

サッカーシミュレーションログデータの圧縮フォーマット

西野順二

電気通信大学 システム工学科

サッカーに代表されるマルチエージェントゲームは、観賞という観点からみても面白い対象となっている。しかしながら、その行動記録の保存方式については、ほとんど議論されず、しかもその容量が本質的に巨大になりやすい。本論文では、PDA など記憶資源が厳しく限られた端末で、サッカー型の多人数ゲームの試合再生を行うための圧縮フォーマットについて検討する。動画などと同様、人間の観賞を前提にした非可逆符号化を前提に、フォーマットを幾つか提案し、RoboCup サッカーシミュレーションのログを対象にして、方式ごとのパフォーマンスを比較検討した。

A soccer log data compression format OZ Soccer To Go

Junji Nishino

Dept. of Systems engineering

The University of Electro-Communications

Multi-Agent-Games, such as soccer games, are exciting target of study, and they are also exciting to watch in joy. However, the data expression and compression method of them is not supported sufficiently. This paper shows several compression methods for soccer simulation log data. As a result, we showed that the LP+VQ with SOM method worked very well for Multi-Agent-Game logs.

1 はじめに

本論文は、多人数ゲームの記録を取る方式について、ユーザによる娯楽的観賞を目的とした、符号化と圧縮表現の方法について検討することを目的とする。

PDA などの携帯情報端末の進歩に伴い、音声や動画をこうした機器により利用するスタイルが増えている。処理速度や記憶容量に制限のあるこうした環境下では、データの圧縮が必要不可欠である。いっぽう、サッカーに代表される多人数ゲームをより臨場感のある再生には、全体を眺めたりボール周辺だけを眺めたりできるスケーラビリティが必要である。計算機的には、各プレイヤーやボールの位置情報を、時間経

過にともなってコーディングすれば良いことになるが、その容量は大きなものとなる。行動分析など学術目的のための符号化では可逆性が不可欠だが、人による観賞を目的とすればそれに応じた非可逆的な圧縮の可能性はある。

一般に、あるデータに対して最適な情報圧縮の手法は目的に応じて異なるものである。可逆なデータ圧縮と、静止画像、動画像および音声に関する非可逆圧縮については、市場の要求から広範な研究がなされており [3, 4]、ほぼ完了していると言っても過言ではない。

いっぽうでサッカーのような多人数ゲームの行動記録について、効率的に圧縮する方法についての検討はほとんどみられない。

本研究では、多人数ゲームの行動記録を人による観賞を主たる目的とした圧縮について検討する。具体的な検討のため、RoboCup サッカーシミュレーションリーグのゲームログを対象として取り上げる。

2 Soccer Server Log とその観賞

RoboCup サッカーシミュレーションリーグは、人間対人工プレイヤーのサッカーゲームを目指した RoboCup プロジェクトのうち、シミュレートされた環境で行うサッカーゲームである [2, 1]。1対1の人工プレイヤーがそれぞれ個別のプログラムで作成され、ネットワークを介してバーチャルサッカー場であるサッカーサーバに接続し、サッカーの試合を行う。サッカーサーバはボールや各プレイヤーの物理運動をシミュレートし、100 ミリ秒単位の実時間で各プレイヤーと通信をすることで、リアルタイム制約のあるバーチャルなサッカー場を実現している。

ボールと各プレイヤーの行動、すなわち試合の全体像はサッカーサーバによってログとして記録されている。

近年、人工プレイヤーの行動判断のレベルが上昇している。試合のログを見ているかぎりでは、人間の行っているサッカープレイとみまごうばかりと言われている [5]。単純にサッカーの試合のリプレイとしての観賞にも耐えるものとなっている。

より高度な戦略を目指す上では、試合結果についてのサッカーの専門家による評価のフィードバックも求められる。このとき評価の記述と実際の行動を時間軸に沿って対応させ記録する必要があるが、行動記録のサイズがあまり大きくなると実装において非現実的となる。XMLなどと連係して、PDA 上で保存できる程度のサイズを目指して検討する。

3 サッカーとしての圧縮

ここでは、対象となるサッカーの試合が持つ情報量について、そのモデル化と見積もりを行なう。

3.1 サッカー試合情報のモデル化

サッカーの試合内容を再現する情報は、全てのプレイヤーとボールの動きが主体である。このほかに、プレイヤーの手や姿勢、首の向き、表情、といった詳細情報や、風の状態、芝の状態、照明、天候の状態といった外乱要因情報もある。しかしこれらは、ボールを取り合うゲームとして見たサッカー試合の情報の再生においては、さして重要ではない。

プレイヤーは試合開始 $t = 0$ から終了時刻 $t = T$ までの間、一連の移動を行なっており、これがゲームの本質である。よってプレイヤー i の試合全体での行動は、時間をパラメータとする平面座標の関数 $P_i(t) = (x_i(t), y_i(t))$ で表される。2人のプレイヤーとボールも同様に含め次の23点さらに x, y 軸に分け、46次元のベクトル関数となる。

$$P(t) = (P_0(t) \cdots P_{22}(t))$$

ただし、 $0 \leq t \leq T$ 、 $i = 0$ はボール、 $i = 1 \sim 11$ はチーム L のプレイヤー、 $i = 12 \sim 22$ はチーム R のプレイヤーを表す。

一般に $P(t)$ は連続な関数となるが、実際上は、適当な t でサンプリングし、センサシステムによる量子化が行なわれる。ここでも $P(t)$ は離散化された時系列データとして扱う。

3.2 基礎的なサイズ見積

RoboCup サッカーは11ずつのエージェント2チーム、合計22プレイヤーにより行なわれる。競技場は $105.0\text{m} \times 68.0\text{m}$ であり、試合時間は600秒間で、行動1ステップの時間刻み、すなわちサンプリングタイムは0.1秒となっており、試合全体は6000ステップとなる。厳密には、試合停止中の時間いわゆるロスタイムを加えたものとなっているが、RoboCup シミュ

レーションリーグにおけるロスタイムはその場で解消されるため、試合ログの上では、必ずしも考慮する必要がない。以下では、試合が行なわれている 6000 ステップのみを対象とする。

RoboCup サッカーシミュレーションは、公開されている公式の Soccer Server で行なわれ、行動ログは、soccerserver-log として自動的に保存される。通常、その中には管理情報など行動情報以外のデータが含まれており、おおむね 10Mbyte ~ 30Mbyte というサイズとなっている。これは不要に大きい。

人間サイズにモデル化されたエージェントの物理的能力の制約から 0.1 メートル程度しか動くことができない。シミュレータ内部での位置情報は 16 ビットで表現されているが、実際には、0.1 刻みで 100m すなわち $\log_2(100/0.1) \approx 10$ bit 程度の解像度で十分と言える。

以上の仮定から、基本的なデータサイズを見積もると次のようになる。

$$size = (10 + 10) \times 23 \times 6000 = 345KB. \quad (1)$$

3.3 圧縮システムの概要

本論文で扱うサッカー試合のデータの主たる用途は、人間が観賞することにある。このため、ユーザの感性にとって違和感のない範囲での、非可逆な圧縮が可能な対象である。

一般に、このような情報源符号化の場合、サンプリング、非可逆圧縮(再量子化)、可逆圧縮(エントロピー圧縮)を順に行なう。

可逆圧縮はハフマン符号化や LZ アルゴリズムに代表される、lossless 圧縮である。すでに原理上改善の余地が無いほど優秀な手法が実用化されている。本研究では圧縮システムのうち非可逆な圧縮のみを対象とし、可逆圧縮には一貫して gzip を用いることとする。

以降の圧縮データ量は、提案した手法で作成したテキストデータを、gzip で圧縮したものである。

まず対象とするサッカーサーバログは、2002 年 4 月に行なわれたジャパンオープンでの、OZ チーム対 YAMAARASHI チームの試合である。プレイヤーとボールの位置情報を 0.1 刻みで量子化した数値をテキストデータとしたときの

サイズは 1,467 KB となる。これを、gzip により圧縮を行なうと、361 KB となる。これは先の式 (1) での見積もりとほぼ一致している。

3.4 圧縮法 1 : 1 次線形予測

線形予測モデルを用いて残差情報のみ圧縮する方法であり、音声の圧縮等に用いられる。ここでは予測関数を $f(t) = 1.0 \times P(t-1)$ で表される 1 次線形関数とした。すなわち予測誤差ベクトルは前時刻との差と等しくなり、 $E(t) = P(t) - P(t-1)$ となる。この $E(t)$ を後段の可逆圧縮の対象とする。

なお、1 次線形予測モデルの自己回帰モデルを作成すると、係数はどの場合でもほぼ 1.0 となり、このモデルの有効性を示している。

圧縮結果は 205Kbyte(86.0%) である。これは量子化を行なわない可逆圧縮である。

3.5 圧縮法 2、座標軸量子化刻みの変更

刻み幅 を 0.1 から 1 に変更する。x 方向 100m は 7bit, y 方向 64m は 6bit に削減される。

圧縮結果は 134Kbyte(90.8%) で、再現のようすを図 1 に示す。

3.6 圧縮法 3、ベクトル量子化

$(P_0(t) \cdots P_{22}(t))$ を、ベクトル量子化してコーディングした。コードブックの生成には、自己組織化マップによる学習を用いた。コードブックとして 16x16 の 2 次元マップを用いた。

圧縮結果は 57.5Kbyte(96.1%) で、図を 2 に示す。

3.7 圧縮法 4、1 次予測 + ベクトル量子化

1 次予測残差、 $(E_0(t) \cdots E_{45}(t))$ について、方法 3 同様にベクトル量子化を行なった。

この際、歪みのある \hat{E} を積算することによる誤差累積が無視できない。これを排除するため、各時点の残差を源情報と伸長予測値の残差

表 1: 手法の比較

手法	サイズ	圧縮率	備考
オリジナルデータ	1467 KB	0 %	可逆圧縮 平均 0.5m の誤差 46 次元の状態を直接コーディング リーブデータがある
gzip のみ	361 KB	85 %	
1 次線形予測	205 KB	86 %	
再量子化	134 KB	91 %	
ベクトル量子化	58 KB	96 %	
線形予測 + ベクトル量子化	50 KB	97 %	

を量子化したものである。次の式で計算する。

$$\hat{E} = CodeBook(P(t) - \hat{P}(t-1))$$

ただし、伸長予測値は次の式で与える。

$$\hat{P}(t) = \hat{P}(t-1) + \hat{E}(t)$$

ここでの残差ベクトルの分布は、コードブックを作成したときの残差ベクトルの分布と多少異なっており、最適性には疑問が残るものの、現実的に大きな問題はない。

この方式でのサイズは 49.5Kbytes(96.6 %) で、図 3 に示す。

4 性能評価

各提案手法の圧縮サイズを表 1 にまとめる。

線形 1 次予測モデルを用いた圧縮は、可逆圧縮でありながら、比較的高いパフォーマンスをあげている。gzip が実質的には統計的圧縮を非常に高い精度で行なっているものの、時刻 $t-1$ との差相関は予測モデルを使わなければ対応できないことを示している。この時点で 200KB 程度であるが、試合に対するコメントが 10 ~ 50KB であることと比較すると、まだ大きすぎると言える。

刻み幅を単純に大きくした再量子化は、アルゴリズムが単純であり、最大誤差がたかだか刻み幅で押えられ、また削減量が容易に見積もれるという利点がある。いっぽうで量子化刻みが、逆にそのまま平均誤差に現れ、今回の観賞用データの再現という目的にはやや誤差が大きすぎるのが問題である。

ベクトル量子化 (VQ) は、非可逆圧縮の手法としては古典的であるものの、計算時間がかかるという難点がある。通常は、ベクトルの要素数が、2 ~ 5 程度でしか用いられていない。

本研究で提案した、自己組織化マップを用いて、圧縮用コードブックを作成する手法は、46 次元という多次元ベクトルを効率的に扱える方法である。また、その圧縮パフォーマンスも 96 % と非常に高いものとなっている。しかしながら、図 2 に見られるように、コードブックに登録された参照ベクトルと遠い孤立点が存在し、その周辺で極端に誤差が発生することが問題である。コードブックのエントリー数を増やすことで、ある程度対応できるものの、圧縮率が低下し、均衡点を探さなければならないこととなる。

1 次線形予測誤差に対して、VQ を適用した、LP+VQ 方式は、全域で誤差が少なく、平均誤差も小さくなっている。一方で、試合進行に起因する、位置の大きな変動のある部位、すなわち系統誤差により速度異常 j に大きい部位では、線形予測モデルの特性から、大きな逸脱が発生している。

しかしながら、圧縮率、精度のバランスも良く、今回対象としている、サッカー様マルチエージェントゲームの試合ログの圧縮に有効であると言えよう。逸脱点は、元来のデータがシステムのリープしているものなので、ここで圧縮データをセグメント分けし例外処理することで、圧縮比率を極端に下げること無く対応が可能である。

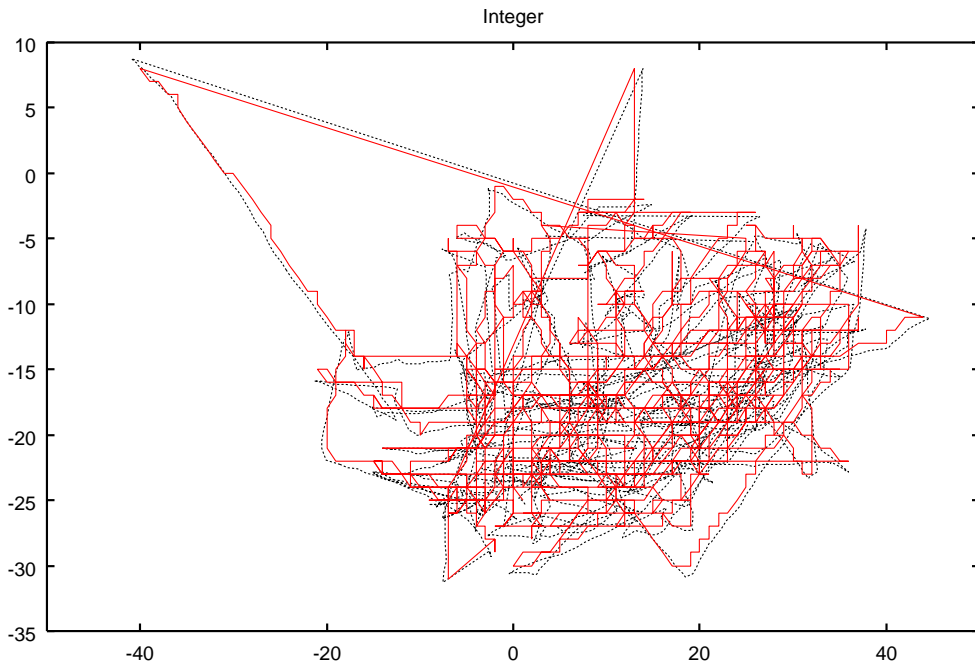


図 1: 量子化刻み=1.0

5 まとめ

従来は、音響や画像情報にのみ用いられてきた非可逆圧縮手法を、人間の観賞を対象とした、サッカーの試合記録の情報圧縮に適用した。複数の情報圧縮手法を比較し、1次線形予測とベクトル量子化の両方を利用した手法が、精度的にも、圧縮量も満足のいくものであった。

今回のベクトル量子化は、サッカーの試合におけるフィールド全体での定型的な動きの冗長度を捉えていると言える。特に、1次線形予測を用いたモデルでの、ベクトル量子化が高圧縮率であったのは、全体での並行移動が多々みられたからにほかならない。

サッカーに限らず、スポーツにおける運動の記録は、現在までのところ主としてビデオ情報にのみ依存している。計算機的に扱いやすく、かつ小さなサイズでの情報保存方法は、スポーツ工学、医学、人間工学や、娯楽の領域で必要とされる技術である。

今回は時間方向については、1次線形予測のみの対応であった。チームワークの存在を考えると、時間とプレイヤーの両軸に広がったより高次元ベクトルでの情報冗長度が存在すると考え

られる。これらのチームワークの存在に基づいて冗長度を削減する保存方法についての拡張が今後の課題である。

参考文献

- [1] A. Birk, S. Coradeschi, and S. Tadokoro, editors. *Robocup 2001 : Robot Soccer World Cup V*, volume 2377 of *LNAI*. Springer, 2002.
- [2] P. Stone, T. Balch, and G. Kraetzschmar, editors. *Robocup 2000 : Robot Soccer World Cup IV*, volume 2019 of *LNAI*. Springer, 2001.
- [3] 情報理論とその応用学会, editor. 情報源符号化 無歪みデータ圧縮. 培風館, 1998.
- [4] 情報理論とその応用学会, editor. 情報源符号化 歪みのあるデータ圧縮. 培風館, 2000.
- [5] 野田五十樹. ロボカップシミュレーションリーグ. In 第7回ゲームプログラミングワークショップ講演論文集, pages 22–27. 情報処理学会, 2002.

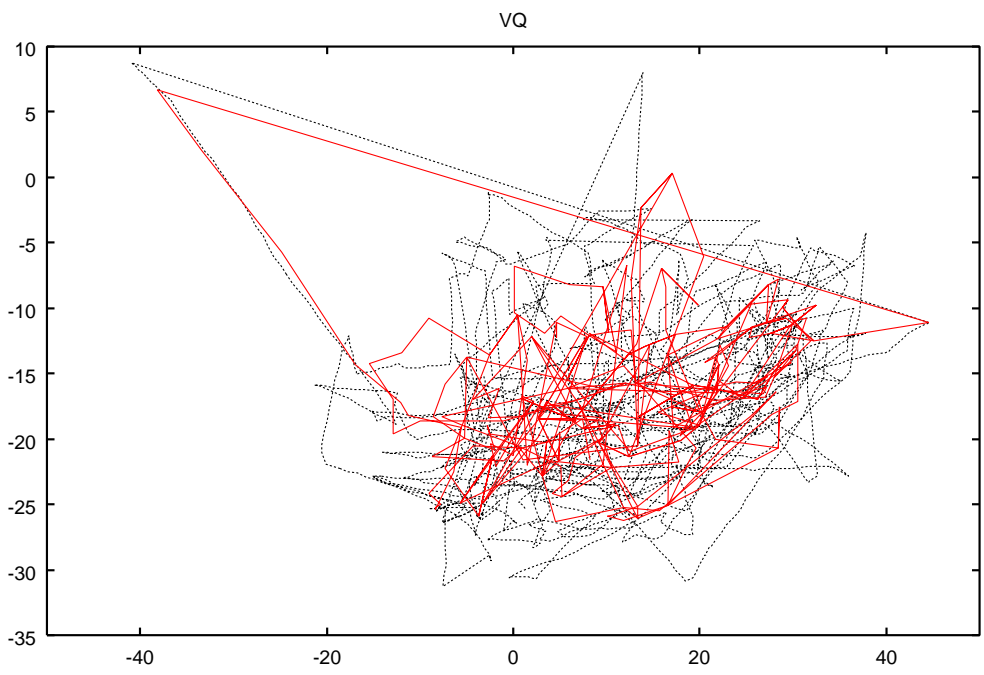


図 2: ベクトル量子化 (46 次元)

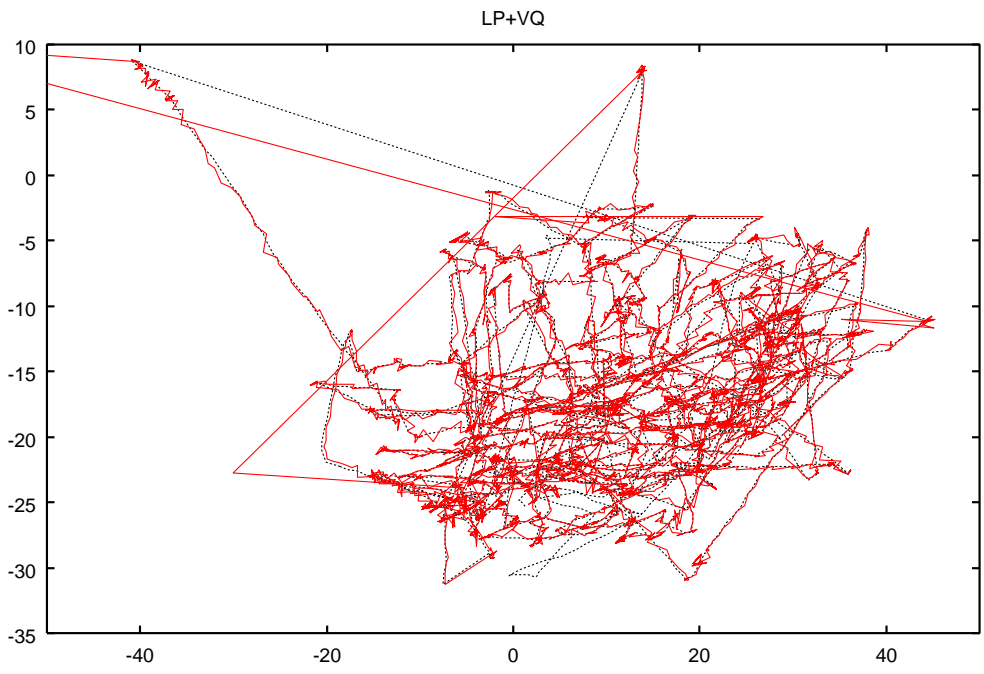


図 3: 予測 + ベクトル量子化