間の差分を求めるためには、両者の座標間の対応関係と色情報の対応関係を予め明確にしておく必要がある。前処理部ではこの対応付けに必要な位置変換と色変換に関する情報をそれぞれ位置変換テーブルと色変換テーブルとして作成する。

位置変換と色変換の情報は、予め準備された試験用画像をディスプレイ表示し(ディスプレイ映像の作成)、それをカメラで撮影して得られたカメラ映像と比較することで求め

る。試験用画像の例を以下の図5に示す。



図5 使用する試験用画像.

左の画像はカメラ映像中にあるディスプレイ面、即ちキャンセリング処理を行うべきキャンセリング対象領域を求めるために用いる。中央の画像はディスプレイ映像とカメラ映像との座標間対応関係、即ち位置変換テーブルを求めるために用いる。右の画像は両映像間の色変換状況を把握し、色変換テーブルを作成するために用いる。以下にこれらの変換テーブルを作成する具体的な手順を詳述する。

3.2.1 位置変換テーブルの作成

ディスプレイ映像とカメラ映像との位置変換は、(1)式で与えられる射影変換を用いて記述することが出来る。

$$\begin{cases} x = \frac{a_1x + b_1y + c_1}{a_0x + b_0y + c_0} \\ y = \frac{a_2x + b_2y + c_2}{a_0x + b_0y + c_0} \end{cases} \quad (1)$$

ここで、 (x, y) と (X, Y) とは変換前後のディスプレイ映像とカメラ映像の対応点の座標であり、点間の対応関係は式中の9つの係数 $a_0, b_0, c_0, a_1, b_1, c_1, a_2, b_2, c_2$ の設定によって決定される。従って、前処理部では試験用ドット画像中のドットそれぞれが、ディスプレイ映像とカメラ映像間で正しく対応がとれるようにこれらの係数の決定を行う。4枚のドット画像を用いるこの係数推定は、ガウス・ジョルダン法を用いて行っている。係数を求めた後、前処理部では後の処理の高速化のためにキャンセリング対象領域内全てのカメラ映像の点における対応するディスプレイ映像の点の座標を使用し、位置変換テーブルとして保存する。

3.2.2 色変換テーブルの作成

ディスプレイ映像とカメラ映像の色変換に用いるカラー・チャート画像は、RGB値のそれぞれに[0, 63, 127, 191, 255]の5つの値を設定して作成する125色のフラグメントの集合体として準備する。それをディスプレイ表示し、そのディスプレイ映像を撮影して得られるカメラ映像中の各フラグメントの色情報と前出のオリジナルの色制御情報とを比較する。こうして得られた対応関係を基に、全ての色に対応する関係を、RGBそれぞれの軸における(2)式で与えられる線形補間式によって構成する。ここで、ディスプレイの色情報 y_1 に対応するカメラ映像の色情報を x_1 とし、ディスプレイ映像の色情報 y_2 に対応するカメラの色情報を x_2 とする。

$$y = \frac{y_2 - y_1}{x_2 - x_1}x + y_1 \quad (2)$$

まず、 x 軸座標をディスプレイ映像の色情報とし、 y 軸をカメラ映像の色情報と考える。そして、2点からその間の全ての点に対応するカメラ座標の色情報を求めるために(2)式のように線形補間を行う。前処理部では上記の処理を RGB

の3次元に対して補間を行い、得られた全色に関する変換関係を1つの変換テーブルとして保存する。

これらの変換テーブルは画素値に変換情報を格納し、画像ファイルとして保存する。作成された変換テーブルを図6に示す。mapU.pngとmapV.pngは位置変換を行う際に、それぞれ座標のX軸とY軸に対応する値を格納している。LUT.pngは色情報に関する変換関係を全ての画素値について格納している。



図6 作成された位置変換テーブルと色変換テーブル。

3.3 キャンセリング部

図4に図解するようにエコー・キャンセリングの手続きは、まず前処理部で作成した位置変換テーブルと色変換テーブルを読み込むことによって開始する。

次のステップは位置変換テーブルを用いてディスプレイ映像中の各点とカメラ映像中の各点を対応付ける。位置変換テーブルからの対応点の求め方はカメラ映像の座標に対し、位置変換テーブル(mapU.png, mapV.png)のRGB値から対応するディスプレイ映像の座標を得る。そこで今度は対応する点間の色変換情報を色変換テーブル(LUT.png)から求め、ディスプレイ映像の色をカメラ映像の色に合わせる。ここまでの手続きで図7に図解するように、撮影に伴って生じる位置と色に関する歪は基本的に補正される。

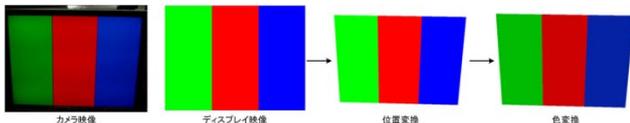


図7 ディスプレイ映像の補正結果。

最後に色変換後のディスプレイ映像とカメラ映像の対応点における差分を計算する。ある点における差の値が大きい場合、その点にはディスプレイに表示しなかった被写体、即ちディスプレイの前にある被写体が映されている可能性が大きい。従って、キャンセリング部はこの差の値に関する閾値を予め定めておき、点毎にディスプレイ前の撮影対象の有無を判断する。差の値が閾値以上の時、その点における透明度を最大値に設定して映っている映像をそのまま残し、それ以外の時には透明度を0として映っている映像を削除する。映像エコー・キャンセリング法によって、差分された映像の例を図8に図解する。

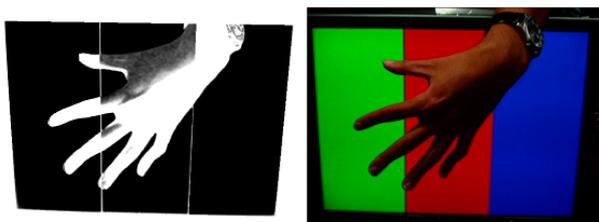


図8 映像差分の結果。

4. t-Roomにおける映像エコー・キャンセリング法の実装

4.1 概要

これまで、本キャンセリング法は、その基本的動作の検証を行うため、t-Roomの制御ソフトウェア(t-Room1.0やt-Room2.0などと呼ばれている)とは別のアプリケーションソフトウェアとして開発を進めてきた²⁾。本稿の研究では、t-Roomソフトウェア中のカメラサーバ機能を拡張し、そこに映像エコー・キャンセリング機能を実装する。

以降、カメラサーバの機能を拡張し、かつ映像エコー・キャンセリング法を実装した新しいカメラサーバをcamsvr3.0と呼ぶこととする。

4.2 開発環境

camsvr3.0の開発には以下の環境を用いた。

表1 開発環境

Operating System	Windows 7 Professional
Development Language	C++
Integrated Development Environment	Visual Studio 2005 Standard Edition
Plugins	Qt 4.5.2
	DirectX SDK November 2008
	Platform SDK for Windows server 2003 R2
t-Room Version	t-Room3.0
Capture Board	I-O DATA GV-VCP3R/PCI
	BUFFALO PC-MV31XR/PCI

QtはC++言語で記述されたGUIツールキットであり、単独のソースコードによりプラットフォームに依存しないアプリケーションの開発が可能である。t-RoomのGUI部はほぼQtで作成されており、GUI部以外にも多くのQtライブラリを使用している。また、t-Roomはカメラ入出力・音声入出力にDirectShow、OpenGL、DirectSoundを使用しており、DirectX SDKとPlatform SDKが必要である。

4.3 camsvr3.0の実装

提案したcamsvr3.0を図9のように実装した。それぞれの機能について詳細を以下に述べる。

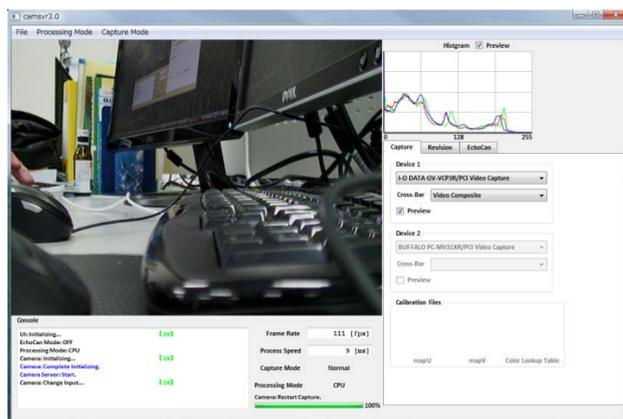


図9 新しく作成したカメラサーバ camsvr3.0。

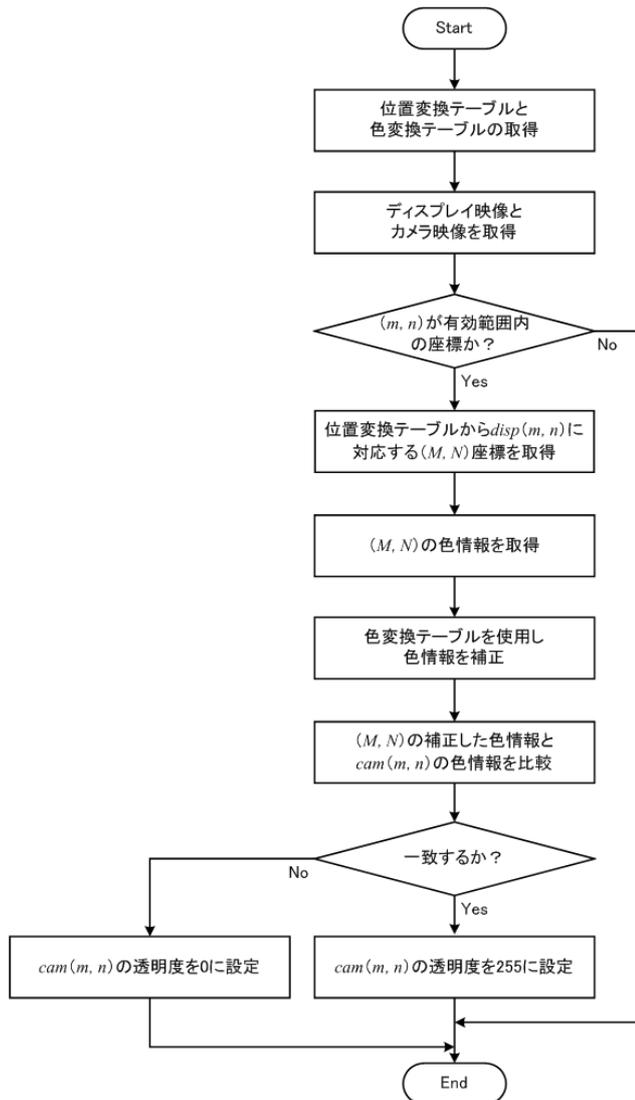


図 10 エコー・キャンセラの処理部の動作の流れ.

カメラサーバ本来の機能である通常のキャプチャモードに加えてエコー・キャンセラを実装し、GUIのメニューバーを押下することで通常のキャプチャとエコー・キャンセラを切替えることができる。また、エコー・キャンセラにはディスプレイ映像の位置変換や色変換の結果を確認する機能や、オーバーサンプリング機能が実装されており、差分の手法もいくつか既に実装されている。camsvr3.0ではユーザがこれらの機能を起動中にGUIで選択することができ、差分のためのパラメータも動的に変更することができる。

動作部では差分処理を行うためには変換テーブルがあればよいため、エコー・キャンセラ内のキャンセリング部のみを実装する。よって、前処理部のカメラ映像とディスプレイ映像の関係を求める機能は別のソフトウェアで実装している。キャンセリング部は主に初期化部と処理部に分けることができる。初期化部ではキャプチャデバイスの初期化や変換テーブルの読み込みなどを行い、処理部では実際に映像の差分処理を行う。エコー・キャンセラによる差分処理の一連の流れを図10に図解する。ここでは、 x, y 座標上のある一つの画素 (m, n) における処理について述べる。また、 $cam(m, n)$ はカメラ映像上の画素を表し、 $disp(m, n)$ はディ

スプレイ映像上の画素を表す。これらの処理を全ての画素に対して行う。

5. t-Room における映像エコー・キャンセリング法の実験的評価

5.1 t-Room に対する適用実験

作成した camsvr3.0 で実際に t-Room 上でエコー・キャンセリングを実施した。ここで、遠隔地点間の通信に用いられる t-Room においてはディスプレイに表示される映像はディスプレイサーバ上で作成(計算)されるものでない点に注意しなければならない。ディスプレイ映像は遠隔地から送られてくるものであり、ディスプレイ映像を取得するため、本実装ではキャプチャ・ボードを使用することにした。

t-Room における映像エコー・キャンセリング法の適用実験環境を図11に図解する。

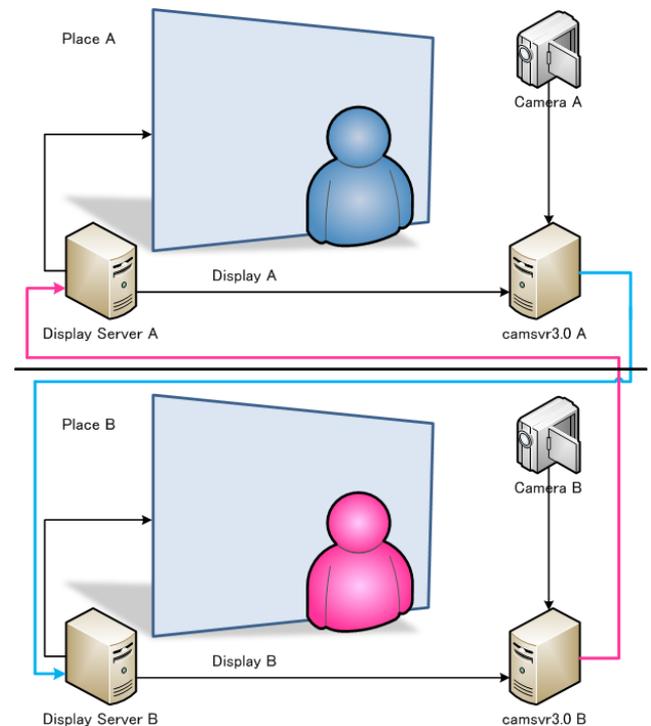


図 11 t-Room に対する適用実験の環境.

キャンセリング処理は camsvr3.0 が起動しているコンピュータ上で行うこととし、このコンピュータに2枚のキャプチャ・ボードを搭載する。まず、遠隔地 t-Room から送信されてくる映像をディスプレイに表示する役割を担うディスプレイサーバが起動しているコンピュータで受信し、そのグラフィック・ボードを経由して camsvr3.0 が起動しているコンピュータのキャプチャ・ボードに入力される。一方、ローカルな t-Room の中でカメラが撮影した映像は残りのもう1枚のキャプチャ・ボードに入力される。こうして、ディスプレイ映像とカメラ映像はいずれもキャプチャ・ボードを経由してキャンセリングを行うコンピュータ上に集められる(図中黒線)。キャンセリング処理が行われた出力映像は通常のカメラサーバと同じ要領で遠隔地のディスプレイサーバへと送信される。

5.2 実行結果と考察

エコー・キャンセリングを実行した結果を図 12 に示す。



図 12 実行結果(左上:カメラ映像, 右上:ディスプレイ映像, 右下:ディスプレイ映像の補正結果, 左下:キャンセリング結果)。

図 12 からわかるようにキャンセリングが不十分であることによって背景が残ってしまっている。t-Room 上では映像のフィードバックが生じることにより、差分しきれなかった領域は映像エコーとしてのハウリングの原因となる。よって、映像ハウリングを回避するため、エコー・キャンセリングの精度の向上が不可欠である。

カメラ映像中のディスプレイ映像とディスプレイの枠との境界部分が差分されずに残ってしまっている。これは射影変換の精度が不十分であるためにディスプレイ映像の位置変換がうまく行われていないために生じていると考えられる。従って射影変換の精度の向上が必要である。また、差分が十分に行われていないために図中の左側の人物が着ている白い服や右側の人物の顔の輪郭などが残ってしまっている。この原因としてディスプレイ映像の色変換の精度が不十分であるために発生していると考えられる。よって、色変換の精度を向上させる必要がある。このように本キャンセリング法は映像間の正確な点間対応や色情報の正確な対応を前提にしており、位置変換の精度が低ければ差分が上手く行われず、色変換の精度が低ければ抽出したい被写体を背景から分離できない。このため、位置変換と色変換の精度の向上のために一層の工夫が必要である。

また、映像エコー・キャンセリング法は上記のような複雑な処理を行っているため、処理速度の低下が懸念される。t-Room において協調作業を行う際、重要な要素の一つとして映像情報のリアルタイム性が挙げられるが、現在のエコー・キャンセラの処理速度は単純な RGB 値による差分法の場合で 20fps から 25fps 程度であり、フィルタ処理を追加した場合には 10fps 程度に低下する。映像のリアルタイム性を確保するためにはフレームレートが最低でも 30fps 以上は必要であり、これを下回るとリアルタイム性が失われて結果的に t-Room の同室感を阻害する恐れがある。また、複雑な処理による CPU の負荷の増大も無視することができない。現在は一つの PC が一つのサーバを使用する関係になっているが、今後は一つの PC に対して多数のサーバを搭載した

環境でシステムを起動することが望まれている。このため、CPU の負荷の増大が予想されるため、CPU の負荷軽減が必要である。

6. 位置変換精度の評価と改善策

6.1 射影変換に伴う誤差の分析

図 12 に見られるキャンセリングの不十分な結果の原因の一つとして、射影変換による位置変換の精度が不十分であることが考えられる。これを定量的に評価するため、35 個の交点からなるチェス盤画像を用いて前出の手続きで作成された位置変換テーブルによってディスプレイ映像から変換したチェス盤画像とカメラ映像中のチェス盤画像との交点間の誤差の分析を行った。

1 点に関して位置補正を施した際に対応する点の座標とどれくらい差が生じるかについて調べる。評価方法としては、チェス盤画像の左上から右下の交点へ走査し、35 点に関してそれぞれ誤差を求める。評価結果を図 13 に示す。

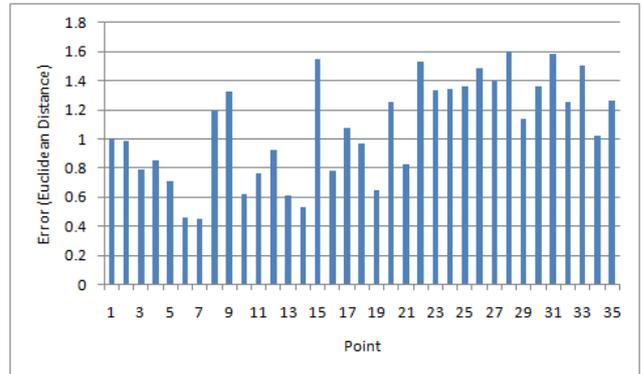


図 13 射影変換による各交点の誤差の分布。

図 13 から見て取れるように全ての点に対して誤差があり、その平均は 1.07 であった。また、誤差が横軸に対して右上がり増加していることから、誤差はカメラの角度やディスプレイとの距離が大きくなる程増加する傾向がある。この誤差がキャンセリングを実施する際に境界部分や細かい画素に対して影響を与えていると考えられる。本キャンセリング法は映像間の正確な点間対応を前提にしており、その一層の品質向上のためにはこの位置変換のずれをさらに低減する必要がある。

6.2 チェス盤画像を用いた対応点座標の取得

従来では射影変換の係数の求め方として、ドットが描かれた画像を表示し、ドットの重心を求めることによって対応点座標を求めていた。しかし、この手法では座標の取得精度がドットの重心を求める処理に左右される恐れがある。そこで、チェス盤画像の交点により対応座標を求める手法を提案する。チェス盤上の交点はドットの重心に比べて座標の決定を行いやすく、射影変換の精度の向上が期待できる。本手法ではチェス盤画像の交点を求める手法に画像処理ライブラリの OpenCV を用いた。従来法と提案した手法との比較を行った結果を図 14 に示す。

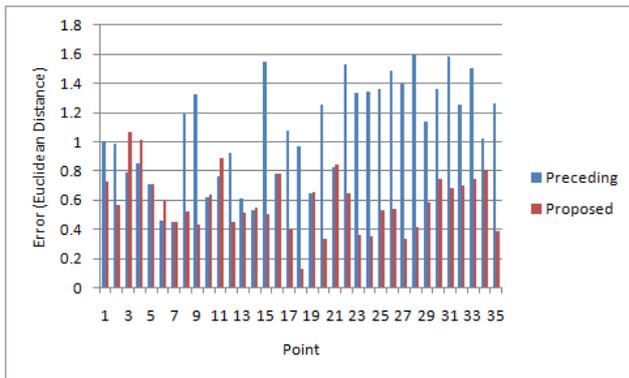


図 14 提案手法による各交点の誤差.

図 14 から見て取れるように提案手法では全ての点に対しての誤差が減少し、その平均は 0.59 となり、射影変換の精度が向上していることがわかる。特に、横軸 28 から 35 の交点の誤差の減少が顕著に見られ、提案手法はカメラとディスプレイとの距離や角度による精度の低下を回避できているといえる。

上記では射影変換による誤差を点間のユークリッド距離で評価したが、実際に画像は 1 ピクセル単位で扱われるためピクセル単位での誤差を調べる。今回は小数点以下を四捨五入し、点間の距離に画素の 8 近傍の移動経路によって距離を表現するチェス盤距離を用いて誤差に対する交点数の分布を求める。結果を図 15 に示す。

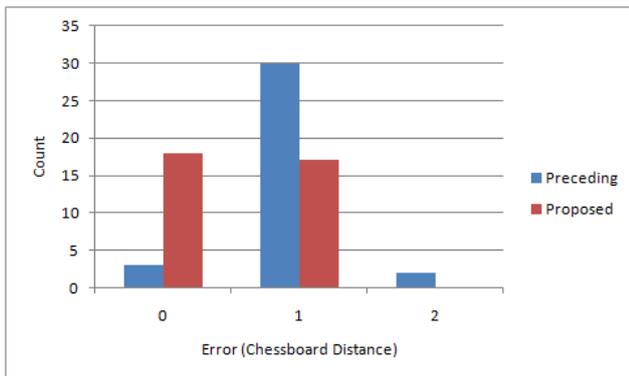


図 15 誤差に対する交点数.

図 15 より、従来法では 35 交点の 7 分の 6 に 1 ピクセルの誤差が発生している。これに対し、提案手法では交点の半分が誤差 0 となり、従来法に比べて誤差が減少していることが確認できる。

以上から、提案した手法は従来法に比べて射影変換の精度が高くなっていることが確認できた。これによって、映像エコー・キャンセリング法に対し、射影変換による位置変換の精度が向上できたといえる。

7. 色変換精度の評価と改善策

7.1 色変換に伴う誤差の分析

色変換テーブルを使用し、変換した色情報が正しい色情報に変換されているかを調べる。今回は色変換テーブルを作成されるときに使用したカラー・チャート 125 色に関して分析した。色ベクトルの要素値(R, G, B)に対する正しい色情報とキャプチャ・ボードから取得した色情報とのユークリッド距離を求め、誤差を求める。ユークリッド距離は RGB を色ベクトルと考え、以下の式で求める。ここで、 r, b, g は出力されたデータの色情報を表し、 r_t, g_t, b_t はカメラ画像の色情報を表す。分析した結果を図 16 に示す。

$$d_i = \sqrt{(r - r_t)^2 + (g - g_t)^2 + (b - b_t)^2} \quad (3)$$

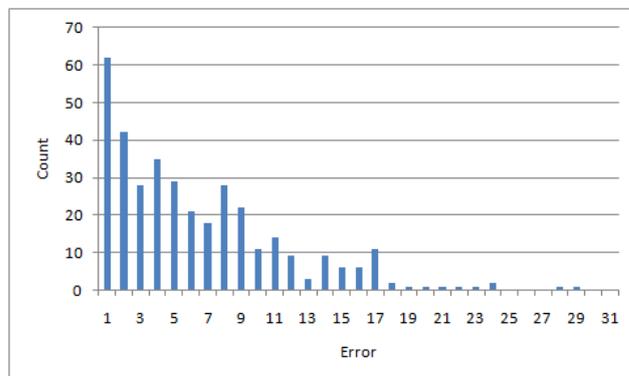


図 16 変換画像とカメラ画像の色情報の誤差.

図 16 から色変換テーブルを使用し、正しい色情報に変換されていないことがわかる。誤差の大きい部分では 30 ピクセルもあり、同じ色情報と判断する閾値を大きく外れる恐れがある。エコー・キャンセリングを行う際のディスプレイ映像はキャプチャ・ボードを使用し取得している。そこで、ディスプレイに表示させようとしている色情報とキャプチャ・ボードを介して取得した色情報が異なるのではないかと推測した。調べた結果、ディスプレイ上に表示させようとしている色情報とキャプチャ・ボードを介して取得したディスプレイ映像の色情報が異なっていることが確認できた(図 17)。

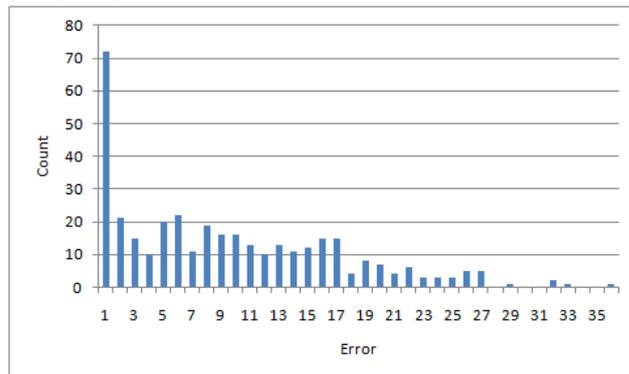


図 17 原画像とキャプチャ・ボードの色情報の差.

ここで、図 17 のような結果が出た原因としては、色変換テーブル作成する際にディスプレイ映像とカメラ映像の対応する色情報の間で補間を行っていないため色変換のずれが生じたことが考えられる。

そこで、キャンセリング部の中でディスプレイ映像を取得し、色変換テーブルに色情報を当てはめる部分に対してキャプチャ・ボードから取得した色情報を色変換テーブルを作成するために使用した色情報に補正する処理を導入することで精度を上げることができないかと推定する。

7.2 ニューラルネットワークを用いた色の補正

色情報を補正するために、今回は限られた色情報を用いて色変換器の学習を行い、その色変換器を用いて全ての色情報に対して補正を行うことを提案する。色変換器として3層ニューラルネットワークを用いた。学習基準として平均二乗誤差(MSE:Mean Squared Error)を使用し、学習手法は誤差逆伝搬法(EBP:Error Back Propagation Method)を用いている。

非線形処理を導入したフィードフォワード型の多層ニューラルネットワークは、中間層の数や内部パラメータを適切に設定することによって多様な連続関数を近似することができる。

変換行列を求めるために、訓練用データとして1000パターンの色情報を用意した(図18)。

16	16	16	0	0	0
40	16	13	27	0	0
69	13	12	55	0	0
99	11	11	83	0	0
130	8	6	111	0	0
160	6	4	139	0	0

図18 訓練用データの例。

この訓練用データは6次元あるデータの左の3次元と1次元を足した4次元を入力データとし、右の3次元を教師信号と設定し、学習させていく。入力データはキャプチャボードを介して取得した色情報、教師信号はディスプレイに表示させようとしている色情報とする。この訓練用データを使用して100000回学習させ、変換関数を求める。テストデータとして未知の8000パターンの色情報を用いて求めた変換関数が正しく動作できるかを調べる。評価方法としては入力データを変換関数に当てはめ、出力されたデータと教師信号のとのユークリッド距離を求め、比較する。

これを8000データに対しての平均を求め、変換関数に当てはめる前と後で比較した結果を以下に示す。このとき、初期学習係数は0.1、シグモイド関数の係数は0.5である。

表2 比較結果。

	元データ(キャプチャボードの色情報)	色補正後のデータ
平均ユークリッド距離	16.62	7.67
最大誤差	39	32.57

表2から平均ユークリッド距離が小さくなっていることが見て取れる。この結果から、キャプチャボードの色情報が正しい色情報に近付いていることがわかる。そして、先程と同様に元のデータ(キャプチャボードの色情報)と色変換後のデータを使用し比較した結果を図19と表3に示す。

表3 比較結果。

	元データ(キャプチャボードの色情報)	色補正後のデータ
平均ユークリッド距離	10.75	6.97
平均二乗誤差	5.52	3.4
標準偏差	4.68	3.24
最大誤差	28	20

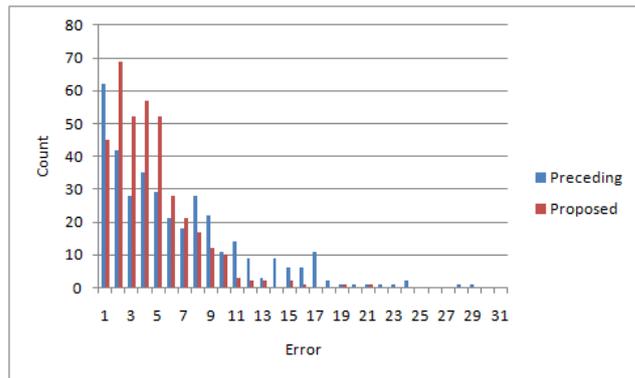


図19 色補正後の誤差。

表3より、同様に平均ユークリッド距離が小さくなっていることが見て取れ、補正後の色情報の方が誤差が減少していることがわかる。また、図19より提案手法により大きい誤差が減り、小さい誤差の方へ分布が移動していることがわかり、このことからt-Room上でも色補正によって誤差を減少させることができた。

以上から、ニューラルネットワークを用いて色情報を学習させることで正しい色情報に近似することができた。従って、提案した手法は従来法に比べて映像エコー・キャンセリング法の色変換における誤差の減少を実現できた。

8. GPUを用いた並列処理による高速化

8.1 GPUによる解決策の提案

映像エコー・キャンセリング法の複雑な処理によって処理速度の低下が発生し、リアルタイム性が失われて結果的にt-Roomの同室感を阻害する恐れがある。また、複雑な処理によるCPUの負荷の増大も無視することができない。そこで、GPU(Graphics Processing Unit)を用いた並列処理による処理速度の向上及びCPUの負荷軽減を提案する(図20)。

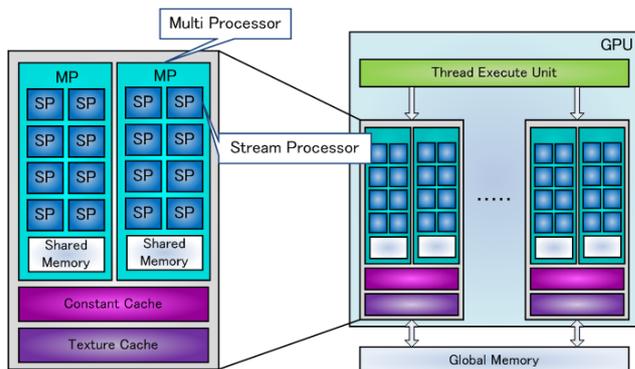


図20 GPUの内部構造。

GPUとは、CPUに代わってレンダリング等の画像処理を行うためのプロセッサである。近年、GPUの演算能力の高さに着目し、GPUでグラフィックス処理以外の汎用的な計算を行わせる「GPGPU(General Purpose Computation on GPU)」に関する研究が盛んになってきている。GPUの内部はマルチプロセッサと呼ばれる複数の回路ブロックにより構成されており、一つのマルチプロセッサ内には演算コアが8個搭載されている。

GPUは計算処理を分割し、演算コアに演算処理を割り振

ることによって並列処理を実現している。これを用いることで、並列処理による処理速度の向上が期待できる。また、処理を別ハードウェアである GPU に任せることにより、OS によるソフトウェアスレッドとは違って CPU の負荷を軽減することが期待できる。

また、GPU を用いるメリットの一つに導入コストの低さが挙げられる。GPU は専用のグラフィック・ボードと開発環境を導入するだけで使用可能であり、コストを抑えることができる。以上の理由により、本稿では映像エコー・キャンセリング法的高速化と CPU 負荷軽減の解決手法をして GPU による並列処理を提案する。

8.2 GPU を用いたエコー・キャンセラの実装と比較実験

映像エコー・キャンセリング法に関して、GPU を用いた並列処理を行うためにはエコー・キャンセラを GPU 処理用に実装する必要がある。よって、C++ で実装されたキャンセリング部のコードを GPU 処理用に書換えを行った。主な違いは変換テーブルのメモリ参照や差分処理の演算などを画素毎に並列に処理するようにした。開発言語は、NVIDIA 社が無償で提供している GPU 向けの C 言語統合開発環境 CUDA フレームワークを使用した^{3),4)}。

GPU 処理によるエコー・キャンセラの性能を評価するため、映像エコー・キャンセリング法で用いられる処理に対し、CPU と GPU との処理速度及び CPU の使用率について調査する。実験を行った処理は、(1)ディスプレイ映像の上下反転処理、(2)ディスプレイ映像の上下反転+位置変換、(3)ディスプレイ映像の上下反転+位置変換+色変換、(4)前後(3)の処理後、カメラ映像と RGB 値で差分し透明度の修正の4種類である。使用したハードウェア環境を以下に示す。

表4 ハードウェア環境。

Computer	Dell XPS 630
CPU	Intel Core 2 Quad Q9550 2.83GHz
RAM	4GB Dual Channel
Graphic Card	NVIDIA GeForce 9800 GT

8.3 処理速度の比較における結果と考察

処理速度における実験結果を図21に図解する。

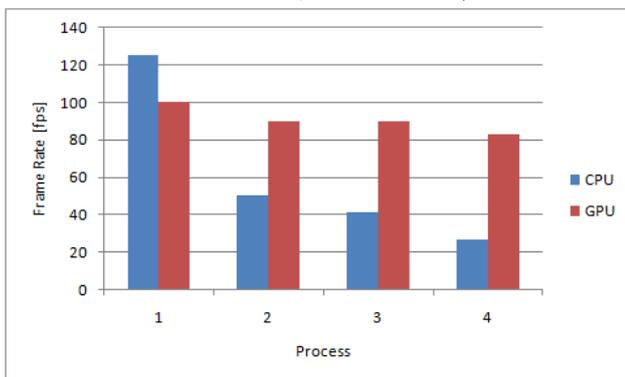


図21 CPU と GPU の処理速度の比較。

(1)の処理の場合、CPU 処理よりも GPU 処理の方が速度が遅いことが確認された。これは取得した映像をそのまま処理できる CPU に対し、GPU では CPU 側と GPU 側とのデー

タの転送がボトルネックになっていると考えられる。従って単純な処理の場合、GPU 処理を行うことにより、返って遅延が増すことがわかる。他の処理に対しては速度の向上が確認でき、最大で約3倍の高速化が実現された。

8.4 CPU 使用率の比較における結果と考察

CPU 使用率における実験結果を図22に図解する。

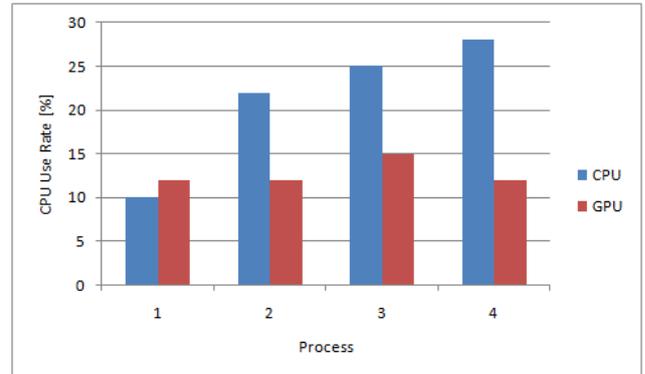


図22 CPU 使用率の比較。

(1)の場合では CPU 処理よりも GPU 処理の方が CPU 使用率が高いことが確認された。これは8.3と同様に CPU 側と GPU 側とのデータ転送が CPU の負荷に繋がっていると考えられる。他の処理に対しては CPU 使用率の減少が確認でき、最大で約2分1の負荷軽減が実現できた。

9. まとめ

映像エコー・キャンセリング法を実際に t-Room 上に実装し、その性能の分析を行った。実際に t-Room 上で映像エコー・キャンセリング法を稼働させた結果、生じた問題に対して本稿で紹介した改善策を適用することにより、それらの問題点の解決をすることができた。しかし、映像差分の精度は未だ不十分であり、撮影したい被写体までもが消されてしまう現象がある。この問題を回避するための一層の研究が必要である。

10. 謝辞

本研究の遂行にあたり、NTT コミュニケーション科学基礎研究所の平田圭二氏や原田康徳氏をはじめとする t-Room 研究グループの皆様には多大な御指導を頂いた。改めて御礼を申し上げる次第です。

参考文献

- 1) Mirai no Denwa t-Room. <http://www.mirainodenwa.com/>
- 2) 小寺晋平: 映像フィードバックに伴うエコーのキャンセリング法に関する実験的評価, 信学技報, Vol.109, No.306, PRMU2009-133, pp.291-296, 2009.
- 3) 青木尊之, 額田彰: ” はじめての CUDA プログラミング ”, 工学社
- 4) NVIDIA CUDA Information Site. <http://gpu.fixstars.com/>