「マルチメディア，分散，協調とモバイル
(DICOMO2014) シンポジウム」 平成26年7月

レーザレンジスキャナーに特化したデータ圧縮手法

孫 為華[1] 柴田 直樹[2] 藤田 和久[3] 廣森 聡仁[3] 山口 弘純[3] 下條 真司[3]

概要：近年，人々の位置情報をセンサで推測・取得してシステムやサービスに活用する技術が注目されている．レーザレンジスキャナー (LRS) による人の計測は極めて高精度である半面，出力される計測ログのサイズが大きい．本稿では，LRS データに特化したデータ圧縮手法を提案し，(1) データ構造をより緊密にさせる，(2) 意味のないデータの除外，という二つのアイディアを用いて，著しくデータサイズを軽減させる．LRS 計測結果ファイルを用いた圧縮実験において，可逆圧縮方式では元データの 1/4 まで圧縮でき，不可逆方式では最大で元データの 3%まで圧縮できたことを確認した．また，HDD/SSD など異なる環境において，既存汎用圧縮手法と圧縮率，圧縮・解凍時間を比較し，両方において提案手法の優位性を実証した．

A Data Compression Method for Laser Range Scanner

Weihua Sun[1] Naoki Shibata[2] Kazuhisa Fujita[3] Akihito Hiromori[3] Hirozumi Yamaguchi[3] Shinji Shimojo[3]

1. はじめに

近年，人々の行動を多様なセンサで推測・取得してシステムやサービスに活用するためのヒューマンセンシング技術が注目されている．特にオフィスや商業施設，イベントスペースなど一定の閉空間を歩き回る人々の正確な位置情報は多くのサービスで期待されており，人々の移動履歴情報に基づくマーケティング調査，スマートホームやスマートビルディングといったエネルギー管理システム，商業施設や博物館・美術館などにおけるパーソナルナビゲーションなどが考えられる．このようなサービスを成立させるには，正確な端末の位置情報を取得できることが必要である．屋外においては，GPS が主な位置情報を取得する手段として実用化されているが，地下街，モールや映画館など室内における位置情報取得手段は，受信電波強度，赤外線，超音波，画像認識など多様な技術が研究開発されている．

一方，高精度トラッキングデバイスであり，プライバシー保護の観点からも優れるレーザレンジスキャナー（Laser Range Scanner, LRS）の活用が期待されてきており，我々の研究グループでも，文部科学省国家課題対応型研究開発推進事業－次世代 IT 基盤構築のための研究開発－「社会システム・サービスの最適化のための IT 統合システムの構築」の一環として，レーザレンジスキャナーを用いた屋内空間の人物トラッキングシステム（図 1）を大阪駅前の大規模商業施設「グランフロント大阪」内に展開し，1 年以上にわたる実証実験を継続している．

LRS は物体までの距離を正確に取得することができるセンサで，広範囲 (例えば市販されているスキャナーの走査範囲は距離 30m 程度，視野角 270 度程度の扇形領域) を一定の高さでスキャンすることが可能である．センサの計測データはセンサに対する方向と距離で表される計測対象の位置情報のみであるため，計測対象となる歩行者のプライバシを侵害するおそれが少なく，抵抗感も少ない．また，個々の走査データサイズは映像等と比べると小さくなるため計算コストも比較的小さい．一方， LRS センサを利用して高い時間密度で計測する際，出力される計測ログを蓄積するとサイズは非常に大きくなる．例えば，400 平方メートルの空間を対象に 4 台の LRS センサで計測する場合，1 時間の計測ログは 5GB を超えるなど，データの

[1] 滋賀大学
Shiga University, Banba, Hikone 522–8522, Japan
[2] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Takayama, Ikoma 630–0192, Japan
[3] 大阪大学
Osaka University, Yamada, Suida 565-0871, Japan

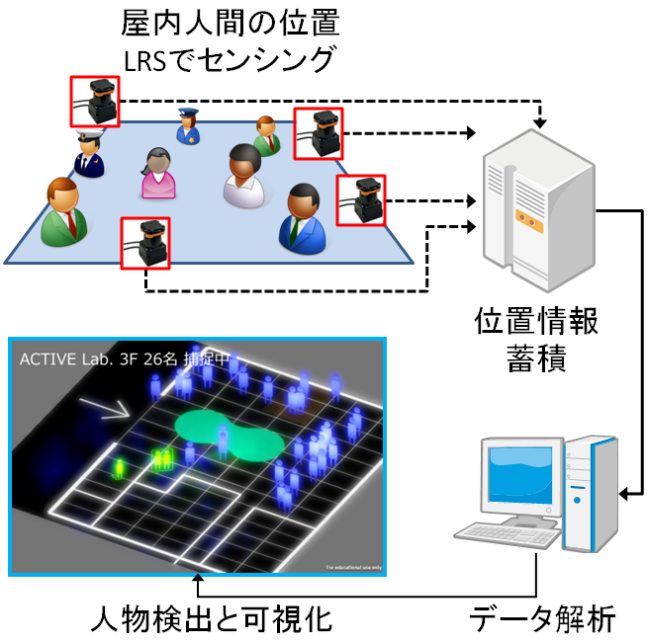


図 1 実証実験を行われている室内位置センシングシステム

保存は大きな問題である．既存の汎用圧縮技術を利用しても，元データの 40% 程度まで圧縮可能であるが，復元後のデータサイズを考えるとその取扱いは容易でなく，根本的な解決とは言えない．

本稿では，LRS データに特化したデータ圧縮手法を提案し，主に二つのアイディアを用いる．**（1）データ構造をより緊密にさせる．**LRS データの性質に着目し，計測データ間の差分をとり符号化することによる，可逆な圧縮方式を提案する．LRS のスキャン間隔と比較して，人間の動きは緩慢であるため，時間軸で観測する場合，データの変化量が小さい．そこで，数値の時間差分をとることにより，元データのサイズが縮小され，より緊密なデータ構造となる．**（2）無意味なデータの除外．**測定対象の歩行者でないと思われるデータを検出し除外すれば，保存対象となるデータの数そのものが減少する．計測結果の中に，壁，柱など，計測目的以外の物が多数存在する場合があり，これらを取り除くことにより著しくデータサイズを軽減できる可能性がある．

提案手法の性能を評価するために，我々は LRS 計測結果ファイルを用いて実験を行い，（1）圧縮率，（2）圧縮・解凍処理時間，（3）復元率の三指標で提案手法の性能を検証した．その結果，可逆圧縮方式では元データの 1/4 まで圧縮でき，不可逆方式では最大で元データの 3%まで圧縮できたことを確認した．また，HDD/SSD など異なる環境において，既存汎用圧縮手法と圧縮率，圧縮・解凍時間を比較し，両方において提案手法の優位性を実証した．不可逆圧縮後のファイルを解凍し，歩行者軌跡解析ツールを用いて解析した結果，測定対象の歩行者のデータが圧縮前後で一致したことも確認できた．本稿は 2 章で位置推定とデータ圧縮に関する分野研究を紹介し，3 章で LRS を紹介し，4 章で提案手法の詳細を示し，5 章で評価実験と考察を述べる．

2. 関連研究及び位置づけ

本節では位置推定手法，一般的なデータ圧縮手法を説明する上，提案手法の位置づけを説明する．

2.1 位置推定手法

一般的に用いられているモバイルデバイスの測位技術に GPS（Global Positioning System）[1], [2] があるが，地下や屋内では衛星からの電波を受信することができず正しい測位ができないうえに，見通しの良い屋外であっても数メートルから数十メートルの誤差を発生する．主に航空機，フェリー，自動車等に利用されるが，屋内には利用できない．IMES (Indoor MEssaging System)[3] などの疑似 GPS 機器などの開発も進んでいるものの，10m 間隔でセンサーを設置する必要があり，コストの課題もあり容易ではない．

モバイルデバイスに搭載されたコンパスやジャイロセンサを用いる自律航法（Dead Reckoning）[4], [5] 方式は車載 GPS と併用されることが多い．車両の距離計や，モバイルデバイスに搭載されている加速度センサーや，ジャイロセンサーから移動方向と移動量を推定する．自律航法のメリットは，GPS の誤差を軽減し，トンネルなどに入って GPS 信号受信ができなくなった場合でも軌跡を推定できることである．しかし，GPS が完全に利用できない屋内では，参照できる絶対位置情報がなく，誤差が累積する問題がある．

また，カメラを利用した人物認識や位置推定も多く研究されている [6], [7]．カメラレジストレーションや画像認識を用いた測位システムとしては，撮影された画像に撮影地点の情報を合わせて記録し，入力画像に対して一般画像認識を行う取り組みが提案されている．最近では携帯電話程度の情報端末でも実現できるようになっている．この方式の最大なメリットは，被写体となる歩行者はデバイスの装着・携帯をする必要がないことである．しかし，照明の暗い場所で撮影画像が不鮮明になること，データ解析に一定の計算能力を要すること，対象空間を全カバーするには数多くのカメラが必要であること，カメラの撮影距離など，問題点も多い．特に近年，プライバシー保護への意識が強くなり，顔が写されることに抵抗感が強い．

RFID（Radio Frequency IDentifier）を用いる位置推定手法 [8], [9] は，RFID タグを携帯した歩行者等がリーダ付近を通過した場合にタグが励起され，デバイスの位置とタグの ID を通知する．しかし，対象領域が広範囲に及ぶと，多数のリーダを設置する必要があり，コストの面で問題が残っている．さらに，RFID タグを搭載したデバイスを歩

行者が所持しなければならない．

複数のアンカーを設置し，歩行者の持つデバイスが発するビーコンの電力強度を測り，アンカーまでの距離や距離差を計測し，位置推定に用いる手法も一般的である．例えば，RADAR[10] は対象領域の受信電力マップを事前に構築し，測定された電力をマップと照合することによって位置推定を行う．また，マルチパスフェージングなどによる誤差を考慮し，過去の推定位置の履歴と再尤法を用いて軌跡を補正する手法 [11] も提案されている．受信電力に基づく手法はスマートフォンにも簡易に実装できる反面，高い測位精度を求める場合には歩行者が連続してビーコンを送信しなければならないこと，誤差は数メートルから十数メートルに達することなどが課題である．

2.2 データ圧縮手法

ハフマン符号 [12] は 1952 年に David A. Huffman によって開発された．コンパクト符号やエントロピー符号の一つであり，現在使われている圧縮ツールやフォーマットのほとんどに利用されている．

LZ77 [13] は，1977 年に Jacob Ziv と Abraham Lempel によって開発されたデータ圧縮アルゴリズムである．データを先頭から順番に符号化していき，現在注目している位置から始まる記号列が，それ以前に出現していたかを探す．発見した場合，記号列をその出現位置と長さのポインタに置き換える．記号列を探す範囲を Slide Window と呼び，これを辞書として使用するため，**辞書式圧縮法**と呼ばれる．派生バージョンである LZSS にハフマン符号化を適用した Deflate [14] アルゴリズムが LZH や ZIP などに採用され，現在も利用されている．

Welch は LZ78（LZ77 後続版）をさらに改良し，LZW アルゴリズム [14] を開発した．圧縮効率と引き換え，Deflate より 30%圧縮効率悪いが，高速化を実現している．GIF や TIFF など画像の圧縮で利用されている．

LZW 圧縮手法は特許の制限があったため，UNIX 上で Deflate アルゴリズムを用いた Gzip 手法 [16] が開発された．高い圧縮率を実現しており，標準入出力もサポートしているため，ファイル圧縮に限らず，多様な目的に使用されている．現在，フリーの PC-UNIX にはほぼ確実に搭載されている．

1996 年にオープンソースのデータ圧縮アルゴリズム Bzip2 [17] が開発された．Bzip2 は圧縮効率の高いブロックソート法，MTF 法，ハフマン符号法を用いており，gzip や ZIP より高い圧縮率を達成している．また，Bzip2 の操作法は意図的に Gzip に似せてあり，Gzip からの移行は容易である．しかし，Bzip2 は処理速度の点で Gzip よりも劣っており，Gzip を完全に置換するには至っていない．近年，Gzip は改良を重ねて，圧縮率において Bzip2 を超えた一方，Bzip2 はマルチスレッドに対応しているため，最近のマルチコア CPU 環境で処理速度が Gzip を追い越し，面白い展開を見せている．

7Z [18] は ZIP 形式に比べて高圧縮のファイルが作成できるほか，様々な圧縮アルゴリズムを組み合わせて使用できる．また，Unicode で全てのデータを格納しているため，異なる文字体系の名前のファイルを混在して圧縮・解凍することができる．

JPEG（Joint Photographic Experts Group）[19], [20] とは，コンピュータで扱われる静止画像の圧縮方式の一つであり，圧縮後，元のデータに戻れない不可逆圧縮方式である．JPEG では，画像を固定サイズのブロックに分割し，ブロック単位で離散コサイン変換を行い，空間領域から周波数領域へ変換する．変換されたデータは，量子化によって情報量が削減され，ハフマン符号によるエントロピー符号化がなされ圧縮が行われる．JPEG の方式は**空間型圧縮**と呼ばれる．MPEG-4 とは，動画・音声をデジタルデータとして扱うための規格であり，一般的に動画の符号化方式を指す．MPEG-4 では，フレーム間予測という技術 [21], [22] を用いて情報量の削減を行う．フレーム間予測では，空間型圧縮で作成した基準になるフレーム（I フレーム，Intra Picture）を設け，そのほか，I フレームとの差異を元にしたフレーム間の圧縮を採用している．すなわち，時間軸順方向の予測を行い，I フレームとの差分だけを残し，同様な部分を削ぎ落とす**時間型圧縮**である．本稿の提案方式ではこの概念を取り入れる．

2.3 提案手法の位置づけ

汎用データ圧縮手法は 60 年近く研究開発され，非常に洗練されている．しかし，汎用である故，LRS データの圧縮に用いる場合，LRS 生データ中に存在する意味のない順序構造と値を保持するために，圧縮の余地を残している．本稿では，LRS データに特化し，データの順序を再構成し，意味のない値を取り除く上，汎用手法を用いて圧縮することにより，さらに高い圧縮率を達成する．

3. レーザレンジスキャナー

レーザレンジスキャナーは赤外線レーザを用いて距離を測定する装置である．測定装置から被写体に向けて発振したレーザは物体表面で反射し，反射したレーザを測定装置が感知するまでに発振した回数から光のフライトタイムを推定し，距離を得る．比較的近距離の対象に対しては電波測距儀よりも顕著に高い精度での測定ができる．

3.1 スキャナー式レンジセンサ UTM-30LX

本稿では，北陽電機に製造されたスキャナー式レンジセンサ UTM-30LX（本稿でレーザレンジスキャナー/LRS と呼ぶ）について簡単に述べる [23]．製品のイメージ及び走査方法を図 2 に示す．この LRS は，レーザ光線により半円

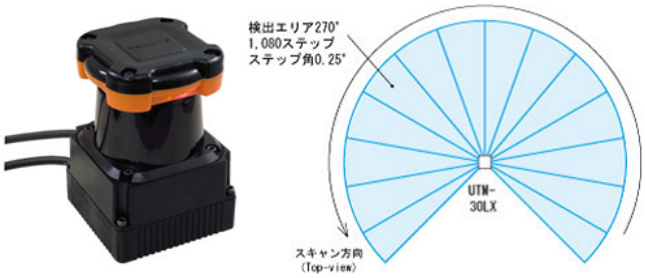


図 2　UTM-30LX センサと走査範囲

状のフィールドをスキャンし，角度ごとの距離データを出力する．スキャン角度は 270 度であり，ステップ角は 0.25 度である．検出距離は 0.1m から 30m まで (最大検出距離は 60m) であり，測距精度は対象物の距離が 10m 以下なら ±30mm，それ以上であれば ±50mm である．測定した距離は 1mm 単位の整数値で出力される．１回の走査にかかる時間は 25ms である．

3.2　LRS データの形式

表 1　25ms 秒ごとに出力される LRS レコード

ID: 8 bit	Time: 64 bit	1080 Results (1 result=16 bit)

上で述べたとおり，1 秒間に 40 回スキャンが行われ，1 回のスキャンあたり 1080 の距離データが出力される．測定した距離は 16 ビットの整数で表される．圧縮の対象とするデータは，複数 (4 台程度) の LRS から同時に出力されたデータであり，1 回のスキャン毎に，スキャンが行われた UNIX 時刻 (64 ビット) と，LRS のデバイス ID(8 ビット) が追加される．これらのデータは，MessagePack 形式 [24] でエンコードされており，本稿において圧縮率について述べるときは，MessagePack 形式でエンコードされたデータとのサイズ比について述べる．

4.　提案手法

この章では，LRS データに特化したデータ圧縮手法について述べる．本稿では，圧縮の対象として，複数の北陽電機のスキャナー式レンジセンサ UTM-30LX により与えられた時間帯に得られる全データを仮定する．

本稿では，可逆圧縮手法と不可逆圧縮手法の両方を提案する．

4.1　概要

LRS データの圧縮手法を開発するに当たり，下記の点をいずれも達成できることを目標とした．

高い圧縮率を達成できること

LRS は非常に多くのデータを出力する．上記のように MessagePack 形式でエンコードされた 4 台の LRS の出力データは，1 時間あたり 1.8G バイトになる．これを，gzip により圧縮しても 1G バイト程度にしかならない．ストレージのコストを削減し，データを扱うための手間を減らすため，本研究ではさらなる圧縮率の改善を目指す．

圧縮・展開が高速であること

上で述べたように，大きなデータを扱うためには，処理時間が必要になる．提案手法は，高速に圧縮・展開ができるようにするため，データ構造を工夫し，マルチスレッドで処理できるようにする．

シークできること

LRS データおよびそれを圧縮したデータはサイズが大きくなり，その中から目的のデータを探し出すために最初から展開する必要があるのであれば，そのために非常に時間がかかってしまう．提案する圧縮方式では，圧縮したデータを，指定された時間からすばやく展開できるような形式とする．

4.2　可逆圧縮アルゴリズム

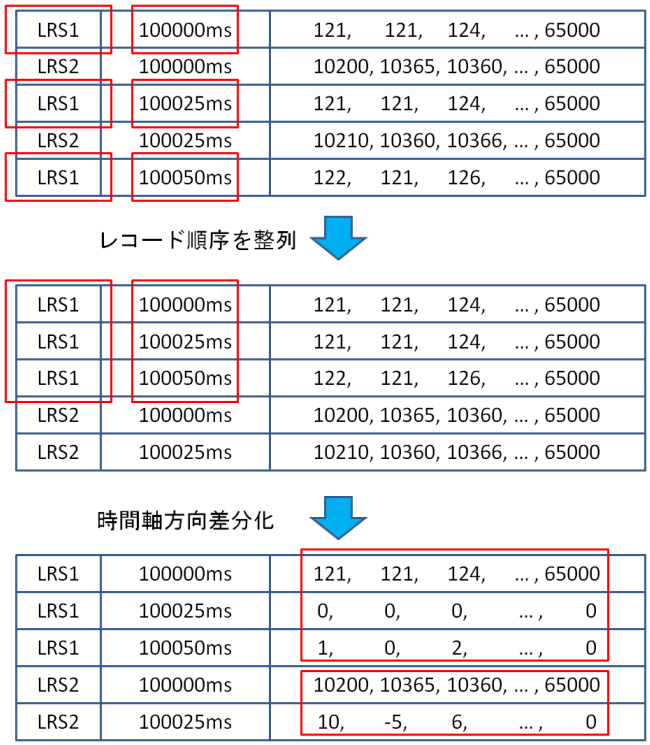


図 3　LRS レコードを整列・差分化

上記の目的を達成するため，次のような手順を用いて圧縮処理を行う．図 3 で手法の概要を示す．まず，一定時間 (2 秒) 分のデータ毎にシークできるようにし，また並列に圧縮・展開処理が行えるようにするため，入力データからこの時間分のデータを切り出す．次に，切り出したデータを LRS の ID 毎に分ける．LRS の同じ角度の距離データは，連続するスキャンの間で相関が高いため，同じ LRS で計測した同じ角度の距離データが並ぶように，データの並べ替えを行う．次に，連続する距離データ間で差分をとり，

その結果を Golomb Coding によりエンコードする．この処理により，データの構造がより緊密になり，サイズが軽減される．この結果を，既存の圧縮方式 [25] を利用して圧縮を行う．さらに，シークのためのワードを付加した上，順に出力する．

4.3 不可逆圧縮アルゴリズム

LRS による計測結果には誤差が含まれており，誤差の部分を丸めることで圧縮率を向上させることが原理的に可能である．その一方で，誤差の部分にも，測定データを活用する上で意味のある情報が含まれている可能性があり，単純にこの部分の情報を消去すると，後々の利用時において不都合が出る可能性がある．本稿では，LRS の周囲にある背景の構造物と考えられる部分に限り，誤差による距離データのランダムな変化を削除することでさらに圧縮率を向上させる．本稿では，LRS を固定して使用し，測定データを，背景の構造物よりも手前を通過する人物などの位置推定に主に使用すると考えている．LRS の周りにあまり人などがいないケースでは，この工夫により，もとのデータの 3%程度にまで圧縮が可能である．

提案手法では，まず入力ファイルをスキャンし，LRS の ID・角度毎に測定された距離の最大値（測定範囲外を示す値は無視する）を求める．次に，入力ファイルを，上述の可逆圧縮アルゴリズムで圧縮する際に，測定された距離の最大値から，LRS の誤差範囲内にある値を全て測定された距離の最大値で上書きする．

4.4 マルチスレッド化

4.2 節で述べたように，提案する圧縮形式は並列に圧縮・展開処理が行えるように，一定時間 (2 秒) 分のデータ・LRS の ID 毎に分割されている．提案方式では，圧縮処理全体を読み出しスレッド・圧縮スレッド（複数）・書き込みスレッドに分けて実行する．圧縮スレッドは同時実行可能なハードウェアスレッドの数と同じ数用意する．読み出しスレッドは，入力データを一定時間 (2 秒) 分読み出して，LRS の ID 毎に分割し，圧縮スレッドに送る．圧縮スレッドでは，データの並び替え・Golomb Coding の適用・bzip2 による圧縮処理を行う．書き込みスレッドは，圧縮スレッドの結果を受け取り，順に並べて書き出す．ディスクの読み書きによる待ち時間は全て読み出しスレッド・書き込みスレッドで消費されるようになっており，ディスクアクセスがボトルネックにならない限り，圧縮スレッドは待ち時間なしに動作する．

上記に加え，MessagePack 形式のエンコード・デコード処理に時間がかかるため，提案手法では独自の実装を用いてこの部分を高速化している．

5. 性能評価

本節で LRS センシングデータを圧縮する際，既存汎用手法，提案手法可逆圧縮アルゴリズム及び不可逆アルゴリズムを適用する際の圧縮率，処理時間を示す．また，データに欠損を与える不可逆アルゴリズムを適用した後，計測対象データの復元率を示す．

5.1 実験環境及び評価項目

圧縮実験を行った計算機の仕様を表 2 で示す．圧縮プログラムを Java で作成したため，異なる計算機，またはオペレーティングシステム環境でも利用可能である．計算機の CPU i7-920 に 4 コア 8 スレッドがあり，これをマルチスレッド対応可能な Bzip2 に利用した．

表 2 ハードディスクを利用した実験環境

項目	仕様
CPU	Intel Core i7-920@2.67GHz
実装メモリ	18GB
HDD	WDC WD10EADS-M2B
Sequential Read	97.3 MB/Sec
Sequential Write	76.2 MB/Sec
Random Read (512)	34.5 MB/Sec
Random Write (512)	63.9 MB/Sec
OS	Windows 7 Professional SP1 x64
Java SE	1.8.0_05 x64
VM Reserved RAM	2GB

実験の圧縮対象である LRS のセンシングデータは，グランフロント大阪ナレッジキャピタル北館 1–3F にある The LAB（国内外の企業，研究機関，大学が参画する合同イベントラボ）において取得したものである．大阪大学の研究チームは場内各フロアに LRS センサー 4 台を設置し，2013 年 4 月下旬から現在まで常時に来場者の位置を取得し，データを保存している．

評価項目

- **圧縮率** 圧縮前のファイルサイズを圧縮後のファイルサイズで割った値である．この値が小さいほど圧縮後のファイルサイズが小さく，圧縮の効果が高い．
- **処理時間** 圧縮や解凍の作業過程にかかる時間である．処理時間が短いほど良い．
- **復元率** 不可逆アルゴリズムだけに対する評価項目である．圧縮後ファイルを解凍し，計測対象データの復元率が高いほど，データの損傷が小さい．

予備実験：汎用ソフトの圧縮結果

汎用圧縮ソフトで圧縮の予備実験を行った．表 3 はその結果である．使用したオリジナルファイルは 2013 年 4 月 30 日の 11 時〜12 時のデータである．当時は開館したばかり，かつゴールデンウィーク中だったため，来場者が殺到

し，非常に密度の高いデータが取得できた．

表 3 Bzip2 と WinRAR での圧縮結果

Original File High Density **Date** 2013/04/30 11-12	**Bzip2 V.1.0.5** Compressibility Time Size	**WinRAR V.5.0.1** Compressibility Time Size
Text Format	22.79%	67.89%
	$\geq$ 15 Min	$\approx$ 30 Sec
2,142,435,920 Byte	488,356,265 Byte	1,454,476,101 Byte
Binary Format	31.58%	54.38%
	$\geq$ 10 Min	$\approx$ 30 Sec
1,297,833,468 Byte	409,828,446 Byte	705,820,246 Byte

予備実験では，2 種類のデータフォーマットを試した．データ構造が比較的緩いテキストファイルと構造が密なバイナリファイルである．テキストフォーマットは LRS データ利用者にとって操作，確認しやすいが，サイズが大きく，通常，センシングデータは自動的にサイズの小さいバイナリファイルに保存される．予備実験では両方のフォーマットを用いて，汎用手法のデータ構造緊密化効果を確認する．汎用圧縮手法として，センシングデータの圧縮保存で使用される Bzip2 Ver.1.0.5 及び Windows で環境でよく使われる WinRAR Ver.5.0.1 を用いた．

Bzip2 で圧縮した結果，2GB 以上のテキストファイルを 15 分以上，1.2GB のバイナリファイルを 10 分以上の時間がかかり，それぞれ約 488MB（22.79%），約 409MB（31.58%）に圧縮した．Bzip2 は圧縮率高いことが特徴であり，フォーマットもサイズも異なる二つのファイルを近いサイズまで圧縮できた．半面，長い時間がかかった原因は，マルチスレッド機能の未設定とデフォルトの使用メモリが小さいことと思われる．提案手法の中で，Bzip2 をソースコードレベルで利用しており，後述実験の中で十分に早い圧縮速度を確認できている．

一方，WinRAR は非常に速い速度で圧縮を完了した．それぞれのファイルを約 30 秒で圧縮できたが，圧縮率は低い．2GB のテキストファイルを約 1G450MB（67.89%），1.2GB のバイナリファイルを約 700MB（54.38%）にしか圧縮てきなかった．

このような結果からも，LRS センシングデータに特化した圧縮プログラムの必要性が伺える．

5.2 提案手法での圧縮実験

提案手法の実験に利用したファイルを四つのカテゴリから計 20 個選別した．全部バイナリファイルで，サイズは 1.2GB（Dense，Night）と 1.8GB（Normal，Sparse）である．測定対象エリアの構造物などの変化でサイズの違いが発生したが，センシングデータの構造は同じである．

- **高密度/Dense** 来場者が殺到する状況である．計測対象のデータが非常に多く，不可逆圧縮でデータの削ぎ落としが困難である．開館当時の 2013 年 4 月 30 日（GW 中）11 時–16 時に取得したデータを用いる．
- **平常時/Normal** 平日の昼間に客が来場する状況である．壁など計測対象でないデータの占める割合が高くなる．2013 年 7 月 1 日（月曜日）11 時–16 時に取得したデータを用いる．
- **閑散時/Sparse** 平日の夜，閉館直前にほとんど客がいない状況である．場内に計測対象でないデータの割合が非常に高い．2013 年 7 月 1 日–5 日平日五日間の夜 21 時–22 時に取得したデータを用いる．
- **深夜/Night** 閉店後スタッフが館内で翌日の準備をする状況である．計測対象でないデータの割合が高い．2013 年 4 月 30 日午前 1 時–2 時に取得したデータを用いる．

圧縮実験結果を図 4 に示す．提案メソッドの可逆圧縮（Lossless）または不可逆圧縮（Lossy）で処理した後，ソースコードレベルで利用可能な汎用圧縮手法 Bzip2 と Gzip で圧縮を行った．ただし，Bzip2 はマルチスレッドを利用可能なため，8 スレッドで圧縮を行う．一方，Gzip はマルチスレッド非対応のため，シングルスレッドのみで圧縮を行う．

可逆圧縮/Bzip2 来場者数高密度の場合，平均圧縮率は約 27%程度，表 3 の Bzip2 単独圧縮の 31.58%と比べて，差はわずか 15%程度である．他の密度の場合，平均圧縮率は 24%前後で，Bzip2 単独より約 3 割弱の効果が出ている．トップレベルの圧縮率を有する Bzip2 と比べて，可逆圧縮アルゴリズムで行っている差分化処理の効果は限られていると言える．一方，Windows で常用されている WinRAR より，半分以下に圧縮できる効果が読み取れる．また，マルチスレッドで稼働するため，圧縮時間が明らかに短い．1.2GB–1.8GB のデータを 50 秒で圧縮することは，半分以上の時間はハードディスク入出力に費やしているため，処理時間自体は十分に短いと言える．

可逆圧縮/Gzip データの圧縮率は 23%–27%弱で，Bzip2 より少しだけ圧縮率が高い．可逆圧縮で軽減したデータサイズは Bzip2 と同じであるため，この差は Gzip と Bzip2 の性能によるものである．一方，処理時間は 44 秒–64 秒で，Bzip2 の平均より長くなっている．Gzip はシングルスレッドでしか動作できず，8 スレッドの Bzip2 より利用可能なリソースが少なかったからである．シングルスレッドでこれだけの速度が達成できた点を考えれば，Gzip の高速化メカニズムは高度なものである．

不可逆圧縮/Bzip2 来場者数高密度，平常時，閑散時，深夜の場合，圧縮率はそれぞれ 17.56%，9.11%，4.55%，6.40%である．これより，計測対象でないデータが大量に削ぎ落とされたことがわかる．特に閑散時と深夜時において，データは非常に小さく圧縮されたことが確認できた．

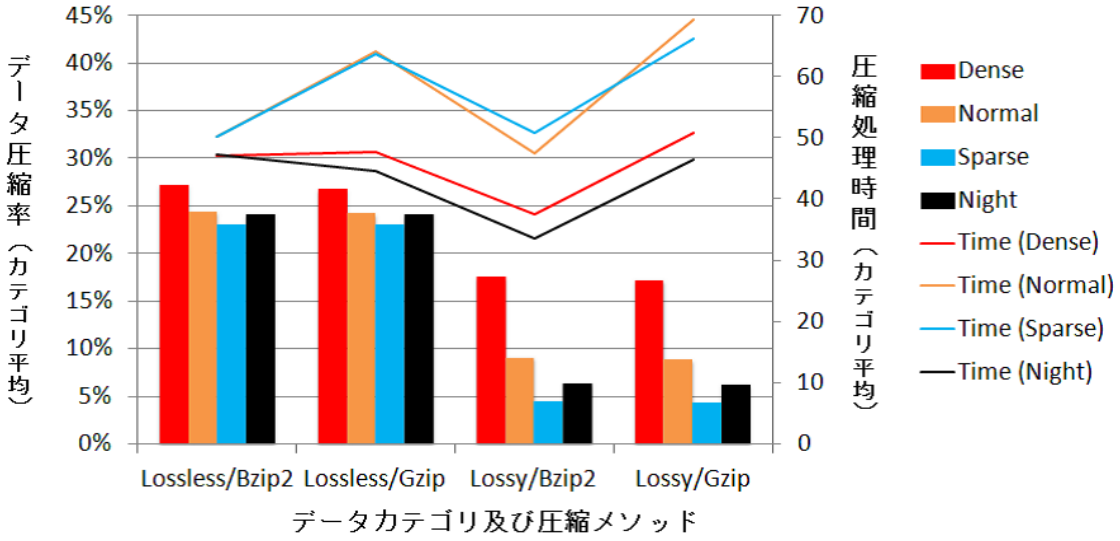


図 4　提案手法を用いた圧縮実験結果

処理時間も 33 秒–50 秒程度で，十分に短いと言える．

不可逆圧縮/Gzip Bzip2 と同様な傾向が見られ，高密度，平常時，閑散時，深夜の場合にそれぞれ 17.25%，9.11%，4.33%，6.30%の圧縮率を達成している．わずかながら Bzip2 より圧縮率が高くなる一方，処理時間は Bzip2 より長くなっている．

Normal 及び Sparse データの処理時間が Dense，Night より長い原因は，元ファイルのサイズの違いである．また，この実験で利用したストレージデバイスは通常のハードディスクであるため，ディスクの読み書き時間が大きなボトルネックとなっており，提案手法の処理時間の評価としては正確ではない．

5.3　SSD ディスクを用いた実験

ハードディスクアクセス時間は提案手法処理時間のボトルネックになるため，ディスクアクセス時間割合の低い SSD ディスクを利用する環境で小規模な実験も行った．用いたデータは 2013 年 5 月 15 日 15 時–16 時に取得したサイズ 1.8GB の高密度データである．実験環境の仕様を表 4，結果を表 5 に示している．

表 4　SSD ディスク実験環境

	Windows 環境
CPU	Intel Core i7-3770T@2.5GHz
HDD	SSD (Read ≥ 300MB/Sec)
OS	Windows 7 Enterprise x64
Java SE	1.8.0_05 x64
	Linux 環境
CPU	Intel Core i7-4770@3.4GHz
HDD	SSD (Read ≥ 300MB/Sec)
OS	Ubuntu Linux 3.11.0
Java SE	1.7.0_51 x64

Windows 環境の実験結果

高密度データに対し，提案手法の Lossless 方式，Lossy 方式いずれも 40 秒台で圧縮でき，10 秒台で解凍できている．ハードディスク環境での実験結果に対し，圧縮処理時間は 15–20 秒軽減された．

表 5　SSD ディスクを利用する環境での実験結果

	圧縮処理時間 **圧縮率**	**解凍処理時間**
Windows 環境		
7-zip LZMA2 8-threads	258 Sec 29.9%	49 Sec
7-zip Bzip2 8-threads	45 Sec 31.4%	31 Sec
Proposed Method Lossless/Bzip2	40 Sec 25.6%	13 Sec
Proposed Method Lossy/Bzip2	41 Sec 15.2%	14 Sec
Ubuntu 環境		
7z a	423 Sec 25.2%	26 Sec
pbzip2 -9	40 Sec 28.7%	13 Sec
gzip -9	78 Sec 52.3%	9 Sec
Proposed Method Lossless/Bzip2	30 Sec 22.5%	20 Sec
Proposed Method Lossy/Bzip2	23 Sec 4.11%	6 Sec

Ubuntu 環境

夜間データに対し，提案手法の Lossless 方式は圧縮に 30 秒，解凍に 20 秒を必要としたが，Lossy 方式は圧縮に 23 秒，解凍に 6 秒という速さで処理できた．計測対象でない

データを削ぎ落とした分，Lossy 方式の解凍は非常に高速になっている．ハードディスク&Windows 環境と比べて，圧縮時間は 20–30 秒ほど軽減されたことが分かる．

SSD ディスクを利用する環境において，ディスクアクセス時間が大幅に短縮されるため，提案手法の処理時間も明かに短くなる．しかし，現状では，大容量データの保存に SSD を利用することはコストの面から到底考えられないため，ハードディスクでの処理時間は現実応用の目安となる．

5.4 不可逆圧縮を行った後の復元率

不可逆圧縮はオリジナルデータを欠損させるため，解凍後，計測対象データの復元率を調べる必要がある．計測対象データを解析・判明するために，大阪大学チームに作成された LRS_Analyzer というプログラムを利用した．LRS_Analyzer は LRS センシングデータを入力とし，レコードを順次読み込み，位置の変化履歴から，リアルタイムにそれが構造物，あるいは被写体（ひと）かを判定し，結果を GUI でのビジュアル表現と解析ファイルの両方に出力する．深夜 1 時–2 時，及び高密度昼 11 時–12 時の解析例をそれぞれ図 5,6 に示した．

図中の緑色線分は被写体と思われるものの軌跡を示し，青色線分は家具や壁などの構造物，赤色文字は LRS_Analyzer に付けられたラベルである．提案手法はファイル内のレコード順序を並べ替えており，解凍時順序の構造まで回復しないため，オリジナルファイルと順序が異なる．その原因で LRS_Analyzer に付けられた赤色ラベルは少し異なるが，計測対象データである被写体の軌跡は同じである．LRS_Analyzer の解析ファイルと比較した結果，計測対象データの欠損が発生しておらず，100%の復元率を達成している．

提案手法は復元率を保証するために，明かに計測対象でないデータ（壁）以外，削ぎ落としていない．不可逆圧縮解凍後の被写体推定図に青色線分が多く残っていることが確認できる．これらも測定対象ではなく，削ぎ落としても測定データの正確性に影響を与えないが，非可逆圧縮アルゴリズムで断定しにくいため，処置を施さなかった．そのため，提案手法には False Negative の性質がある．

6. まとめ

本稿では，レーザレンジスキャナー LRS のセンシングデータに特化した圧縮手法を提案している．提案手法では，データの構造を緊密化させ，さらに計測対象でないデータを削ぎ落とすことにより，データサイズを削減する．

実データを用いた圧縮実験を通して，提案手法の圧縮率，処理速度の面における優位性が確認できた．また，不可逆圧縮を適用したデータを解凍し，元データと解析照合することで，計測対象データの完全なる復元性が確認できた．

一方，提案手法では，計測対象データの欠損を危惧し，不可逆圧縮アルゴリズムでは疑わしくても，計測対象でないと断定できないデータを残すなど，更なる圧縮の余地を残している．今後は検出精度を向上させ，さらに高い圧縮率を達成するよう，アルゴリズムを改良する予定である．

謝辞 本研究は文部科学省国家課題対応型研究開発推進事業 －次世代 IT 基盤構築のための研究開発－「社会システム・サービスの最適化のための IT 統合システムの構築」（2012 年度～2016 年度）の助成を受けたものです．

参考文献

[1] Misra, Pratap, and Per Enge, "Global Positioning System: Signals, Measurements and Performance Second Edition", Massachusetts: Ganga-Jamuna Press, 2006.

[2] Tomkiewicz, Stanley M., Mark R. Fuller, John G. Kie, and Kirk K. Bates, "Global positioning system and associated technologies in animal behaviour and ecological research", Philosophical Transactions of the Royal Society B: Biological Sciences 365, no. 1550, pp. 2163-2176, 2010.

[3] Fujimoto, Junya, S. Hotta, K. Sawada, Y. Hada, K. Hida, and Shinichiro Mori, "Hybrid positioning system combining spatially continuous and discrete information for indoor location-based service", In Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), pp. 1-6, 2012.

[4] Geier, George J., Ardalan Heshmati, Kelly G. Johnson, and Patricia W. McLain, "Position and velocity estimation system for adaptive weighting of GPS and dead-reckoning information", U.S. Patent 5,416,712, issued May 16, 1995.

[5] Asano, Satoshi, Yuki Wakuda, Noboru Koshizuka, and Ken Sakamura, "A robust pedestrian dead-reckoning positioning based on pedestrian behavior and sensor validity", In Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION, pp. 328-333. IEEE, 2012.

[6] Hachet, Martin, Fabrice Declec, Sebastian Knodel, and Pascal Guitton, "Navidget for easy 3d camera positioning from 2d inputs", In 3D User Interfaces 2008 (3DUI 2008) IEEE Symposium on, pp. 83-89. IEEE, 2008.

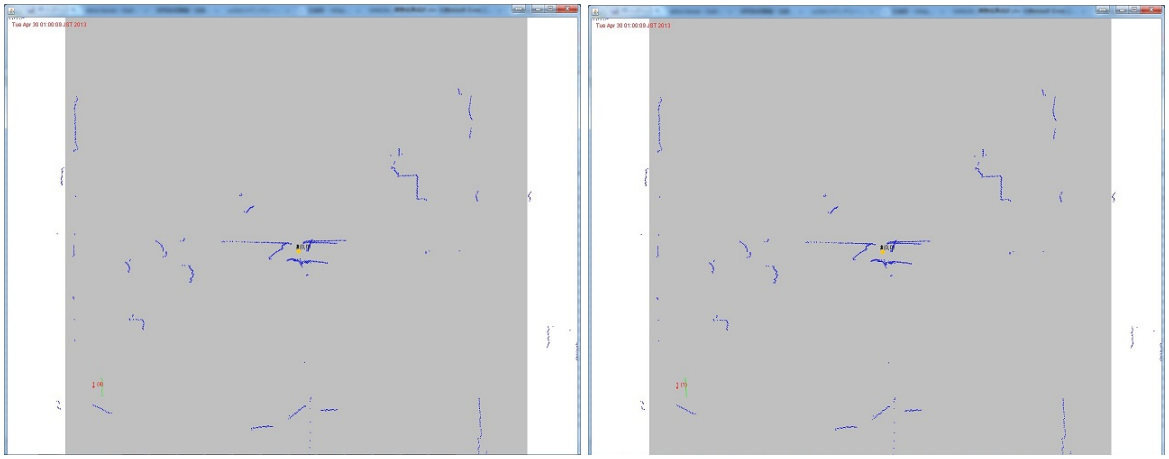
[7] Boyden, James H., Kory D. Christensen, and David W. Meibos, "Camera positioning system and method for eye-to-eye communication", U.S. Patent 20,030,112,325, issued June 19, 2003.

[8] 小田 英雄, 久保田 創一, 岡本 栄晴, "RFID を利用した安全運転支援システムにおける位置・進行方向・速度情報による安全情報伝達方法の検討", 情報処理学会研究報告, 2006(103), pp. 31–36, 2006.

[9] Koyuncu, Hakan, and Shuang Hua Yang, "A survey of indoor positioning and object locating systems", IJCSNS International Journal of Computer Science and Network Security 10, no. 5, pp. 121-128, 2010.

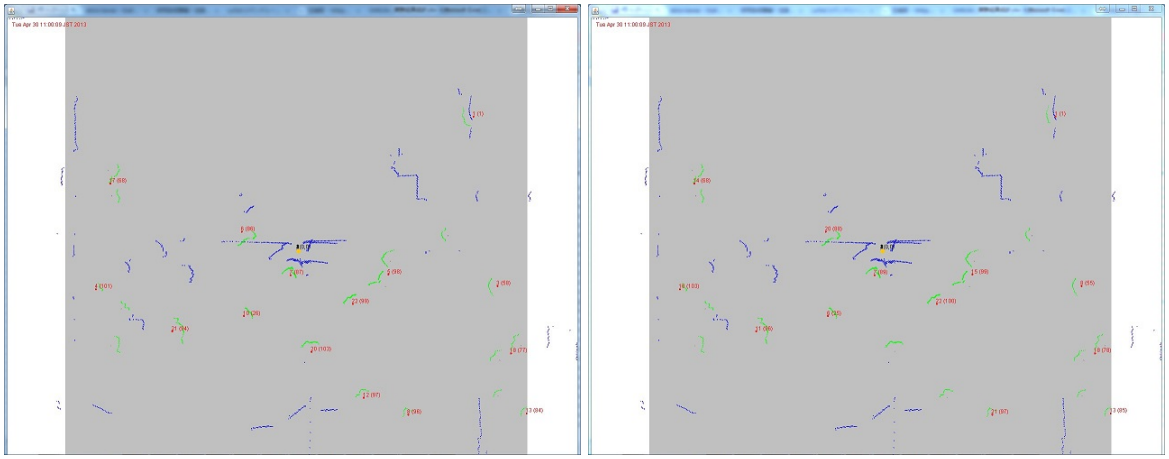
[10] Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system", In Proceedings of the International Conference on Computer Communications (INFOCOM 2000), pp. 775–784, 2000.

[11] Weihua Sun, Hirozumi Yamaguchi, Keiichi Yasumoto, and Minoru Ito, "Range-based Localization for Estimating Pedestrian Trajectory in Intersection with Roadside Anchors", in Proc. of the First IEEE Vehicular Networking Conference (VNC2009), in-DVD, Oct. 2009.



(a) 元ファイルの被写体推定図　　(b) 不可逆圧縮解凍後の被写体推定図

図 5　深夜 1 時–2 時の被写体推定（緑色は被写体軌跡）



(a) 元ファイルの被写体推定図　　(b) 不可逆圧縮解凍後の被写体推定図

図 6　高密度，昼 11 時–12 時の被写体推定（緑色は被写体軌跡）

[12] Wikipedia, Huffman Coding, `http://en.wikipedia.org/wiki/Huffman_coding`

[13] Wikipedia, LZ77 and LZ78, `http://en.wikipedia.org/wiki/LZ77_and_LZ78`

[14] Wikipedia, Deflate, `http://en.wikipedia.org/wiki/DEFLATE`

[15] Nelson, Mark R., "LZW data compression." Dr. Dobb's Journal 14, no. 10, pp. 29-36, 1989.

[16] Deutsch, L. Peter., "GZIP file format specification version 4.3", 1996.

[17] Bzip2, Bzip2 and libbzip2, `http://www.bzip.org/`

[18] Konecki, Mladen, Robert Kudelic, and Alen Lovrencic, "Efficiency of lossless data compression", In MIPRO, 2011 Proceedings of the 34th International Convention, pp. 810-815, IEEE, 2011.

[19] Wikipedia, JPEG, `http://ja.wikipedia.org/wiki/JPEG`

[20] Adams, Michael D., "The JPEG-2000 Still Image Compression Standard, ISO/IEC JTC 1/SC 29/WG 1N 2412", Dept. of Electrical and Computer Engineering, University of Victoria, Canada, December 2005.

[21] Wikipedia, Motion Compensation, `http://en.wikipedia.org/wiki/Motion_compensation`

[22] Vetro, Anthony, Thomas Wiegand, and Gary J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard", Proceedings of the IEEE 99, no. 4, pp. 626-642, 2011.

[23] 北洋電機 : "測域センサ Scanninglaser range sensor UTM-30LX/LN 仕様書"

[24] MessagePack : `http://msgpack.org/`

[25] Burrows, M., Wheeler, D, J. : "A block sorting lossless data compression algorithm," Technical Report 124, Digital Equipment Corporation, 1994.