

ステレオ画像圧縮のための視差補償法に関する検討

原, 学

九州大学大学院システム情報科学府知能システム学専攻 : 修士課程

内田, 誠一

九州大学大学院システム情報科学府知能システム学部門

迫江, 博昭

九州大学大学院システム情報科学府知能システム学部門

<https://doi.org/10.15017/1516209>

出版情報 : 九州大学大学院システム情報科学紀要. 10 (1), pp.57-62, 2005-03-25. 九州大学大学院システム情報科学府知能システム学部門

バージョン :

権利関係 :



ステレオ画像圧縮のための視差補償法に関する検討

原 学*・内田誠一**・迫江博昭**

A Data Compression Technique for Stereo-pairs Using Pixel-based Disparity Compensation

Manabu HARA, Seichi UCHIDA and Hiroaki SAKOE

(Received December 24, 2004)

Abstract: In this paper, we describe a data compression technique for stereo-pairs using pixel-based disparity compensation (DC). The bit-rate of prediction residual of the proposed pixel-based DC is lower than that of block-based DC, which has been used commonly. Although the bit-rate of disparity becomes high on the pixel-based DC, we can relax this problem by imposing several restrictions on DC. It is also shown that the performance of the proposed technique can be improved by the following two modifications. Firstly, previous pixel prediction is selectively used around occlusion areas. Secondly, instead of pixels, blocks of one pixel width are employed as a unit of DC. The effectiveness of the proposed technique is observed through experiments.

Keywords: Stereo image compression, Disparity compensation, Predictive coding, Previous pixel prediction, Occlusion, Dynamic programming

1. はじめに

本論文では予測に基づくステレオ画像圧縮の一方式を提案する。ステレオ画像は、同じ物体を異なる角度で撮影したものであるため、左右画像の同一エピポーラ線上に近い値を持つ画素対が多く存在する。これらの画素対には撮影された物体の奥行きに応じた視差が存在する。この視差を推定し、それを補償しながら予測に用いる画素を決定することで、予測残差の符号量を大幅に低減できる。既存のステレオ画像圧縮技術¹⁾⁻⁸⁾の多くはこの性質を利用している。

従来のステレオ画像圧縮手法では、ブロックを単位として視差補償を行っている。視差の似通った画素からなるブロックを一括に対応付けるため、視差の符号量を小さくできる。反面、視差境界がブロックに含まれると残差の符号量が大きくなってしまいう問題がある。

本論文では画素レベルで視差補償を行う手法を提案する。画素レベルで視差補償を行うため、ブロック単位での視差補償より残差の符号量を低減できる。これに伴い増加する視差の符号量については、視差に制約を加えることで抑制する。

本論文では、以上の画素レベル視差補償法の提案に加え、以下の2点による視差と残差の符号量の更なる低減についても検討する。

第1は、オクルージョン領域など適切な対応点のない画素について、同一画像上の隣接画素を予測に用いる方法(前値予測)に切り替えるという改良である。例えば、オクルージョン領域には対応する画素が存在しないため、左右のエピポーラ線上で視差補償を行っても残差が大きくなってしまふ。そこで、そうした画素については適宜前値予測に切り替えることで残差を最小化する。この手法は、オクルージョン領域だけでなくノイズなどが原因で適切なステレオ予測ができない場合にも有効と考えられる。

第2は、エピポーラ線に対して垂直方向に細長いブロックを単位として視差補償を行うという改良である。一般にステレオ画像では、垂直方向に同じ位置にある画素群の視差がほぼ等しい場合が多い。従って、こうした画素群を単位として視差補償を行うことで、残差の増加を抑えつつ視差の符号量を低減できると考えられる。

本手法は視差補償に基づく予測符号化法の一つであり、原理的に可逆圧縮にも非可逆圧縮にも適用できるが、本論文ではこれを可逆圧縮の枠組みで評価する。なお本手法を非可逆圧縮に適用する場合は、SN比^{2),3)}および3次元情報復元精度に関する主観評価^{4),5)}等の評価手段が考えられるが、いずれも今後の課題と考えている。

2. 画素レベルの視差補償

ステレオ画像の予測符号化のために左右同一のエピポーラ線上での視差補償を行う。この際画素を単位とすることで、ブロック単位よりも原理的に高い精度の視差補償が可能である。すなわち、予測残差を小さくできる

平成16年12月24日受付

* 知能システム学専攻修士課程

** 知能システム学部

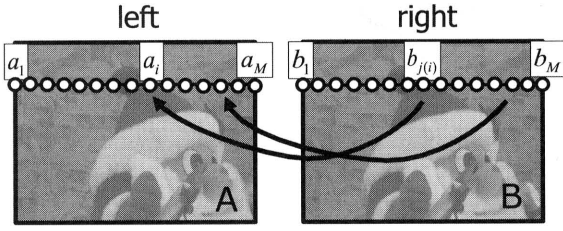


Fig. 1 Pixel-based disparity detection.

```

/* Initialization */
1  g(1,1) = ||a_1 - b_1||
/* DP-Recursion */
2  for i = 2 to M do begin
3    for j = 1 to M do begin
4      g(i,j) = ||a_i - b_j|| + min_{p=0,1,2} g(i-1, j-p)
5      p(i,j) = arg min_{p=0,1,2} g(i-1, j-p)
6    end
7  end
/* Backtrack operation */
8  j = M
9  for i = M downto 2 do begin
10   Δa_i = a_i - b_j
11   Δj(i) = p(i, j)
12   j = j - Δj(i)
13 end
14 Δa_1 = a_1 - b_1
    
```

Fig. 2 DP algorithm for stereo prediction.

ため、その符号量の大幅な低減を期待できる。

2.1 視差関数の定義

左の画像をA、右の画像をBとする。両画像はエピポーラ線が同一行になるようキャリブレーションされているものとする。また、以下では右画像Bは別途何らかの方法で符号化されるものとし、左画像Aの符号化について考える。画像Aと画像Bの第*l*行($1 \leq l \leq L$)の画素値系列をそれぞれ $a_1, a_2, \dots, a_i, \dots, a_M, b_1, b_2, \dots, b_j, \dots, b_M$ と表す。ここで a_i および b_j はそれぞれRGB成分よりなる3次元ベクトルとする。

左画像Aと右画像Bには、撮影された物体の奥行きに応じた視差がある。この視差を各行毎に、視差関数 $j = j(i)$ として表現し、その最適推定を行う。視差関数 $j(i)$ により a_i は $b_{j(i)}$ に対応付けられる(Fig.1)。

視差関数 $j(i)$ による画素対応の結果を用いて、画素 a_i の予測残差を

$$\Delta a_i = a_i - b_{j(i)} \quad (1)$$

と定義する。エントロピー符号化を行うためにはこの残差を小さく、すなわち Δa_i を0付近に集中させる必要がある。従って、視差関数 $j(i)$ の最適推定の基準を以下のように定める。

$$\sum_{i=1}^M \|a_i - b_{j(i)}\| \rightarrow \text{minimize} \quad (2)$$

この基準により、視差関数によって対応付けられる画素間の残差が同一エピポーラ線間で最小化される。

2.2 視差の制約

本手法では各画素について視差情報 $j(i)$ を伝送する必要がある。この視差の符号量を低減させるため、視差差分 $\Delta j(i) = j(i) - j(i-1)$ に次の単調連続性制約を課する。

$$0 \leq \Delta j(i) \leq 2 \quad (3)$$

視差 $j(i)$ の代わりにこの差分 $\Delta j(i)$ を伝送することになると、視差情報の符号量は $\log_2 3$ ビット/画素に抑え込める。

2.3 DPによる最適視差の決定

同一エピポーラ線上の画素間の視差関数決定問題、すなわち制約条件(3)の下での目的関数(2)の最小化問題は、動的計画法(DP)により効率的に解くことができる⁹⁾。アルゴリズムをFig.2に示す。ここで、 $g(i, j)$ は最小コスト、 $p(i, j)$ はバックポイントである。ステップ4のいわゆるDP漸化式を計算していくことで、最終的に $g(M, M)$ として目的関数(2)の最小値が得られる。ステップ8以降はバックトラック処理と呼ばれるもので、計算済みの $p(i, j)$ を用いて、 $i = M$ から1まで遡りながら、 $j(i), \Delta j(i), \Delta a_i$ を求めている。

符号化器では、以上で得られた Δa_i と $\Delta j(i)$ をエントロピー符号化したものを全ての*i*について送信する。復号の際は、 $\Delta j(i)$ から $j(i)$ を復元し、 Δa_i と別途送信されている b_j から a_i を復元すればよい。復号の際はDPによる視差最適化を行う必要はなく、非常に高速な処理となる。

3. 予測方式の自動切替

残差符号量のさらなる低減をねらった、予測方式の自動切替について検討する。例えばオクルージョン領域では、左右画像上に対応する画素対が存在しないため、視差補償を行っても残差が大きくなってしまふ。また、ノイズや照明条件によって、本来対応すべき画素であっても

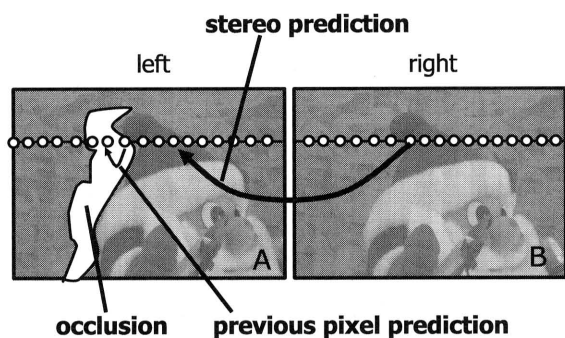


Fig. 3 Selective prediction.

```

/* Initialization */
1   $g(1, 1) = \|a_1 - b_1\|$ 
/* DP-Recursion */
2  for  $i = 2$  to  $M$  do begin
3    for  $j = 1$  to  $M$  do begin
4      if  $\|a_i - b_j\| < \|a_i - a_{i-1}\|$  then begin
/* Stereo prediction */
           $g(i, j) = \|a_i - b_j\| + \min_{p=0,1,2} g(i-1, j-p)$ 
           $f(i, j) = 0$ 
        end
      else begin
/* Previous pixel prediction */
           $g(i, j) = \|a_i - a_{i-1}\| + \min_{p=0,1,2} g(i-1, j-p)$ 
           $f(i, j) = 1$ 
        end
5       $p(i, j) = \arg \min_{p=0,1,2} g(i-1, j-p)$ 
6    end
7  end
/* Backtrack operation */
8   $j = M$ 
9  for  $i = M$  downto 2 do begin
10   if  $f(i, j) = 0$  then
         $\Delta a_i = a_i - b_j$ 
      if  $f(i, j) = 1$  then
         $\Delta a_i = a_i - a_{i-1}$ 
11    $\Delta j(i) = p(i, j)$ 
12    $j = j - \Delta j(i)$ 
13 end
14  $\Delta a_1 = a_1 - b_1$ 

```

Fig. 4 DP algorithm for selective prediction.

その輝度値は大きく異なる場合もある。そこで、これまで述べてきたもう一方の画像から予測(ステレオ予測)する方法と、同一画像上の隣接画素との残差を用いて予測する方法(前値予測)とを自動的に切り替える手法を提案

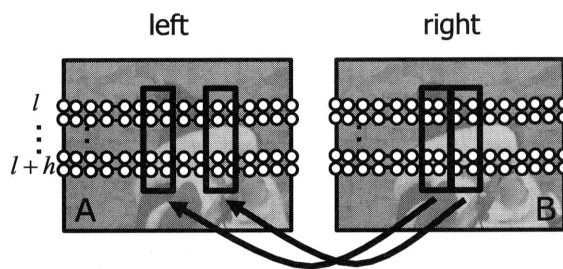


Fig. 5 Stripping.

する(Fig.3). これにより、オクルージョン領域などで適切な対応画素が右画像B上にはない場合には同一画像Aから予測画素が選ばれ残差を小さくできる。

以上の予測切替を組み込んだDPアルゴリズムをFig.4に示す。ここで $f(i, j)$ は予測切替情報である。各 i において前値予測による残差 $\Delta a_i = a_i - a_{i+1}$ の絶対値が視差関数による対応画素との残差(1)の絶対値よりも小さければ、そちらを選ぶように漸化式を変更する。すなわち、Fig.2のステップ1, 4において、前値予測による残差とステレオ予測による残差(1)の大小関係を判断する。そして、ステップ10, 14のバックトラック処理において、 $f(i, j)$ を用いた予測切替を行い、 Δa_i を求める。このようにDPアルゴリズムによって、前値予測への切替が残差最小化の枠組みで統一かつ自動的に処理できる。

4. 縦ブロック化

視差符号量のさらなる低減をねらって、ある程度の精度を保ちながら部分的にブロックマッチングの考えを採用した、視差補償の縦ブロック化について述べる。

キャリブレーションされたステレオ画像では、垂直に同じ位置にある画素同士の視差は類似している可能性が一般に高い。そこで、あくまで水平方向には画素単位の視差補償を実現しながら、横幅が1の垂直方向に長い縦ブロックを単位として視差補償を行うことを考える(Fig.5)。これらによって残差の増加を抑えつつ1画素あたりの視差情報の符号量を小さくできる。この場合のアルゴリズムも基本的には以上と同じであり、隣接縦ブロック間には式(3)と同様の単調連続性制約を課される。

5. 実験

本手法を可逆圧縮に用いた場合の性能評価実験を行った。

5.1 使用画像

実験では、Czech Technical UniversityのWeb上で公開されている画像(grove)¹⁰⁾、筑波大学多視点画像データベース(sant)、および当研究室で実際にカメラ(Point Grey Research社, Bumblebee BB-COL-60)を

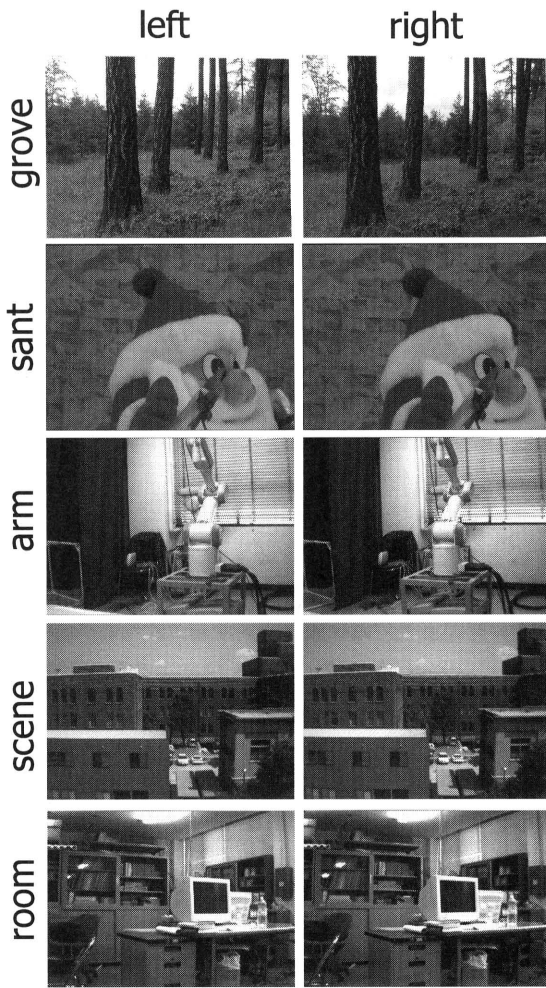


Fig. 6 Stereo images used in the experiment.

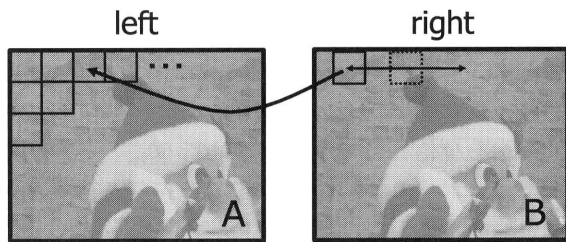


Fig. 7 Block matching.

用いて撮影したステレオ画像 (arm, scene, room) を用いた。Fig. 6 に実験で用いた画像を示す。なお、筑波大のデータベースには他のデータも準備されているが、実験結果はほぼ同一傾向であったので略す。

5.2 比較対象

比較評価の対象として以下に述べるブロックマッチング法¹⁾を用いた。

まず画像Aを一定の大きさのブロックに分割し、各ブロックを単位として対応するブロックを画像Bから探索

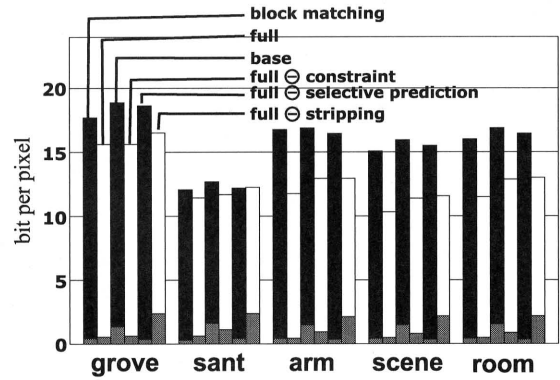


Fig. 8 Compression result.

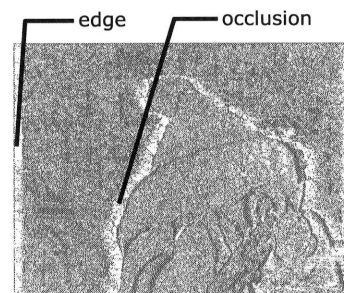


Fig. 9 Pixels (displayed as white dots) where selective prediction is used.

する (Fig. 7)。ブロックの探索は、次式のような二乗誤差が最小となるように行う。

$$\sum_l \sum_m \|a_{l,m} - b_{l+j(k),m}\|^2 \quad (4)$$

ここで、 $a_{l,m}$ 、 $b_{l,m}$ は画像A、Bそれぞれのブロック内に位置する第 l 行 m 列目の画素値、 $j(k)$ は第 k 番目のブロックの対応関係を表す視差関数である。探索をブロック単位で行うため、差分の符号量が大きくなる反面、視差の符号量が小さいため全体の符号量を小さくできる。

さらに3節および4節で述べた改良の効果を見るために、Table 1 のように予測切替あるいは縦ブロック化のいずれか一方のみを、画素単位の視差補償にそれぞれ組み込んだ手法についても比較実験を行った。

5.3 評価基準

本実験では、評価基準として左画像の1画素当りの符号量を用いた。具体的には左画像に対して、残差、視差差分独立にハフマン符号化を行い、全画素分の残差符号量、視差差分符号量を算出した。それにハフマン木と予測切替情報の符号量を加算して、全画素数で除したものを評価基準とした。

Table 1 Prediction method examined in the experiment.

methods		pixel-based DC (sec 2.1)	constraint (sec 2.2)	selective prediction (sec 3)	stripping (sec 4)	block-based DC (sec5.2)
block matching						○
proposed	full	○	○	○	○	
	base	○	○			
	full \ominus constraint	○		○	○	
	full \ominus selective prediction	○	○		○	
	full \ominus stripping	○	○	○		

5.4 結果と考察

各手法による符号量をFig.8に示す。各棒グラフにおいて上側が残差の符号量、下側が視差の符号量である。また、縦ブロック化の際のブロックサイズは 4×1 、ブロックマッチング法では 4×4 とした。

実験の結果、提案手法(full)はブロックマッチング法よりも符号量が平均で3.19bpp低減できた。この理由として以下の4つが考えられる。

第1に、画素レベルで視差補償を行ったことにより残差の符号量を低減できたことが考えられる。実際に、ブロック単位で視差補償を行った手法と比べて、画素レベルで視差補償を行った手法(base)では残差の符号量が平均で1.92bpp低減している。

第2に、視差に制約条件を加えたことにより視差の符号量を低減できたことが考えられる。実際、視差に制約条件を加えない手法(full \ominus constraint)と比べて、視差に制約条件を加えた手法(full)では視差の符号量が平均で0.36bpp低減している。

第3に、予測切替によって予測残差を極小化できたことが考えられる。予測切替が組み込まれていない手法(full \ominus selective prediction)と比べて、予測切替が組み込まれた手法(full)では残差の符号量が平均で4.23bpp低減している。Fig.9は、sant画像における予測切替の結果、前値予測が選択された画素を白色で示したものである。この図を見ると、特にオクルージョン領域や画像の端部といった対応する画素が存在しない領域で前値予測が選択されていることが分かる。なお、Fig.8のグラフには示していないが、ブロックベースの視差補償に予測切替を組み込んで、提案手法(full)より性能的に劣ることを確認している。このことは、ブロックベースの視差補償に予測切替を加えるよりも、画素ベースの視差補償に予測切替を加えた方が相性がよいことを示唆していると言える。

第4に、縦ブロックベースの視差補償によって、残差の符号量の増加を抑えつつ視差の符号量を低減できたことが考えられる。実際、縦ブロックベースの視差補償を用いない手法(full \ominus stripping)と比べて、用いた手法(full)では残差の符号量が0.61bpp増加しているが、視差の符号量は1.97bpp低減している。このことは、視差は縦ブロック内(同一水平位置)において類似していることを裏づけていると言える。

6. 結 論

本論文では、画素レベルの視差補償に基づくステレオ画像可逆圧縮手法を提案し、さらに予測切替と縦ブロックベースの視差補償を組み込むことで圧縮性能の向上を図ることを提案した。そして、本手法の評価実験を行い、従来法であるブロックベースの視差補償に比べて符号量を1画素当たり平均で3.19bpp低減できたことを確認した。

謝辞

本研究の一部は文部科学省科学研究費補助金(No.14780293)によった。

参 考 文 献

- 1) M. G. Perkins, "Data Compression of Stereopairs," IEEE Trans. on Communications, vol.40, no.4, pp.684-696, 1992.
- 2) 新井陽介, 岡本教佳, 南 敏, "ブロックマッチングを用いたステレオ動画像の符号化," 電子情報通信学会総合大会, D-11-5, March 2000.
- 3) 田中耕平, 福田光一, 川中 彰, "視差補償を用いたステレオ画像符号化," 電子情報通信学会総合大会, D-307, March 1996.
- 4) 浜村倫行, 権田晃平, 小泉博一, 相田 仁, 斉藤忠夫, "視差情報を用いた多眼3次元画像の効率的符号化法," 情報処理学会研究報告CVIM, no.110-004, pp.25-32, 1997.
- 5) 斉藤義孝, 桐澤 潔, 涌井秀治, "左右画像共通領域抽出によるステレオ画像符号化," 電子情報通信学会システムソサイエティ大会, D-11-67, Sep. 1997.
- 6) 浜村倫行, 権田晃平, 小泉博一, 相田 仁, 斉藤忠夫, "ステレオ画像の視差検出と符号化への応用," 情報処理学会研究報告CVIM, no.115-009, pp.65-72, 1998.
- 7) 赤松秀樹, 奈倉理一, 森本雅和, "多方向撮像ステレオ画像

- の高効率データ圧縮方式に関する研究,” 信学技報, PRMU99-168, pp.21-26, Dec. 1992.
- 8) 山崎浩一, 森本雅和, 奈倉理一, “衛星ステレオ画像データの可逆符号化方式に関する検討,” 信学技報, CS98-124, pp.73-78, Dec. 1998.
- 9) 大田友一, 山田博三, “動的計画法によるパターンマッチング,” 信学論, vol.30, no.9, pp.1058-1066, Sep. 1989.
- 10) <http://cmp.felk.cvut.cz/cmp/demos/Stereo/New/Matching/smm.html>
- ~~~~~