

ライフログのためのセンサデータと 画像ストリーム処理ミドルウェアの構築

岩木紗恵子[†] 村瀬 勉^{††} 小口 正人[†]

[†] お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

^{††} NEC システムプラットフォーム研究所 〒211-8666 神奈川県川崎市中原区下沼部 1753

E-mail: [†]saeko@ogl.is.ocha.ac.jp, ^{††}t-murase@ap.jp.nec.com, ^{†††}oguchi@computer.org

あらまし 近年のデバイスやネットワーク技術の発展により、ライフログを収集することは比較的容易になった。しかし、蓄積されたライフログを有効に活用することが出来ていないのが現状である。ライフログを高度なアプリケーションに適用するためには、データを収集や蓄積する際の品質の管理が重要になる。そこで本研究では、アプリケーションが必要とするデバイスと品質を指定してライフログのデータを利用することができるミドルウェア環境の構築を行う。そしてライフログから情報抽出を行うアプリケーションの例として、現在開発が行われているセンサデータや画像ストリームなどの収集したデータから人間の行動を言語化するアプリケーションを取り上げ、本ミドルウェアに適用した結果を報告する。

キーワード ライフログ, センサネットワーク, 画像ストリーム, ミドルウェア

Implementation of Middleware supporting Sensor Data and Stream of Motion Picture for Lifelog

Saeko IWAKI[†], Tsutomu MURASE^{††}, and Masato OGUCHI[†]

[†] Ochanomizu University Otsuka 2-1-1, Bunkyo-Ku, Tokyo 112-8610 Japan

^{††} System Platforms Research Laboratories NEC Corporation

1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211-8666, Japan

E-mail: [†]saeko@ogl.is.ocha.ac.jp, ^{††}t-murase@ap.jp.nec.com, ^{†††}oguchi@computer.org

Abstract It becomes easier to collect lifelog, thanks to recent advancement of device and network technology. However, it is still difficult to take advantage of the accumulated lifelog. In order to apply it to advanced applications, it is important to management quality of data when the data is collected and stored. Thus, in this research, a middleware environment that can use the data of life log is going to be constructed when an application specifies devices or qualities that it requires. Moreover, as an example of the developed application, a research of verbalization of human's action from data of sensor network and stream of motion picture, is applied to middleware.

Key words Lifelog, Sensor Network, Graphical Stream, Middleware

1. はじめに

今日の情報量の多さから、ユーザにとって本当に必要な情報の取捨選択が困難となっている。また、ライフスタイルの多様化により個人の求める情報も多種多様となっている。そのような中で個々のユーザに適切な情報やアプリケーションを提供するためには、ライフログの取得が必要不可欠である。

1.1 ライフログ

ライフログとは、生活の記録をデジタルデータとして保存す

ることである。ライフログを取得し、蓄積、解析、加工することで、個人のニーズに合わせた、より便利なアプリケーションの開発が見込まれる [1]。

正確なライフログの収集とその活用のためには、大量のデータを取得が必要である。ライフログは、断片的な個別の記録の蓄積だけでは意味をなさないが、多くのデータを関連性を持って蓄積していくことで、意味のあるデータとして利用することができる。収集、蓄積されたライフログを必要に応じて取得し、解析、加工することでユーザに便利な情報やアプリケーションを提供できるようになる。

1.2 ライフログの活用例

ライフログの活用例を挙げる。

携帯に搭載されている加速度センサと、GPS 情報を用いることにより、移動中の状態（静止しているか、歩いているか、電車に乗っているかなど）を正確に把握できるようになるが、この情報を蓄積していくと、今いる場所の情報だけでなく、行動範囲を考慮した情報の提供が可能となる。

他に、ドライバの健康データを利用することで、休憩のタイミングや時間を自動で設定するドライブ中の体調管理システムの構築が可能となる。

このように、ライフログを活用することで、より生活が便利になると考えられる。

1.3 ライフログ活用に向けた課題

2009 年は、ライフログ元年と呼ばれるほどライフログに注目が集まった年である。

近年のネットワークやデバイス技術の進歩により、ライフログのような大量のデータ取得や蓄積は比較的容易になってきた。しかし、データを大量に貯められるようになった一方でそのデータを上手く使いきれていないのが現状である。そこで、大量に蓄積されたデータを有効に活用するシステムが必要であると考えられる。ライフログを活用するためには、最適な品質のデータを収集し、適切に管理する必要がある。

2. 研究内容

2.1 ミドルウェアの構築

本研究は、大量に取得、蓄積されたライフログの高度なデータ処理を行うアプリケーションのためのミドルウェアを構築することを目的とする。具体的には、アプリケーションが要求するデバイス（カメラ、センサなど）のデータをアプリケーションの要求する品質で返すシステムを構築する。このミドルウェアの概念図を図 1 に示す。

また、データ収集にネットワークを介する場合、アプリケーションが必要とするデータの品質とネットワークの品質の兼ね合いを考慮し、データの品質面、ネットワークの品質面の両方にとって最適なデータを収集し、アプリケーションに提供する。

さらに、ネットワーク混雑時などにおいても、アプリケーションの求めるデータ品質を保證することが可能なミドルウェアの構築を目標とする。

2.2 ミドルウェア

ミドルウェアとは、OS とアプリケーションの中間的な処理、動作を行うソフトウェアのことである。ミドルウェアが存在することにより、OS は必要最小限の機能で済み、アプリケーションはアプリケーションのことだけを考えればよくなるため、アプリケーションの開発が容易になる。また、ミドルウェアがシステム環境による違いを吸収するため、OS は基本的な動作を提供するだけで、アプリケーションは OS だけは提供されない高度な機能を受け取ることができるようになる。

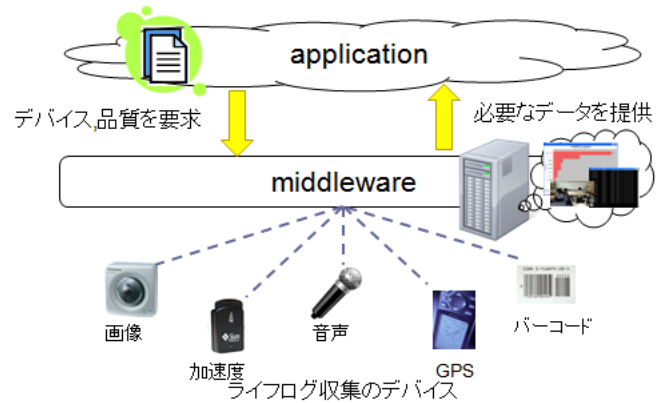


図 1 ミドルウェア概念図

2.3 アプリケーション例

アプリケーションの具体的な例として、お茶の水女子大学小林研究室で行われている「画像から人間の行動を言語化」する研究が挙げられる。この概要を図 2 に示す。この研究はカメラからの画像とセンサデータを用い解析することで人の行動を言語化するもので、プライバシーを考慮したライフログの収集や、人の行動を検索キーとして画像を検索することが可能になる。

このアプリケーションからミドルウェアに要求されることは、大量のデータから適切なデータを取得することと、アプリケーションの実行に必要な品質の保証である。また、画像とセンサデータを統合して用いるためにはカメラとセンサの正確な同期が必須である。そのため本研究では、このアプリケーションのミドルウェアとしてカメラとセンサの同期、カメラの画質の品質保証を行う。

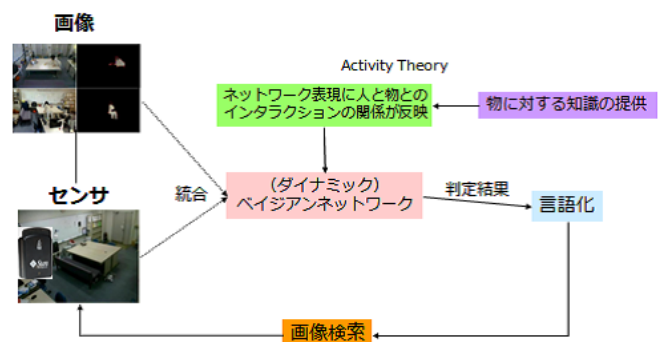


図 2 画像から人間の行動を言語化

3. 実験概要

3.1 実験環境

センサは SunSPOT [2] [3]、カメラは IP ネットワークを通じて動画を送信できるネットワークカメラを使用する。本実験では、Panasonic BB-HCM715 を使用した [4]。SunSPOT のプログラム開発は NetBeans IDE 6.5.1 を使用し、Java で開発を進める。画像処理には Microsoft Visual C++ 2008 を開発用ソ

ソフトウェアとして使用し、画像処理ライブラリ OpenCV1.1 を用いる。加速度のデータを記録するデータベース（以下 DB）には PostgreSQL8.4, JDBC には postgresql-8.4-701.jdbc3 を使用した。

ライフログの収集実験は、後述のお茶の水女子大学が所有する実験住宅 Ocha House で行う予定である。

3.2 センサノード SunSPOT

SunSPOT は、CPU 上で直接実行できる Java ベースのセンサーアプリケーションを構築するための技術として、サン・ラボで開発された無線センサネットワークデバイスである。加速度、温度、照度センサーと 8 個の 3 色 LED が搭載されており、CPU 上で直接実行できる Java ベースのアプリケーションを簡単に構築できる。また、IEEE802.15 無線ネットワーク経由でセンサ間の無線通信が可能である。

本研究では、SunSPOT センサノードと PC の双方に、SunSPOT の加速度センサのデータを利用するプログラムを実装した。このプログラムにより、センサノードで取得された加速度データが、IEEE802.15 無線ネットワークを通して SunSPOT ベースステーションに送信され、ベースステーションが接続された PC が、この加速度データを受信することができる。



図 3 SunSPOT

3.3 統合開発環境 NetBeans

NetBeans は、サン・マイクロシステムズを中心に開発されたオープンソースの Java 用統合開発環境である。SunSPOT のプログラム開発を行うために使用する。

3.4 ネットワークカメラ Panasonic BB-HCM715

ネットワークカメラとは、動画を IP ネットワークを通じて送信できるカメラの総称で、パソコンと直接 USB などで接続して使用する web カメラとは異なる。

本研究では、Panasonic 社の BB-HCM715 を使用した。このカメラで対応している方式は、

- JPEG
- MPEG-4
- H.264

がある。

対応している解像度は、

- 1280 × 960
- 640 × 480
- 320 × 240

である。

BB-HCM715 は、MPEG-4 や H.264 の動画で、ネットワークカメラとしては高解像度な 1280 × 960 で動作し、100 万画素以

上の解像度が必要なアプリケーションの要求にも応えられる。

3.5 統合開発環境 Microsoft Visual C++

Microsoft Visual C++ は、マイクロシステム製の C, C++ 用の統合環境開発である。最適化性能が高い。OpenCV を用い、画像処理を行うのに使用する。

3.6 画像処理ライブラリ OpenCV

OpenCV は、インテルが開発したオープンソースのコンピュータビジョン向けのライブラリである。画像処理に適している。本研究ではカメラの画像処理に使用する。本研究では OpenCV1.1 をインストールした。

3.7 データベース PostgreSQL

PostgreSQL は、BSD ライセンスにより配布されているオープンソースのオブジェクト関係データベースである。世界的な利用状況は低いですが、日本においては MySQL と二分するほどの利用率の高さを誇る。本研究では、PostgreSQL8.4 を利用する。

3.8 JDBC

JDBC は、Java とデータベースの接続のために使用される API である。Java において SQL を使用し、データベース管理システムなどと接続する機能を標準化している。本研究では、postgresql-8.4-70.jdbc3 を使用した。

3.9 実験住宅 Ocha House

お茶の水女子大学ユビキタスコンピューティング実験住宅（通称 Ocha House）は、人間の生活に身近な住居において、人間と情報環境が調和的に相互作用を行うための基盤技術の開発を目的として建てられた [5]。外観を図 4 に示す。

建物内には台所、トイレ、お風呂、寝室があり、通常の生活が送れるようになっている。建物内にセンサを設置し、人の居る場所を把握する実験や、高齢者の介護に活かせる研究なども行われている。2009 年度グッドデザイン賞を受賞した。



図 4 Ocha House

4. 同期システム構築

4.1 加速度センサ処理

SunSPOT から収集した x 軸方向、y 軸方向、z 軸方向の加速度データとその時の時刻をデータベース（以下 DB）に収納するプログラムを作成した。使用言語は Java である。

4.2 カメラの動画処理

OpenCV では、ネットワークカメラへの直接的なアクセスはサポートされておらず、また、画像のデータ処理を行うためには

リアルタイムでの処理よりもファイルに保存されたデータを処理するほうが汎用性が高いと考えられる。そのため、ネットワークカメラから取得されるブラウザ上の動画を動画キャプチャソフトでキャプチャし、avi形式で保存し、画像ファイルを利用する手法を採用した。

今回の実験では、MPEG-4の320×240の画像ストリームを、B's動画レコーダで、フレームレートは10fps、コーデックはMicrosoft MPEG-4 Video Codec V3でキャプチャしたaviファイルを使用した。

4.3 同期手順

カメラとセンサを同期させるために図5に示すシステムを構築した。まず、ユーザがある時刻を指定すると、その時刻をもってDBに問い合わせを行い、その時刻の加速度を返すプログラムを作成した。また、動画ファイルのタイムスタンプを取得し、ファイル作成時刻と指定された時刻との差分フレーム数を計算して、そのフレームからの再生を行うことで、指定時刻からの動画ファイルの再生を可能とするプログラムを作成した。使用言語はC++である。

このようなシステムを構築することで、センサとカメラの画像の時刻による同期を可能とした。

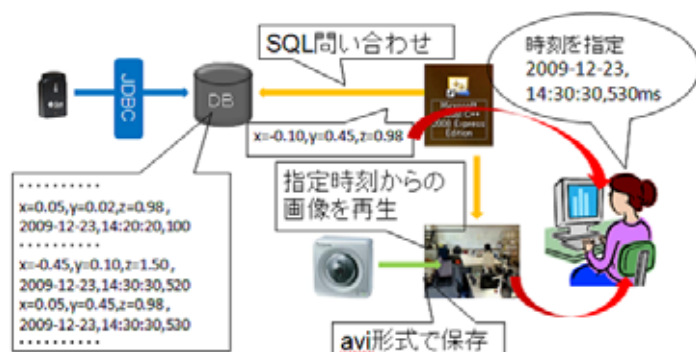


図5 同期システム

4.4 フレーム計算例

ファイルの作成日時が2009年12月23日14時30分28秒150ミリ秒で、10fpsの画像ファイルに対して、2009年12月23日14時30分30秒530ミリ秒を指定するとする。すると、ミドルウェアは、この時刻をもってDBにSQL問い合わせを行い、この時刻の加速度をDBから取得する。また、作成時刻と指定時刻の差が2秒380ミリ秒であるので、24フレームから動画ファイルの再生を行う（10ミリ秒四捨五入）。

5. まとめと今後の課題

本研究では、加速度センサと動画ファイルの時刻による同期を行うプログラムを作成した。今後はこのシステムをアプリケーションで実際に使用し、品質に関する評価を行う。画像の解像度の違いによるアプリケーションの品質の違いや、ネットワークカメラの使用時における有線と無線の品質の違いなどについて検

証していく。また、アプリケーションの要求するQoS(Quality of Service)を探り、データの品質を保証するミドルウェアを構築していく。

文 献

- [1] 寺田真二, ライフログビジネス, インプレス R&D, 2009
- [2] サン・マイクロシステムズ-Sun SPOT 無線センサーネットワークデバイス
<http://jp.sun.com/products/software/sunspot/>
- [3] Sun SPOT memo wiki
<http://www.klab.ai.kyutech.ac.jp/yatti/sunspot/index.php?Sun%20SPOT%20memo%20wiki>
- [4] Panasonic BB-HCM715
<http://panasonic.biz/netsys/netwcam/lineup/hcm715.html>
- [5] OchaHouse - PukiWiki
<http://ochahouse.com/>