

奥行き情報を利用した FTV 情報圧縮
FTV Data Compression Using Depth Map

石橋 崇司[†]
Takashi Ishibashi

圓道 知博[†]
Tomohiro Yendo

パナヒプル テヘラニ メヒルダド[†]
Mehrdad Panahpour Tehrani

藤井 俊彰[†]
Toshiaki Fujii

谷本 正幸[†]
Masayuki Tanimoto

1. はじめに

FTV(Free-viewpoint TV)[1]とは、多視点映像をもとに仮想的なカメラの映像の視聴を実現するものである。これが実現すれば、我々はテレビの中の空間を自由に移動できるようになる。このような自由視点の画像を生成するためには、計算機上に3次元空間の再構築が必要となる。

コンピュータグラフィックスの分野では、モデルベースの3次元空間の記述法が広く用いられている。この手法は、3次元空間中に物体の3次元情報を記述してモデル化を行う手法である。このような手法を MBR(Model Based Rendering)[2]と呼ぶ。しかし、この手法では詳細な3次元モデルを作成することに多大な労力が必要な上、実在する環境のような細かなモデルを構築することには限界がある。そのため、モデルベースにより作成された画像には人工的な不自然さが生じるという問題がある。

これに対して、実際に撮影された2次元の画像群から3次元空間を再構築する IBR(Image Based Rendering)[3, 4, 5]という手法がある。今回はこの一種である光線空間法[6]を基にしている。

カメラ配置の物理的な制約などによりカメラアレイにおいて光線空間は十分に密なものではなく、補間処理を必要とする。つまり、補間処理を行うという前提において同一時間上の多視点画像を光線空間情報と等価なものとして扱っている。その時の問題点として、多視点動画は符号量がとても多くなってしまふことと、本来重い処理が多い補間の際の視差推定の処理を高速かつ正確に行う必要があるという点である。[7]そこで本稿では、補間処理を補助する役割を持つ奥行き情報画像を用いて、光線空間としての扱いやすさと符号量の削減を同時に達成できるような多視点画像に代わる新しいデータの記述方法を提案する。

なお、自由視点画像生成の処理は非常に重いため、その処理を軽減させるための奥行き情報が予め生成されていることを前提とする。

2. 光線空間法

2.1 光線空間の概要

広い範囲の3次元空間を画像として表現するには、3次元空間情報を効率的に記述する必要がある。画像を光線の集合と考えれば、記述された3次元空間中の光線情報を読

み出すことで、画像を生成することができる。Plenoptic Function[8]では、3次元空間中の光線強度を、通過する点 (x, y, z) 、方向 (θ, ϕ) 、波長 λ 、時間 t の7次元のパラメータで表現している。空間中の光線を同定するパラメータは必要に応じて様々に選ぶことが可能であるが、ここでは、一直線に配置したカメラアレイに適したパラメータを考える。まず、3次元空間内に基準平面を設定する。光線の直進・比減衰を仮定し、時間と波長を省くことにより、 (x, y) と (θ, ϕ) の4つのパラメータ(光線パラメータ)により一意に表すことができる。すなわち、この空間中の光線は4次元空間 (x, y, θ, ϕ) の内の点に対応させることができ、この3次元空間内の光線情報を $f(x, y, \theta, \phi)$ と表すことができる。また、縦方向の視差を考えない直線カメラ配置における直交座標系では、光線情報を $f(x, y, \theta)$ と表すことができる。3次元空間情報をこのような関数 $f(p)$ (p は光線パラメータ)により表現する手法を3次元空間情報の光線群表現と呼び、光線パラメータにより張られる空間を光線パラメータ空間、または光線空間と呼ぶ。

2.1 光線空間の取得

光線空間の取得方法として一般的に2つの手法がある。最も一般的に研究されている手法が多視点カメラ方式である。[9]この方法では複数のカメラを用いて多視点画像を撮影し、光線空間を構築する。また、インテグラル・フォトグラフィ(Integral Photography; IP)と呼ばれる3次元画像の手法を応用した手法もある。これはレンズアレイをカメラの前端に置くことにより4次元の光線空間を2次元に変換し、それを2次元映像として取得する手法もある。[10]

これら2つの手法の本質的な原理は同じで、カメラを通過する光線を記録する装置として考え、カメラの1画素によって1本の光線情報を取得するのと何も変わらない。つまり、これらの手法では、光線空間を取得するために、取得する光線の本数と同じ数のカメラの画素が必要になる。従って、多くの光線情報を持つ密な光線空間を取得するためには、膨大な量のデータ量を取得しなければならない。

2.2 光線空間の特徴

直線に配置したカメラアレイの光線空間は、直線的な光線情報によって構成されている。光線空間は、直線的に並べたカメラがそれぞれ撮影した画像を平行に並べることで構築される。この光線空間のある高さで水平で切り出した断面画像をEPI(Epipolar Plane Image)と呼ぶ。図2.1にEPIの例を示す。EPIは直線的な光線情報によって構成されていることが分かる。

なぜ直線的な光線情報によって構成されるかを述べる。直線に配置したカメラアレイで光線情報を取得した場合、3次元空間内の光線情報 $f(X, Y, Z, \theta, \phi)$ は、3次元光線空間

[†]名古屋大学大学院 工学研究科

Graduate School of Engineering Nagoya University

[‡]東京工業大学大学院 理工学研究科

Graduate School of Engineering Tokyo Institute of Technology

$f(x, y, \theta)$ に変換することができ、その幾何学的性質より変換式は次式で与えられる。

$$x = X - Z \tan \theta \quad (2.1)$$

$u = \tan \theta$ としたとき、実空間中の一点 (X, Y, Z) を通る光線群が光線空間 $f(x, u)$ で描く軌跡は

$$X = x + uZ \quad (2.2)$$

と与えられる。これは実空間中の一点 (X, Y, Z) を通る光線群は、光線空間の $x-u$ 平面で直線を描くことを示している。

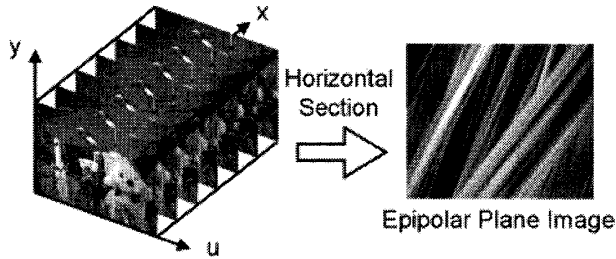


図 2.1 光線空間と EPI

3. 奥行き情報

奥行き情報を求めるために奥行きと画像間での視差の関係を幾何学的に導き出す。

図 3.1 に奥行きと視差の幾何関係を示す。簡単のため高さ方向を考慮せず 2次元平面で考える。高さ方向を考慮しても同様に処理することができる。左右のカメラを cameraL, cameraR とし、位置 α にある補間で作り出す画像の仮想カメラを cameraC とする。仮想カメラの x に入射する光の光線源 M の奥行きを D と仮定する。この時、この光線源が放つ光線は左右のカメラ、cameraL と cameraR のそれぞれ $u_L; u_R$ に入射する。同一の光線源が放つ光線を、異なるカメラが捕らえた点（この場合は u_L と u_R を対応点と呼ぶ。また、カメラ間の対応点の差、 $|u_L - u_R|$ を視差と呼び、本稿では d と記述する。

次に、奥行きと視差の関係を求める。図 3.1 において、カメラの焦点距離を f 、左右のカメラ間距離を I 、光線源 M と cameraL の X 座標の距離を m とした時、簡単な幾何関係から、図中の d_1 及び d_2 は以下のように求まる。

$$d_1 = \frac{f(I-m)}{D}, \quad d_2 = \frac{fm}{D} \quad (3.1)$$

視差 d は $d_1 + d_2$ で与えられるため、

$$d = \frac{fI}{D} \quad (3.2)$$

となる。つまり、視差 d はカメラ間隔に比例し、光線源の奥行きに反比例したものとなる。

本稿の奥行き情報は多視点のカメラ画像のみから、視差 d だけを参考にすべての光線源における相対的な奥行き D を求める。これにより、カメラ画像のみから図 3.2 のような奥行き情報を表すグレースケールの画像が生成できる。この奥行き情報画像は暗い所程奥行きが大きいことを表している。

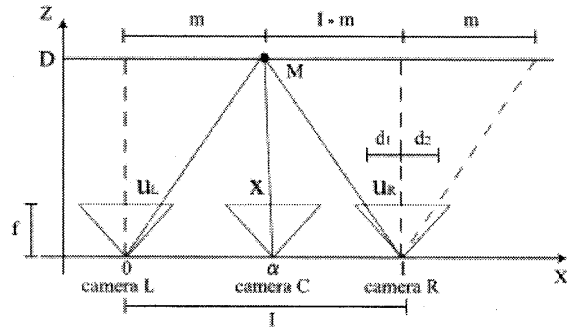


図 3.1 奥行きと視差の幾何関係

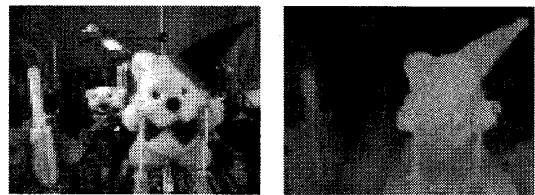


図 3.2 カメラ画像(左)とその奥行き情報画像(右)

4. 従来手法

4.1 MVC(Multi-view Video Coding)

MVC[11] は動画像符号化方式の H.264/AVC を、空間方向に拡張した符号化方式である。空間方向の符号化にも時間方向と同様に予測符号化が用いられている。予測された参照画像と取得画像の差分を離散コサイン変換することで、情報を削減し、エントロピー符号化により情報が圧縮される。予測画像は動き補償予測と画面内予測により生成される。予測に用いられるブロックサイズは可変であり、ブロックサイズが小さければ、よりきめ細かな予測を行うことができる。しかし、ブロックごとに動きベクトル情報を符号化しなければならないため、このための符号量が増加する。そこで、いくつかの動き補償ブロックサイズの中から、最適なものを切り替える仕組みが取り入れられ、全体の符号量を最小化するように最適化される。

MVC の構成は図 4.1 に示すように、前後の画面とは関係なく、その画面内(Intra)だけで独立して符号化することで得られる I ピクチャ、画面間の順方向予測により符号化が行われる P ピクチャ、前後の双方向からの予測によって得られる B ピクチャから構成される。

4.2 圧縮から自由視点生成までの全体像

MVC は奥行き情報を利用した圧縮符号化方式ではない。そのため、多視点動画とそれに対応する奥行き情報をそれぞれ別々に MVC を用いて圧縮符号化を行う。自由視点生成を行う際に圧縮符号化してある多視点動画と対応する奥行き情報をデータを復号化し、それらを自由視点画像生成の処理に用いている。

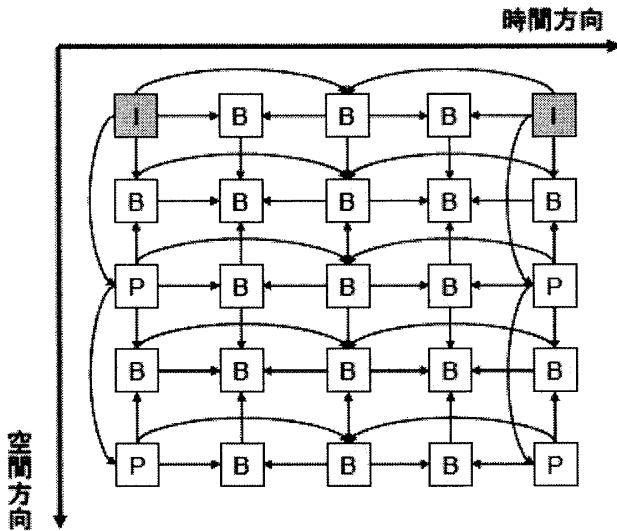


図 4.1 MVC の構成

5. 提案手法

5.1 提案手法の概要

本稿では直線カメラ配置により構築された光線空間の水平断面画像(EPI) 上での特徴に着目した圧縮符号化を提案する。その特徴とは、実空間上での1点はEPI上で直線の軌跡を描くという点と、その直線の傾きはその点の奥行きに依存するという点である。つまり、取得画像の画素列を直線により表現するという事は、撮影対象の幾何形状を持つということと同時に、冗長性の高い画素列を集めるということになる。

処理の流れとしては、まず、多視点の奥行き情報画像のEPIを直線分解し、一つに統合する。(GDMの作成) 次に、分解した直線に対応する色情報を多視点画像のEPIを参考に取得する。(TMの作成) この色情報を2次元状に並べることによって、多視点画像の統合を行う。そして、統合された画像の時間方向の冗長性を省くことによって圧縮符号化を行う。復号化後、仮想的な映像の生成はGDMとTMを用いることによりEPIや元の画像を復元することなく可能にする。

5.2 GDM(Global Depth Map)

5.2.1 GDMの概要

GDMは光線空間のある点を通る軌跡を奥行き画像を用いて補間し、全ての直線の傾きを表したものである。つまり、GDMは多視点の奥行き情報画像間を補完しながら一つに統合したものであると言える。これは奥行き画像のEPIを直線分解することと等価である。3×10pixelの奥行き情報画像のEPIの直線分解の例を図5.1に示す。全てのpixelの中心上において最適な傾きの直線を求める。その直線こそが光線空間のある点を通る軌跡であり、その全ての直線の傾きと通る点のデータがGDMとなる。

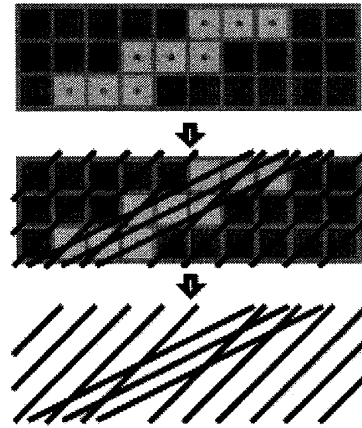


図 5.1 直線分解の例

5.2.2 奥行き情報画像EPIの直線分解の手法

本稿では左側から順に奥行き情報画像を直線で埋めていく手法を用いている。まず直線が通る候補となる点を奥行き情報画像のEPI上の全ての列のpixelに1つ設定する。始めに埋まる候補となる点は奥行き画像のEPI上の全ての列の一番左側のpixelの中心とする。それらの点を通るあらゆる傾きの直線の中で一番Costが低いものを選ぶ。Costはその直線が通るEPI内の全てのpixelにおいて、その輝度値とCostを計算している直線の傾きから算出した平均2乗誤差を用いている。また、奥行き情報画像のEPIを左側から直線で埋めているので最適なものとして選ばれる直線は埋まる点候補上もしくはその左側を通っていなければならない。

最適な直線が求められた後、次の最適な直線を求めるためには次に埋まる点候補の更新が必要となる。先に最適とされた直線が通る位置と同じ列の埋まる点候補とを比較する。直線が通る位置が同じ列の候補点と同じもしくは左側1pixel未満の位置にある場合、その列の候補の位置を1pixel右にずらす。こうすることによって左から順にEPIを直線で埋めることができる。一番右下のpixel上が最適な直線で埋められた時点で直線分解を終了する。

5.3 TM(Texture Map)

TMとは多視点画像を補完しながら統合したものである。これは、GDMの画像を統合する概念とは異なる。GDMが光線空間のある点の軌跡を表しているのに対し、TMはその軌跡の色を表しているものである。GDMの直線一つ一つに色情報を持たせ、それを画像表現することにより多視点画像を統合している。図5.2にTMの例を示す。

色情報を取得する方法として提案手法では、GDMを作成する際の奥行き情報画像のEPIに対応したカメラ画像のEPIから色情報を求める。具体的には、色情報を取得したい直線に対応するカメラ画像のEPI上に重ね、その直線上の色の平均値を色情報として取得した。

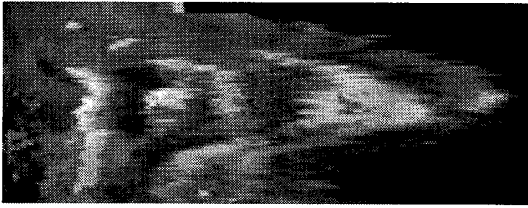


図 5.2 Texture Map の例

5.4 TM と GDM からの画像生成

TM と GDM はすでに密な光線空間に補完してあるものであると考えられる。GDM は被写体のある 1 点が光線空間においてどのような直線を描くのか、TM はその直線はどのような色をしているのかを表している。したがって、分解した直線に線形補間を行うことにより、実際にカメラで撮影した視点の画像を生成すると共にカメラ間の仮想的な視点の画像も生成することが可能である。

6. 実験

6.1 実験条件

提案手法の有効性を確かめるために、実際に図 6.1 のようなシーンを撮影したシーケンスとそれに対応する奥行き情報画像を用いて圧縮を行った。今回は多視点動画ではなく、静止シーンのシーケンスを用いて GDM と TM 作成しているため、時間方向の圧縮は行っていない。シーケンスは最大 25 枚で構成されており、それらは全て同じ方向を向き、視点が等間隔にずれているものである。画像サイズは 640×480 pixel、視差は $5 \sim 16$ pixel となっている。

6.2 実験内容

6.2.1 実験 1

2~25 枚のシーケンスにおいてそれぞれ提案手法で圧縮し、GDM と TM を作成後、TM の画像サイズと元のカメラ画像群の画像サイズの合計より、圧縮率を調べる。その後実際にカメラが存在する視点の画像を復元し、PSNR を測定する。シーケンスの枚数により圧縮率と PSNR がどのように変化するかを確かめる。

なお、シーケンスは左側の視点の画像から利用し、利用していない視点の画像において PSNR は測定しないものとする。

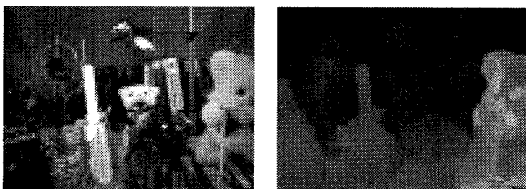


図 6.1 カメラ画像(左)とその奥行き情報画像(右)

6.3 結果

実験の圧縮率の測定結果を図 6.2 に示す。縦軸は圧縮率、横軸は圧縮し、GDM と TM にした画像の枚数となっている。PSNR を測定結果を図 6.3 に示す。縦軸は PSNR の平均、横軸は GDM と TM にした画像の枚数となっている。また、2, 5, 10, 15, 25 枚をそれぞれ圧縮し、復元した画像の一つを図 6.5, 6.6, 6.7, 6.8, 6.9 に示す。全て同じ視点であり、図 6.4 が原画像である。

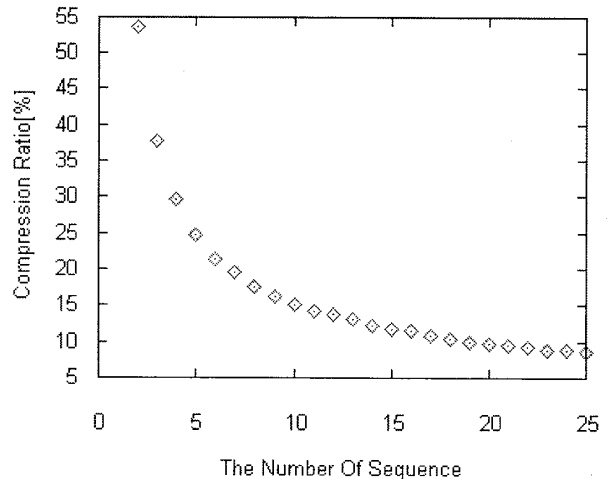


図 6.2 圧縮率[%]と統合した画像の枚数の関係

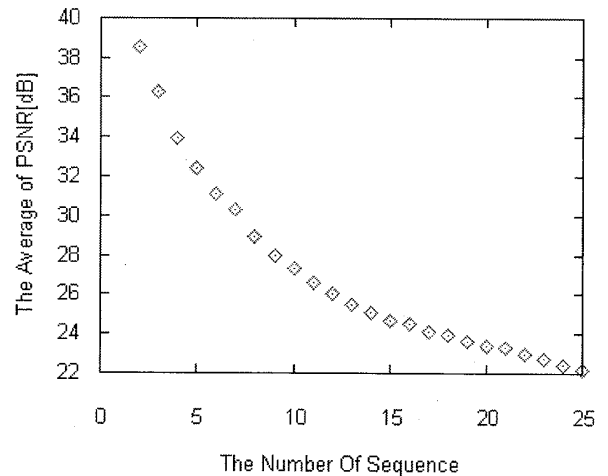


図 6.3 PSNR[dB]の平均と統合した画像の枚数の関係

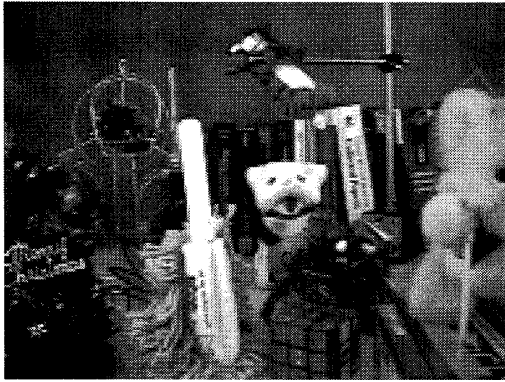


図 6.4 原画像

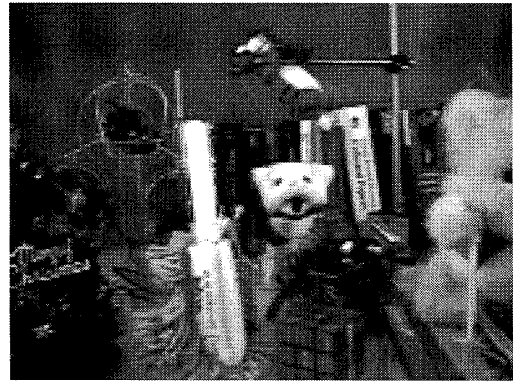


図 6.7 10枚を圧縮後復元した画像



図 6.5 2枚を圧縮後復元した画像

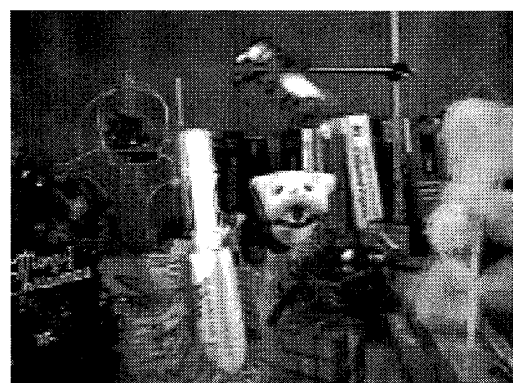


図 6.8 15枚を圧縮後復元した画像

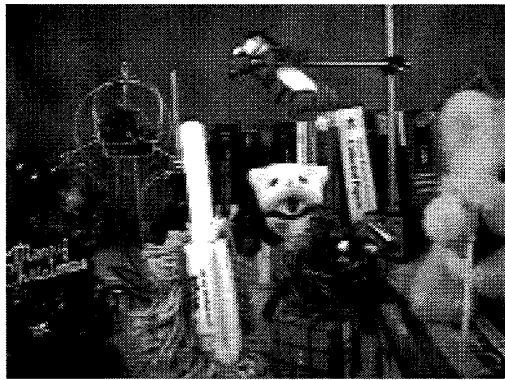


図 6.6 5枚を圧縮後復元した画像

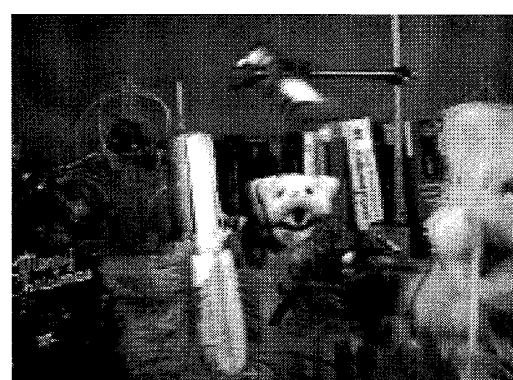


図 6.9 25枚を圧縮後復元した画像

6.4 考察

実験の結果の図 6.2 より圧縮する画像の枚数が多いほど圧縮率が良くなっているのがわかる。また、特に枚数が少ない時ほど圧縮率の変化が大きい。これは、圧縮する画像が少ないほど画像 1 枚 1 枚が直線分解に大きな影響を与えるからであると考えられる。

実験結果の図 6.3 より画像の枚数が多いほど PSNR の値が下がることが分かる。また、図 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 より 2 枚で圧縮後復元した画像は被写体の輪郭のぶれも少なくきれいなものとなっている。5 枚で圧縮後復元した画像では若干被写体の輪郭のぶれが見え始めている。10 枚で圧縮後復元した画像を見ると被写体の輪郭のぶれが目立つようになっている。さらに枚数を増やしていくと、徐々にそのぶれが大きくなっていることが分かる。この劣化の原因は、GDM と TM を作成する際、視点によっては見えない場所の存在が正しい直線分解と色情報を取得できなくしているのではないかと考えられる。また、奥行き情報画像の生成技術がまだ発展途上であり、理想的なものではないことが挙げられる。

7. むすび

本稿では直線的に配置された多視点画像において、従来とは違い、奥行き情報の EPI 特性を利用する新しいデータの圧縮方式を提案した。光線空間の光線源の軌跡を一つにまとめることによりデータ量を減らすことができた。そして、統合する画像の枚数を変えると圧縮率や PSNR がどのようなになるかを確認した。

しかしながら、復元した画像の、特に物体の境界線で少なからず劣化が起こってしまった。今後は更に劣化を抑えた奥行き情報画像 EPI の直線分解の手法と TM の作成法の検討、シーケンスに動画を用いての実験、カメラ間隔を広げての圧縮率や PSNR の測定を行う予定である。

参考文献

- [1] T. Fujii and M. Tanimoto, "Free-viewpoint Television based on the Ray-Space representation", SPIE ITCOM'02, pp.175-189, Aug.2002.
- [2] 松山隆司 "3次元ビデオ映像", 3D Image Conference 2004, pp.151-156, 2004.
- [3] S.C. Chan, H.Y. Shum, and K.T. Ng, "Image-based rendering and synthesis", IEEE Signal Processing Mag. vol. 24, no. 7, pp. 22-33, Nov. 2007.
- [4] M. Levoy, P. Hanrahan, "Light Field Rendering", Proc. ACM SIGGRAPH'96, pp. 31-42, 1996.
- [5] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The Lumigraph", Proc. ACM SIGGRAPH'96, pp. 43-54, 1996.
- [6] T. Fujii, "テレビ誌 Vol.50", No.9, pp.1312-1318, 1996.
- [7] N. Fukushima, et al., "Proc. SPIE vol.58" 6016, 2005.1
- [8] Jin-Xiang Chai, Shing-Chow Chan, Heung-Yeung Shum, Xin Tong, "Plenoptic sampling", Proc.SIGGRAPH '00, pp.307-318, 2000
- [9] 藤井俊彰, 森健策, 武田一哉, 間瀬健二, 谷本正幸, 末永康仁: "大規模実世界データベース構築のための多元多点計測装置の開発", 3次元画像コンファレンス 2006, pp. 41-44 (2006.7)

- [10] T.Yamamoto, M.Kojima, T.Naemura, "LIFLET:Light Field Live with Thousands of Lenslets", SIGGRAPH 2004, etech 0130(2004)
- [11] ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC 14496-10:200X/FDAM 1 Multiview Video Coding", Doc. N9978, Hannover, Germany, July 2008.