

# 仮想平面を用いた多視点映像の予測内挿符号化方式の検討

鈴木 裕高†  
Hirotaka Suzuki

久保田 彰†  
Akira Kubota

羽鳥好律†  
Yoshinori Hatrori

## 1. まえがき

多視点映像とはあらかじめ複数のカメラでシーンを撮影し、視聴者が自由に視点を変えながら視聴する映像表現である。多視点映像通信では、送信側ではあらかじめカメラの位置情報がキャリブレーションされており、この位置情報を多視点映像データとともに圧縮して送信する。受信側ではこの多視点映像データをカメラの位置情報をもとに復号する。

多視点映像の圧縮符号化では、従来、多視点映像データを動画像として扱い、H.264などのブロックマッチングを適用する手法が多く提案されていた。この手法は膨大な多視点映像データを効率よく圧縮することが期待できる。しかし、カメラの間隔を広く取ることができず、設置するカメラの台数が多くなってしまふ。そこで、我々は仮想平面を使った予測符号化を提案する。提案手法ではカメラ間隔を広く取ることが可能になると期待される。

## 2. 仮想平面を用いた予測画像生成

本研究では仮想平面を用いて予測画像を生成し、圧縮符号化を行う。仮想平面とは空間全体をカメラからある距離にある一様な平面として仮定したものである。この仮想平面をカメラの前方に置き、左右両側のカメラ画像から対応点を求める。左右のカメラ画像の対応する画素情報を参照し、真ん中のカメラの予測画像を生成する。このとき、シーンの中で仮想平面と同じ距離にある物体は鮮鋭に合成されるが、仮想平面から離れた距離にある物体はうまく合成することができない。そこで、我々はシーンに複数の仮想平面を設定した。

今回のシミュレーションではカメラをこのステージの周囲に等間隔に円周上に配置して撮影を行った。よってカメラの配置は図1ようになる。

まず、カメラ#0の前方に仮想平面を設定する。その平面上に空間全体が一様に存在するとして、カメラ#0の各画素

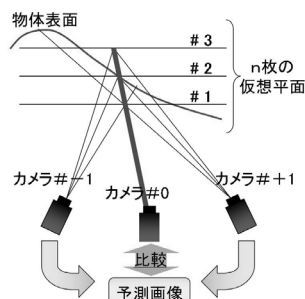


図1: 仮想平面

が左右のカメラ#-1と#+1のどの画素に写るのかを計算する。こうしてカメラ#-1と#+1の対応する座標の画素値を平均し

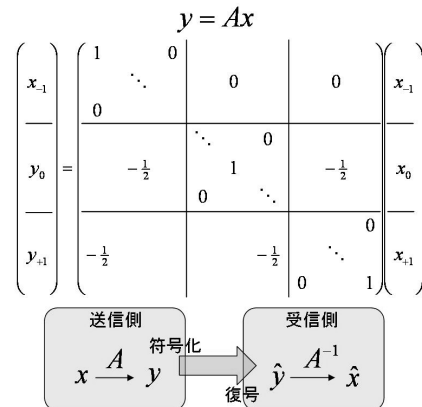


図2: 多視点映像符号化

て、カメラ#0の予測値とする。

この作業を全ての仮想平面に対して行い、この予測画像をそれぞれ実際に撮影された真ん中のカメラ画像と比較し、その差分を予測誤差とする。予測誤差が最も小さかった仮想平面の情報がその画素で適用される。こうして画素ごとに参照する仮想平面の位置が決定する。

全ての画素において参照する仮想平面の位置を決定すると、設定した仮想平面の枚数分だけのレベルに分けられた奥行きマップが生成される。この奥行きマップの情報と予測誤差を用いることでカメラ#0の画像が再構成される。

## 3. 内挿予測符号化

今回の提案手法では内挿予測符号化による圧縮を想定している。前章で説明した仮想平面の情報と予測誤差を用いて内挿予測符号化を行う。前述のように仮想平面をもとに、座標を変換すると参照画素の位置を求めることができた。求めたカメラ#-1と#+1のカメラ画像の参照画素値の平均と実際のカメラ#0の画素値を比較することで予測誤差  $y$  を以下のように求めることができる。

$$y = x_{0(m,n)} - \frac{1}{2}(x_{-1(s,t)} + x_{+1(u,v)}) \quad (3.1)$$

$x_{0(m,n)}$   $x_{-1(s,t)}$   $x_{+1(u,v)}$  はそれぞれカメラ#-1,#0,#+1の座標  $(m,n)$ ,  $(s,t)$ ,  $(u,v)$  における画素値である。 $(s,t)$ ,  $(u,v)$  は  $(m,n)$  の対応点である。

図3に内挿予測の概要を示す。図2はカメラが3台の場合の内挿予測方式である。右側の行列  $x$  は3台のカメラの1ラインの画素値である。式(3.1)を全ての画素に適用すると予測誤差  $y$  を求めるための変換行列  $A$  を決定することができる。つまり、画素ごとの仮想平面の位置(奥行きマップ)が決まれば、変換行列  $A$  を得ることができる。送信側では、仮想平面の参照位置情報をもとに行列  $A$  を決定し、予測誤差行列  $y$  を生成する。そして、この  $y$  行列と奥行きマップが圧縮され送信される。受信側では、奥

† 東京工業大学 大学院 総合理工学研究科  
物理情報システム専攻羽鳥研究室



4(a)



4(b)

4(c)

図 3: それぞれのカメラから撮影された元の画像



図 4: 奥行きマップ

行きマップから  $A$  を求め、送られてきた  $y$  に  $A^{-1}$  をかけることで画像  $x$  を再現する。

#### 4. シミュレーション

今回のシミュレーションは、ステージ上に数人の人を配置し、カメラをこのステージの周囲に等間隔に 30 台円状に配置して撮影したデータセットを使用した。そのうちの 3 台のカメラの画像を用いて左右のカメラ画像から真ん中のカメラ画像を予測、再構成した。予測する真ん中のカメラ画像は図 3(a) である。そのカメラの左右から撮影されたカメラ画像は図 3(b), 3(c) である。

今回は仮想平面を 16 枚、32 枚、64 枚と設定してシミュレーションを行った。図 4 に仮想平面を 64 枚仮定した場合の (64 段階の奥行きで表される) 奥行きマップを示す。白い領域がより近い距離にある仮想平面を参照していることを表している。図 4 を見ると奥行きが正確に推定されていない領域がある。これは、オクルージョンや光沢などの影響で、視点によって見える輝度値が極端に変化する領域である。このような領域では画素値をうまく予測できず予測誤差が大きくなる傾向にある。

次に仮想平面の枚数を変えて測誤差のエントロピーを求めた。仮想平面が 16 枚のとき 4.6bit/pixel、32 枚のとき 4.1bit/pixel、64 枚のとき 3.7bit/pixel だった。仮想平面を増やすとエントロピーが減少し、高い圧縮効率が期待できることがわかった。

提案手法の有効性を示すために従来のブロックマッチングによる手法も検証した。ブロックサイズを 8 画素×8 画素とし、参照ブロックと最も相関の高いブロックを隣のカメラの画像から 1 画素単位で探索し、予測誤差のエントロピーを求めた。これと提案手法の予測誤差のエントロピーを比較したところ、提案手法 (仮想平面 64 枚) のほうが 0.5bit/pixel ながら低くなった。

次に図 4 の奥行きマップを圧縮し、カメラ#0 の予測画像を生成した。PNG (可逆圧縮) で圧縮した奥行きマップを用いて予測した画像が図 5(a) である。これは実際に撮影さ



5(a)

5(b)

図 5: 予測画像

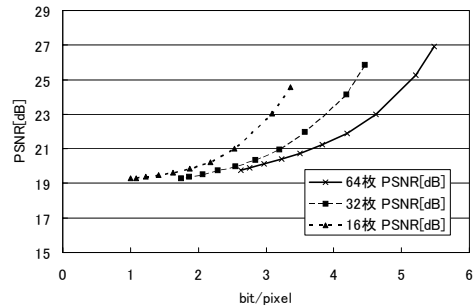


図 6: 圧縮率と SNR

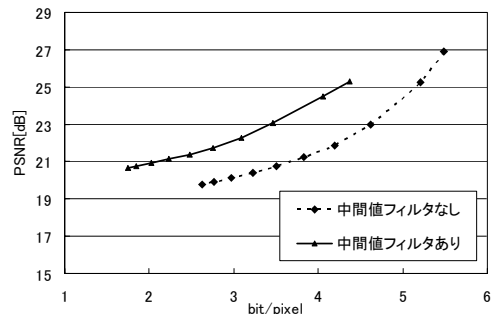


図 7: 中間値フィルタの効果

れた画像 (図 3(a)) とほぼ同じ画像が予測できている。次に JPEG(30%)で圧縮した奥行きマップを用いて予測した画像が図 5(b) である。不可逆圧縮のため予測画像の画質が劣化していることがわかる。これは圧縮によって奥行きマップにおける仮想平面の位置が正しく判定されなくなるためである。今後、奥行きマップを平滑化するなど最適処理が必要である。

また、奥行きマップを徐々に圧縮した場合の予測画像の PSNR を図 6 に示す。仮想平面が多いときは圧縮の影響を受けて PSNR が著しく低下してしまうことがわかる。

さらに、中間値フィルタを用いて奥行きマップの粒状ノイズを除去してから圧縮した。これを比較したのが図 7 である。圧縮の前に中間値フィルタをかけると圧縮率が向上することがわかった。

#### 5. まとめ

内挿予測符号化方式のための予備実験として、仮想平面の位置情報と予測誤差から画像の再構成を行った。

今後は 3 章で説明した内挿予測方式を検討していく。内挿予測方式によって符号化したデータをもとに画像を再構成する予定である。

#### 6. 参考文献

Yoshinori, H. "Optimal Quantizing Schemes in Interpolative Prediction." IEICE Japan, vol. J66-B, no. 5, pp. 599-606. 1983.