

再生希望時間を考慮した携帯無線端末向け ビデオストリーミングシステム

玉井 森彦[†] 安本 慶一[†]
柴田 直樹^{††} 伊藤 実[†]

携帯無線端末上での動画のストリーミング再生は、動画のデコード処理、描画、無線通信デバイスの稼働に多くの電力を消費する。そのため、希望する再生時間の間バッテリーがもつよう省電力制御を行いたいといった要求や、バッテリー容量の指定した割合（たとえば、50%）は今後の作業のために残しておきたいといった要求が生じる。本論文では、携帯端末上での動画のストリーミング再生において、トランスコードによる動画の低品質化およびバッファリングによる無線デバイスの稼働時間の削減により、再生に要するバッテリー量を削減し、かつ、省電力化の度合いをユーザが指定した再生希望時間、動画の各品質項目に対する重要度および使用を許可したバッテリー量に従って制御することを目的としたビデオストリーミングシステムの提案を行う。提案システムを実装し、PDA およびノート PC を用いて実験を行った結果、PDA で 2.8 倍、ノート PC で 1.6 倍程度までバッテリー持続時間を延長できることを確認した。また、実用上十分な精度で、再生希望時間を満足できるようにバッテリー持続時間を制御できることを確認した。

Battery-aware Video Streaming System to Play Back for Specified Duration on Portable Wireless Devices

MORHIKO TAMAI,[†] KEIICHI YASUMOTO,[†] NAOKI SHIBATA^{††}
and MINORU ITO[†]

In streaming playback of video, portable wireless devices consume a lot of electric power for video data decoding and wireless communication. Therefore there are demands to control the power consumption so that the battery life can be extended to a desirable playback duration and/or the specified percentage of the battery capacity is left for the other applications. In this paper, we propose a battery-aware video streaming system to play back for specified duration considering a preference of quality of video specified by each user and the remaining battery amount for portable wireless devices. In the system, power consumption is reduced by using transcoding of video and periodic bulk transfer of video data. We have implemented the system and carried out several experiments using a PDA and a laptop PC. As a result, we have confirmed that our system can extend the battery life up to 2.8 times on the PDA and up to 1.6 times on the laptop PC, and can control the battery life with reasonably small error.

1. はじめに

PDA などの携帯無線端末の高性能化と無線通信機能の発展・普及、さらには無線 LAN ホットスポットや定額制 PHS、定額制 3G 携帯電話の登場により、ユーザが時や場所を選ばず、携帯端末上で映画やスポーツ中継などの動画コンテンツを楽しむことが可能となっ

てきた。しかし、携帯端末上での動画のストリーミング再生には、動画のデコード・描画に加え、無線通信に大きな電力を必要とするため、そのバッテリー消費は他のアプリケーションに比較して大きい。そのため、動画を見ている途中でバッテリーが尽きてしまわないよう省電力化を行いたいという要求や、バッテリー容量のたとえば 50% を、指定した時間の間動画再生に使用し、残りの 50% は作業用に残しておきたい、という要求が生じる。したがって、ユーザが指定した再生時間（以下、再生希望時間とよぶ）の間、動画再生を可能とするための省電力制御機構の実現が求められる。

これまでのところ、動画再生および無線通信におけ

[†] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

^{††} 滋賀大学経済学部

Faculty of Economics, Shiga University

る省電力化へのアプローチは、ハードウェアによるものが多い¹⁾⁻³⁾。一方、ソフトウェアによる無線通信のための省電力化手法として、現在までに、IEEE802.11b規格の無線 LAN において、電波の出力を動的に変更することにより省電力化を実現する方式⁴⁾や、無線アドホックネットワークにおいて、パス上のすべてのノードにおける電力消費の和を最小化するパスを選択するための省電力ルーティング方式⁵⁾などが提案されている。また、文献6)では、無線 LAN 経由でのデータ受信時において、BWT, LZ77などの可逆圧縮を用いて通信時間を短縮することにより省電力化を実現する方式が提案されている。一方、動画データを対象とした省電力化手法として、文献7)では、携帯端末上で録画した動画のエンコードおよび無線経由での動画の送信時において、I, P, B フレームのいずれかを選択的に間引くことで転送すべきビット数を減らし、省電力化を実現する手法が提案されている。

本論文では、携帯端末上での動画のストリーミング再生において、再生に要するバッテリー量を削減し、かつ、省電力化の度合いをユーザが指定した再生希望時間および使用可能なバッテリー量に従って制御することを目的としたビデオストリーミングシステムの提案を行う。

省電力化手法として、(i) トランスコード技術を用いた手法、(ii) バッファリング再生を用いた手法、の2つを考案した。(i)では、トランスコードを実行する中間ノードによって、動画ストリームをリアルタイムで変換し、動画の品質(画像サイズ、フレームレート、ビットレート(以下、この3つ組を動画パラメータとよぶ)の各値)を下げることで、省電力化を実現する。(ii)では、無線 LAN I/F カードの稼働時間中に、なるべく多くの動画ストリームを受信してバッファリングしておき、バッファ内にデータが存在する間、通信デバイスの電源を切ることによりバッテリー消費量の削減を実現する。

上記の省電力化手法を使用する際、再生希望時間を満足できるようなトランスコード後の動画パラメータ値をユーザが自分で求めることは困難である。そこで提案システムでは、動画再生時の端末における電力消費を近似するためのモデルを使用し、トランスコード後の動画パラメータ値を自動的に決定する。その際、ユーザからは動画の品質に関するプリファレンス値(動画パラメータ値の各重要度を表す重み)を入力として受け取り、変換後の動画がプリファレンス値を考慮した品質となるように調節する。また、携帯端末に固有の電力消費の特性を考慮するために(2) 端末情報

(ある動画を再生したときの単位バッテリー量を消費するまでの時間、利用可能な通信帯域など)を利用する。

提案システムを実装し、PDA およびノート PC を用いて実験を行った結果、提案する省電力化手法が有効であることを確認した。また、電力消費モデルにより算出した品質で実際に動画を再生した結果、実用上十分な精度で再生希望時間を満足できるようにバッテリー持続時間を制御できることを確認した。

2. 省電力ビデオストリーミングシステム

携帯端末上で動画のストリーミング再生を行う状況として、本論文では、バッテリー駆動型の携帯端末を用い、無線通信デバイスとして無線 LAN を使用する環境を対象とする。このような環境下で、ユーザが

- 動画の保存場所
- 再生希望時間(たとえば、1時間)
- 動画の品質に関するプリファレンス値

を指定することで、再生希望時間を満足できるように省電力制御を行うビデオストリーミングシステムを提案する。上記の環境として、たとえば飛行機、電車、喫茶店内での無線 LAN ホットスポットにおいて比較的長時間の動画の視聴を行う場合が考えられ、提案する省電力制御の仕組みは現実の環境で有用であると考える。

以下では、提案システムにおける消費電力の削減方針を述べる。

2.1 省電力化の方針

携帯端末上でストリーミング再生を行う際に消費される電力は、大きく次の3つに分類できる。

- (1) 動画のデコード・描画に必要な電力
- (2) 無線通信デバイスの稼働・パケット処理に必要な電力
- (3) 上記以外で必要となる電力(OSの稼働・バックライトの点灯などに必要な電力)

動画の再生に関する電力((1)と(2)の和)の、全消費電力に占める割合を調べるため、PDA と、CF カード型 802.11b 対応無線 LAN I/F カードを用いて(機器の詳細は表2参照)、動画(圧縮符号化方式 MPEG-1, 画像サイズ 288×216 pixel, フレームレート 24 fps, ビットレート 327 kbps)をストリーミング再生したときのバッテリー持続時間 T_s を測定し、動画再生および通信をいっさい行わないときのバッテリー持続時間 T_b との比較を行った。動画再生および通信をいっさい行わないときの消費電力を P_b 、ストリーミング再生を行うことによる上記(1)、(2)に関する消費電力の増分を P_s 、バッテリー容量を E とすると、

$E = T_b P_b$, $E = T_s (P_b + P_s)$ が成り立つ. T_b , T_s の実測値はそれぞれ 440 分, 120 分となった. したがって, $278P_b = 120P_s$ となるため, $P_b + P_s = 1$ とおくと, 上記 (1), (2) の和は全消費電力に対して約 78% を占めることが分かる. また, ノート PC (機器の詳細は表 2 参照) に関しても, 560×420 pixel, 24 fps, 818 kbps の動画と最大充電時の 10% 分のバッテリー量を用いて, T_b , T_s の実測値を求めた. T_b , T_s の実測値はそれぞれ 43 分, 27 分となり, (1) と (2) の和は全消費電力に対して約 63% を占めることが分かる. 以上より, PDA, ノート PC いずれにおいても (1) と (2) による消費電力を削減することで大きな省電力化が可能となる.

2.1.1 動画のデコード・描画に必要な電力の削減

動画のデコード・描画に必要な電力は, 動画の画像サイズ, フレームレート, ビットレートの増加にともなって大きくなる. したがって, これらの値を下げることで, 省電力化が可能となる. 提案システムでは, トランスコード技術を用いて, 動画配信サーバから送信される動画を, それよりも少ない消費電力で再生可能な動画へとリアルタイムで変換することで省電力化を実現する. ただし, トランスコードを携帯端末上で実行すると, 変換処理によって新たな電力消費を招くため, 動画配信サーバと携帯端末の間にトランスコードを行う中間ノード (以下, トランスコードボックスとよぶ) を設置し, 変換後のストリームを携帯端末へ中継することにより, 省電力化を実現する.

2.1.2 無線通信によって消費される電力の削減

無線通信によって消費される電力の削減に関して, IEEE802.11b 規格の無線 LAN I/F カード (以下, WNIC とよぶ) を用いて次のような予備実験を行った. 予備実験 同一バッテリー量のもとで, 特定の伝送速度でデータを受信し続けた場合の, 伝送速度の変化に対するバッテリー持続時間の変化を測定した. 測定には, 表 2 における PDA と, 固有の WNIC に依存した測定結果ではないことを確認するため, 2 種類の WNIC (WNIC1: WN-B11/CF (I-O Data), WNIC2: WLI-CF-S11G (BUFFALO)) を使用した. 実験結果を図 1 に示す. また, WNIC を稼働しない場合のバッテリー持続時間は 440 分であった.

予備実験の結果より, PDA を用いた無線 LAN 経由でのデータ受信においては, 同じデータ量を受信する場合, 低い伝送速度で長時間データを受信するよりも, 高い伝送速度で短時間にデータを受信した方が, 電力量の消費が少ないことが分かる. これは, WNIC による電波の出力が一定である場合, 消費電力はその

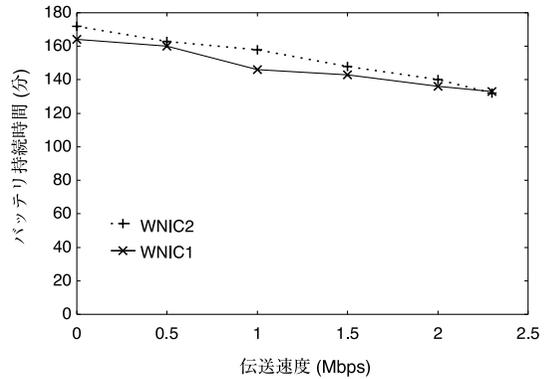


図 1 WNIC の伝送速度の変化に対するバッテリー持続時間の変化
Fig. 1 Battery life vs. transmission rate on WNIC.

出力に大きく依存し, 伝送速度の増加に対する消費電力の増分の割合は, 比較的軽微であるためだと思われる. なお, 図 1 で見られる伝送速度の増加に対する消費電力の増加は, OS におけるパケット処理によるものが大きい.

また, ノート PC に関しても同様の性質が満たされるかを確認するため, 表 2 におけるノート PC と 2 種類の WNIC (WNIC1: CF カード, GW-CF11H (Planex), Prism 2.5, WNIC2: PCMCIA type 2 カード, GW-NS11H (Planex), Prism 3) を用いて, 次の 3 種類のバッテリー持続時間 (総バッテリー容量の 10% 分を使用) を調べた: (1) WNIC を稼働しないとき, (2) WNIC を稼働し, 通信をいっさい行わないとき, (3) WNIC を稼働し, 特定の伝送速度 (WNIC1: 2.0 Mbps, WNIC2: 1.2 Mbps) で受信を行うとき. バッテリー持続時間はそれぞれ (1) 43 分, (2) WNIC1: 38 分, WNIC2: 39 分, (3) WNIC1: 37 分, WNIC2: 38 分となった. このことから, ノート PC においても, PDA の場合と同様の傾向を持つことが分かる.

ストリーミング再生する場合, 動画がビットレート b でエンコードされているならば, 通常, 平均伝送速度 b で動画ストリームの受信を行うことになり, この間 WNIC は常時稼働状態にある. したがって, WNIC が上記のような性質を持ち, かつ, 利用可能帯域に余裕がある場合には, 利用可能な最大の伝送速度で動画ストリームを一度に受信してバッファリングしておき, 余裕のできた時間の間, WNIC の電源を切ることで省電力化が可能となる. 具体的には, 次のような手順で動画の再生を行う.

- (1) サイズ M ビットのバッファを用意する.
- (2) WNIC を稼働し, 動画ストリームを伝送速度 B_{bf} (ただし, $B_{bf} > b$) でバッファ内へ受信するスレッド T_r と, バッファ内の動画データを

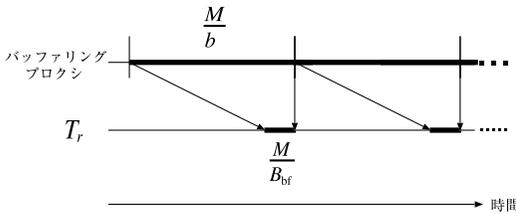


図2 バッファリングプロキシと T_r 間の通信時のタイムチャート
Fig. 2 Time chart of communication between buffering proxy and a user terminal.

- 再生するスレッド T_p を生成し、並列実行する。
- (3) バッファが満たされたら、 T_r を待機状態にし、WNIC の電源を切る。
 - (4) バッファが空になったら、WNIC を稼働し、 T_r を実行状態にする。
 - (5) (3) からの処理を繰り返す。

以降では、この手順を用いた動画再生方式をバッファリング再生とよぶこととする。バッファリング再生はトランスコードによる省電力化との併用が可能であり、その場合にはトランスコードプロキシから送信される変換後の動画ストリームをバッファリング再生する。

動画の配信がリアルタイムに行われている場合には、利用可能帯域に余裕がある場合でも、動画ストリームをビットレート b 以上の伝送速度で受信することはできないため、上記のままではバッファリング再生ができない。そこで、携帯端末上のバッファサイズと同一サイズのバッファを持つ中間ノード（以下、バッファリングプロキシとよぶ）を携帯端末と動画配信サーバ（もしくはトランスコードプロキシ）間に設置する。初めにバッファリングプロキシが、動画ストリームをバッファが満たされるまでバッファリングする。次に、携帯端末は、バッファリングプロキシがバッファリングした動画ストリームを、伝送速度 B_{bf} で受信することにより、バッファリング再生が可能となる。

バッファリングプロキシと携帯端末内のスレッド T_r 間の、通信時のタイムチャートを図2に示す。図2において、バッファリングプロキシの太線は、動画配信サーバからの動画ストリームの受信を表しており、 T_r の太線は、携帯端末における、バッファリングプロキシからの動画ストリームの受信を表している。リアルタイム配信であるため、バッファリングプロキシは常時、動画ストリームの受信を行っていなければならない。

バッファリングプロキシを使用する場合、動画の再生を開始するためには、バッファリングプロキシ上のバッファが満たされるまでの、 $(M/b) - (M/B_{bf})$ 秒

の待ち時間が必要となる。待ち時間はバッファサイズ M に依存するが、 M は再生を開始するまでの許容待ち時間をユーザから入力として受け取ることで決定できる。

トランスコードプロキシとバッファリングプロキシは、別々のマシン上に配置することが可能であるが、以下では簡単のために、トランスコードプロキシとバッファリングプロキシを同一マシン上に配置することとする。

2.2 システム構成

提案する省電力ビデオストリーミングシステムの構成を、図3に示す。

ユーザは、見たい動画の保存場所と再生希望時間をトランスコードプロキシに与える。プロキシは指定されたバッテリー量のもとで、再生希望時間を満足できる動画パラメータ値を求めるが、一般に解は複数存在する。そこで、ユーザから動画の品質に関するプリファレンス値を入力として受け取り、その値に基づいて解を一意に決定する。また、端末ごとに固有な電力消費特性を考慮するために、端末情報を利用する（プリファレンス値、端末情報の詳細は次章で述べる）。これらの情報から、プロキシは次章で述べる省電力パラメータ決定機構を用いて変換後の動画パラメータ値を求める。

3. 省電力パラメータ決定機構

3.1 電力消費モデル

本節では、省電力パラメータ決定機構が使用する、動画再生時の端末における電力消費を近似するためのモデルについて述べる。

提案モデルでは、動画の再生によって消費される電力を P_v 、動画データの受信によって消費される電力を P_n 、それ以外の消費電力（OSの稼働・バックライトなどによって消費される電力）を S としたとき、端末の消費電力を次式で表す。

$$S + P_v + P_n \quad (1)$$

3.1.1 動画の再生によって消費される電力

動画パラメータ値が (r, f, b) である動画の再生によって消費される電力 P_v を考える。ただし、 r は画像サイズ（画素数）、 f はフレームレート、 b はビットレートである。MPEG-1 フォーマットの動画を再生するのに必要な処理は、動画のデコード処理と表示処理に分類できる。デコードに必要な処理はさらに可変長復号、逆量子化、逆離散コサイン変換に分類できる。可変長復号に必要な電力はビットレートに比例し、逆量子化、逆離散コサイン変換に必要な電力は画素数に

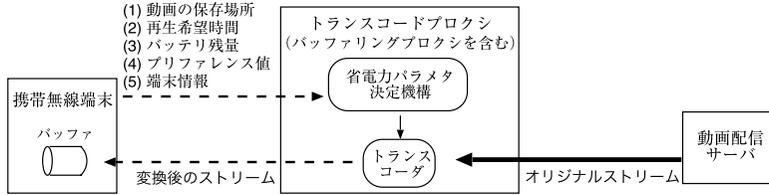


図3 省電力ビデオストリーミングシステムの構成
Fig. 3 Battery-aware video streaming system.

比例する。また、表示処理におけるビデオデバイスへの書き込み、描画に必要な電力は画素数に比例する。以上より、動画の再生によって消費される電力 P_v は、 α, β をある定数として、以下の式で表される。

$$P_v(r, f, b) = \alpha r f + \beta b \quad (2)$$

したがって式 (1) より、動画再生時 (通信は行わない) に消費される合計電力は、次式で表される。

$$S + \alpha r f + \beta b$$

S, α, β は端末ごとに固有の定数値であり、実際の値は、異なった動画パラメータ値を持つ n 個の動画 $\{(r_1, f_1, b_1), \dots, (r_n, f_n, b_n)\}$ に対して、特定のバッテリー量 (たとえば、最大充電時のバッテリー量) E_0 を使い果たすまでの時間 t_{vi} ($i = 1, 2, \dots, n$) を測定して得られる n 個の方程式 $(S + \alpha r_i f_i + \beta b_i) = E_0 / t_{vi}$ から求めることができる。求めたい定数の数は 3 個であるため、 $3 \leq n$ でなければならない。また、 $4 \leq n$ とすることで、重回帰分析によって、より精度の高い S, α, β の値を求めることができる。

3.1.2 データの受信によって消費される電力

平均伝送速度 B でデータを受信する際に、WNIC およびパケット処理によって消費される電力 P_n を考える。2.1.2 項の予備実験の結果より、WNIC によって消費される電力および、パケット処理によって消費される電力は、平均伝送速度 B の一次式で表される。よって、 N, γ をある定数として、 P_n は次式で表される。

$$P_n(B) = N + \gamma B \quad (3)$$

式 (1) より、データ受信時 (動画の再生は行わない) に消費される合計電力は、次式で表される。

$$S + N + \gamma B$$

N, γ は端末ごとに固有の定数値であり、実際の値は、異なった伝送速度 B_i ($i = 1, 2, \dots, m$) に対して、特定のバッテリー量 E_0 を使い果たすまでの時間 t_{ni} を測定して得られる m 個の方程式 $(S + N + \gamma B_i) = E_0 / t_{ni}$ から求めることができる。 P_v の場合と同様に、 $3 \leq m$ である場合は重回帰分析によって、より精度の高い N, γ の値を求めることができる。

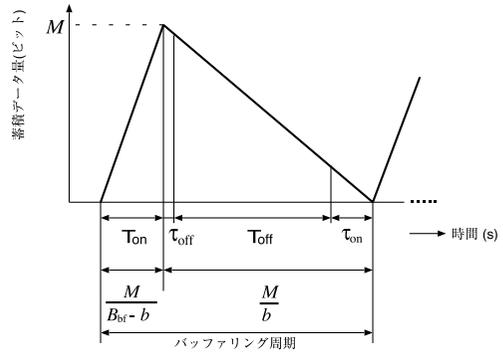


図4 バッファリング周期内におけるバッファ内の蓄積データ量の時間変化
Fig. 4 Time chart of the remaining data in a buffering period.

3.1.3 バッファリング再生時の消費電力

バッファリング再生時の平均伝送速度を B_{bf} とする。受信スレッドがバッファ内に動画ストリームを受信し始めてから、バッファが満たされ、再度バッファが空になるまでの時間を、バッファリング周期とよぶこととする。バッファリング周期内で、WNIC が稼働状態である時間を T_{on} 、電源を切った状態である時間を T_{off} とする。また、WNIC が電源を切った状態から稼働状態になるまでの時間を τ_{on} 、稼働状態から電源を切った状態になるまでの時間を τ_{off} とする。 $\tau = \tau_{on} + \tau_{off}$ とおく。

バッファリング周期内における、バッファ (サイズ M ビット) 内の蓄積データ量の時間変化を図 4 に示す。ただし、動画はビットレート b でエンコードされているとする。図 4 より、

$$T_{on} = \frac{M}{B_{bf} - b}, \quad T_{off} = \frac{M}{b} - \tau$$

となる。

バッファリング再生時に、動画データの受信によって消費される電力 P_{bf} は、WNIC が稼働状態の場合には $P_n(B_{bf})$ 、電源を切った状態の場合には 0 であるが、平均的には次式で表すことができる。

$$P_{bf}(b) = \frac{T_{on}}{T_{on} + T_{off} + \tau} P_n(B_{bf})$$

$$= \frac{b}{B_{bf}} (N + \gamma B_{bf})$$

WNICは τ の間、IPアドレスの設定(DHCPサーバを使用する場合には、IPアドレスの取得も含む)や無線LANアクセスポイントとのネゴシエーションなどにより、データの送受信はできないが、電力を消費する。 τ の間のWNICでの消費電力を $P_n(0)$ とすると、バッファリング再生時に、 τ の間に消費される電力は、平均的に次式で表すことができる。

$$P_{oh}(b) = \frac{\tau}{T_{on} + T_{off} + \tau} P_n(0)$$

$$= \frac{b(B_{bf} - b)\tau}{MB_{bf}} N$$

以上より、動画パラメータ値が (r, f, b) の動画をバッファサイズ M 、伝送速度 B_{bf} でバッファリング再生するときの、バッテリー量 E を使い果たすまでの時間 t は、次式で与えられる。

$$t = E / (S + \alpha r f + \beta b + P_{bf}(b) + P_{oh}(b)) \quad (4)$$

3.2 トランスコード後の動画パラメータ値の決定

本節では、上記の電力消費モデルを使用して、再生希望時間を満足できるような変換後の動画パラメータ値を自動的に決定する方法を述べる。

動画配信サーバから送信される動画のパラメータ値を (r_0, f_0, b_0) とする。トランスコードプロキシによる変換後の動画パラメータ値を (r', f', b') とする。再生希望時間を満足できる変換後の動画パラメータ値は複数存在する。一方、ユーザの要求として、画像サイズ、フレームレート、ビットレートのどれを重視するかを指定したいということが考えられる。そこで、画像サイズ、フレームレート、ビットレートの、各重要度を表す重みの組 (w_r, w_f, w_b) ($1 \leq w_r, w_f, w_b$)をユーザから入力として受け取り、これに基づいて (r', f', b') を決定する。具体的には、 (w_r, w_f, w_b) 、 (r_0, f_0, b_0) 、 (r', f', b') の間に、次式の関係が成り立つとする。

$$\frac{r'}{r_0} : \frac{f'}{f_0} : \frac{b'}{b_0} = w_r : w_f : w_b \quad (5)$$

式(5)は、たとえばフレームレートを重視して、 $(w_r, w_f, w_b) = (1, 2, 1)$ とした場合、動画配信サーバからの動画の画像サイズが、トランスコードによって1/3に縮小されるとき、フレームレート、ビットレートはそれぞれ2/3、1/3となることを意味している。式(5)より、 $r'/r_0 = s$ とするとき次式が成り立つ。

$$r' = s r_0$$

$$f' = s f_0 w_f / w_r$$

$$b' = s b_0 w_b / w_r$$

以上より、バッファリング再生を使用する際に、

- 動画配信サーバからの動画のパラメータ値 (r_0, f_0, b_0)
- 再生希望時間 t_d
- バッテリー残量 E_r
- プリファレンス値 (w_r, w_f, w_b)
- 端末情報 $S, \alpha, \beta, N, \gamma, M, B_{bf}, \tau$

が与えられたとき、 t_d を満足する変換後の動画パラメータ値 (r', f', b') は次のように求めることができる：式(4)に対し、 t に再生希望時間 t_d を、 E にバッテリー残量 E_r を、 (r, f, b) に (r', f', b') をそれぞれ代入して得られる s の方程式をニュートン法などで解く。

3.3 主観評価に基づいた動画の品質に関するモデル

3.2節では、変換後の動画パラメータ値を決定するにあたって、動画パラメータの各値に対するユーザの重要度を利用する方法を述べた。しかし、ビットレートの重要度を高くとる場合、変換後の画像サイズ、フレームレートに対して、変換前の動画と同程度の量子化ノイズを維持するのに十分である値以上のビットレートを割り当ててしまう可能性が生じる。本節では、変換前と同程度の量子化ノイズである動画を得るためには、どの程度のビットレートでエンコードすればよいかを、主観評価に基づいて求める方法を述べる。

次の仮定が成り立つとする。

仮定 フレームレート(画像サイズ)が一定のもとで画像サイズ(フレームレート)を変化させたとき、変換前と同程度の量子化ノイズを維持するのに必要なビットレートは、画像サイズ(フレームレート)の一次式で近似できる。

この仮定より、画像サイズ r (pixel)、フレームレート f (fps)の動画のビットレート b (kbps)は、 c_0, c_1, c_2, c_3 をある定数として以下の式により求められる。

$$b = c_0 r f + c_1 r + c_2 f + c_3 \quad (6)$$

定数 c_0, c_1, c_2, c_3 を求めるために、次の実験を行った：画像サイズとフレームレートの64個の組合せに対して、各々同程度の量子化ノイズを維持するのに必要なビットレートを主観評価によって測定した。式(6)に実験の測定値 (r_i, f_j, b_{ij}) ($i, j = 1, 2, \dots, 8$)を代入し、重回帰分析によって c_0, c_1, c_2, c_3 を求めた結果、それぞれの値は $c_0 = 7.9 \times 10^{-5}$ 、 $c_1 = 4.2 \times 10^{-4}$ 、 $c_2 = 13$ 、 $c_3 = -16$ となった。

式(6)を使用すると、変換後のビットレート b' は、ビットレートに対する重要度を表す重み w_b からでは

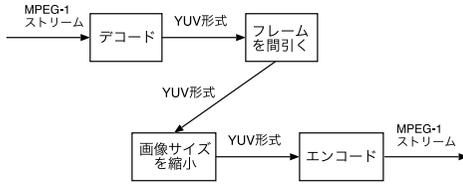


図 5 MPEG-1 ストリームの変換手順

Fig. 5 Processes to transcode MPEG-1 stream.

なく、 $b' = c_0r'f' + c_1r' + c_2f' + c_3$ によって求めることができる。

4. 実 装

動画プレイヤーの実装には、Berkeley MPEG Player⁹⁾ を利用した。省電力パラメータ決定機構は C 言語を用いて実装した。

動画ストリームの変換を実現するためには、画像サイズの縮小、フレームを間引く、ビットレートを下げるといった動画パラメータ値の変換処理を実装する必要がある。ここでは、MPEG-1 ストリームを一度デコードし、YUV 形式の画像のシーケンスに対して処理を行う方式をとった。処理の手順を図 5 に示す。図 5 におけるデコード、画像サイズの縮小、エンコードの処理には、MJPEG Tools¹⁰⁾ を使用した。また、フレームを間引く処理には、C 言語による自作のプログラムを使用した。

トランスコーダの性能

提案システムの実用のためには、動画ストリームの変換がリアルタイムで行え、また、必要となる計算機リソースが、現実的な範囲内であることが求められる。そこで、一般に入手可能である PC/AT 互換機 (Intel Pentium 4 2.40 GHz×2, 1.0 GB RAM, Debian GNU/Linux 3.0, Kernel 2.4.27) を用いて動画ストリームの変換を行い、変換時の処理速度を計測した。具体的には、再生時間が 300 秒である動画を使用し、それを特定のパラメータ値を持つ動画へと変換する際の 1 秒ごとの処理速度 (fps) を記録し、300 秒間における平均処理速度を求めた。この平均処理速度が、変換後の動画における再生速度 (変換後の動画のフレームレート) よりも大きければ、トランスコーダが動画データを変換しながら動画プレイヤーへ送信することで、動画プレイヤー上では変換後の動画を途中で途切れることなく再生できる。実験結果を表 1 に示す。

表 1 より、各変換ともリアルタイムで実行可能であることが分かる。また、画像サイズの縮小やフレームを間引くことで、変換に必要な処理時間が減少することが分かる。これは、処理時間の大部分を占める

表 1 トランスコーダの処理速度

Table 1 Processing speed for video transcoding.

動画パラメータ値 (pixel, fps, kbps)	処理速度 (fps)
変換前の動画	
(640 × 480, 24, 1100)	—
変換後の動画	
(560 × 420, 24, 818)	25.9
(560 × 420, 12, 788)	48.2
(320 × 240, 12, 788)	163.2
(320 × 240, 12, 294)	177.2

表 2 実験環境

Table 2 Experimental environments.

携帯端末の種類, 名前, 設定	CPU, OS	無線 LAN カードの種類, 名前, 設定
PDA, Sharp Zaurus	XScale PXA250 400 MHz,	CF, WN-B11/CF
SL-C700, 輝度 小	Linux (Embedix)	(I/O Data), 省電力モード off
laptop PC, IBM Thinkpad	PentiumIII 600 MHz,	CF, GW-CF11H
s30, 輝度 小	Linux 2.4.27	(Planex), 省電力モード off

エンコードに要する計算量が、画素数の減少とともに小さくなるためである。

5. 実験 結果

5.1 実験 環境

提案した省電力化手法の有効性、および電力消費モデルの予測精度が妥当であることを検証するため、表 2 に示す 2 つの環境下での実験を行った。なお、ノート PC での実験では、HDD の回転による電力消費を抑えるため、noflushd¹¹⁾ を使用した。

端末情報 $S, \alpha, \beta, N, \gamma, \tau$ の各値を PDA, ノート PC 上でそれぞれ求めた。なお S, α, β の各値を求めるため、PDA, ノート PC それぞれ表 3, 表 4 に示す 4 種類の動画を再生し、バッテリー量 E_0 を使い果たすまでの時間を測定した。また、 N, γ の各値を求めるため、PDA, ノート PC 上で、それぞれ表 5, 表 6 に示す 2 種類の伝送速度でデータを受信し、バッテリー量 E_0 を使い果たすまでの時間を計測した。また、 E_0 として、PDA, ノート PC 各々最大充電時のバッテリー容量を使用した。PDA, ノート PC における端末情報の各値を表 7 に示す。

5.2 動画のデコード・描画に対する電力消費モデルの精度

3.1.1 項で提案した動画のデコード・描画に対する電力消費モデルの妥当性を検証するため、バッテリー量

表 3 S, α, β を求めるために使用した動画 (PDA)Table 3 Videos used for measuring a value of S, α, β for the PDA.

動画パラメータ値 (pixel, fps, kbps)	E_0 が尽きるまでの時間 (min)
(166 × 124, 24, 112)	309
(166 × 124, 24, 321)	293
(288 × 216, 24, 327)	216
(288 × 216, 8, 317)	294

表 4 S, α, β を求めるために使用した動画 (ノート PC)Table 4 Videos used for measuring values of S, α, β for the laptop PC.

動画パラメータ値 (pixel, fps, kbps)	E_0 が尽きるまでの時間 (min)
(320 × 240, 24, 299)	374
(320 × 240, 24, 813)	366
(560 × 420, 24, 818)	281
(560 × 420, 8, 788)	363

表 5 N, γ を求めるために使用した伝送速度 (PDA)Table 5 Transmission rates used for measuring a value of N, γ for the PDA.

伝送速度 (Mbps)	E_0 が尽きるまでの時間 (min)
0.5	160
2.3	133

表 6 N, γ を求めるために使用した伝送速度 (ノート PC)Table 6 Transmission rates used for measuring values of N, γ for the laptop PC.

伝送速度 (Mbps)	E_0 が尽きるまでの時間 (min)
0.5	395
2.0	362

E_0 のもとで動画の再生を行い (通信は行わない), E_0 が尽きるまでの時間 (以下, 実測バッテリー持続時間とよぶ) を測定した. なお, 提案システムの実際の使用においては, 再生希望時間を与え動画パラメータ値を決定するが, ここではモデルの妥当性を検証することが目的であるため, 動画パラメータの各値が比較的離れた値を持つ複数の動画をあらかじめ指定し, それを再生したときの, 電力消費モデルによるバッテリー持続時間の予測値 (以下, 予測バッテリー持続時間とよぶ) を求め, バッテリー持続時間との比較を行った.

PDA, ノート PC の実験結果をそれぞれ表 8, 表 9 に示す. なお, 予測バッテリー持続時間とともに予測誤差 (予測バッテリー持続時間 t_e の, 実測バッテリー持続時間 t に対する予測誤差を $(|t - t_e|/t_e)$ と定義する) を示した.

表 3, 4 および表 8, 9 より, 画像サイズ, フレー

表 7 電力消費モデルで使用する定数値

Table 7 Constant values for energy consumption model.

定数	PDA	ノート PC
S	0.00254238 E_0	0.00223922 E_0
α	1.21931e-09 E_0	2.15017e-10 E_0
β	8.15911e-07 E_0	1.29409e-07 E_0
N	0.00335518 E_0	0.000215492 E_0
γ	0.000704887 E_0	0.000153857 E_0
τ	3	3

表 8 動画再生における (通信は行わない) 実測バッテリー持続時間と予測バッテリー持続時間 (PDA)

Table 8 Actual and predicted battery lives without communication on the PDA.

動画パラメータ値 (pixel, fps, kbps)	バッテリー持続時間 (min) 実測値/予測値 (誤差 %)
(288 × 216, 24, 123)	235/224 (4.9%)
(288 × 216, 8, 110)	318/309 (2.9%)
(166 × 124, 8, 231)	363/341 (6.5%)
(166 × 124, 8, 109)	370/353 (4.8%)

表 9 動画再生における (通信は行わない) 実測バッテリー持続時間と予測バッテリー持続時間 (ノート PC)

Table 9 Actual and predicted battery lives without communication on the laptop PC.

動画パラメータ値 (pixel, fps, kbps)	バッテリー持続時間 (min) 実測値/予測値 (誤差 %)
(560 × 420, 24, 458)	290/284 (2.1%)
(560 × 420, 8, 286)	374/373 (0.3%)
(320 × 240, 8, 788)	407/404 (0.7%)
(320 × 240, 8, 294)	415/415 (0.0%)

ムレート, ビットレートの各値を下げることで消費電力の削減が可能であることが分かる. また, 単位時間あたりに処理すべき画素数の削減による省電力化の効果は, ビットレートの削減による省電力化の効果に比べて大きいことが分かる. また, 電力消費モデルによる予測バッテリー持続時間が実際のバッテリー持続時間を 6.5% 以内の誤差で近似できていることが分かる. したがって, 3.1.1 項で提案した動画のデコード・描画に対する電力消費モデルは実用上十分妥当であるといえる.

5.3 トランスコードとバッファリング再生を併用した場合の省電力効果と予測精度

トランスコードとバッファリング再生の併用による, 省電力化の効果および, 電力消費モデルによる予測精度を調べるため, バッテリー量 E_0 のもとでバッファリング再生を行い, その実測バッテリー持続時間と電力消費モデルによる予測バッテリー持続時間を求めた. バッファリング再生時の平均伝送速度は, PDA, ノート PC でそれぞれ 2.3 Mbps, 1.8 Mbps であり, バッファサイズはともに 5 M バイトである. PDA, ノート PC

表 10 トランスコードとバッファリング再生の併用時の実測バッテリー持続時間と予測バッテリー持続時間 (PDA)

Table 10 Actual and predicted battery lives using both transcoding and buffering playback on the PDA.

動画パラメータ値 (pixel, fps, kbps)	バッテリー持続時間 (min) 実測値/予測値 (誤差 %)	
	ストリーミング再生	バッファリング再生
(288 × 216, 24, 327)	120 / 122 (1.6%)	189 / 185 (2.2%)
(166 × 124, 24, 321)	142 / 143 (0.7%)	242 / 240 (0.8%)
(288 × 216, 8, 110)	149 / 150 (0.7%)	289 / 285 (1.4%)
(166 × 124, 8, 109)	163 / 160 (1.9%)	340 / 323 (5.3%)

表 11 トランスコードとバッファリング再生の併用時の実測バッテリー持続時間と予測バッテリー持続時間 (ノート PC)

Table 11 Actual and predicted battery lives using both transcoding and buffering playback on the laptop PC.

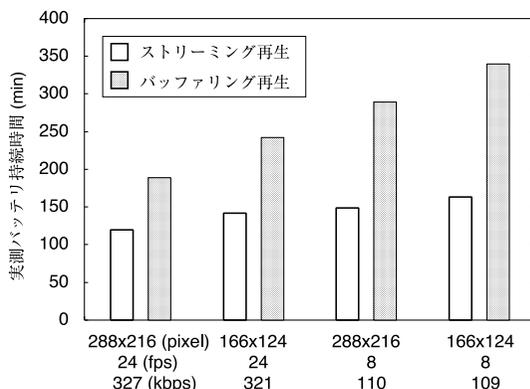
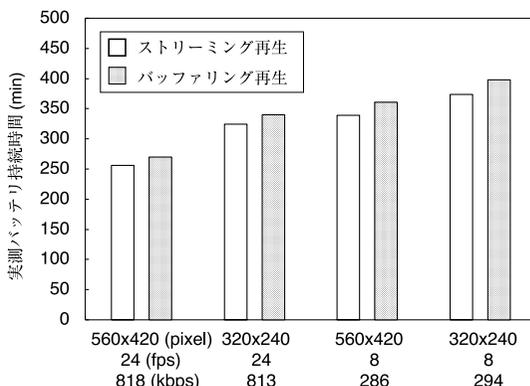
動画パラメータ値 (pixel, fps, kbps)	バッテリー持続時間 (min) 実測値/予測値 (誤差 %)	
	ストリーミング再生	バッファリング再生
(560 × 420, 24, 818)	256/256 (0.0%)	270/264 (2.3%)
(320 × 240, 24, 813)	324/325 (0.3%)	340/337 (0.9%)
(560 × 420, 8, 286)	339/340 (0.3%)	361/362 (0.2%)
(320 × 240, 8, 294)	374/375 (0.3%)	398/401 (0.7%)

上での実験結果をそれぞれ表 10, 表 11 に示す. また, 表 10, 11 における実測バッテリー持続時間を図 6, 図 7 にそれぞれグラフ化して示した.

表 10, 11, 図 6, 7 より, バッファリング再生による省電力化が有効であることが分かる. また, バッファリング再生はノート PC よりも PDA に対して, より有効であることが分かる. これは, PDA における WNIC の稼働による電力消費の割合が, ノート PC に比べて大きいためである. また, 電力消費モデルによる予測誤差は, 5.3%以内に抑えられており, 提案システムを使用することで, 実用上十分な精度で再生希望時間を満足できるようにバッテリー持続時間を制御できることが分かる. なお, 実際に提案システムを使用する際には, 誤差のため再生希望時間よりも少ない時間でバッテリー量が尽きることが考えられるが, 誤差の分だけバッテリー残量を少なめに見積もっておくなどの方法で対応できる.

6. おわりに

本論文では, 携帯無線端末上での動画のストリーミング再生における, 再生希望時間とバッテリー残量および動画の品質に関するプリファレンス値を考慮した, 省電力ビデオストリーミングシステムを提案した. 提案システムを実装し, PDA およびノート PC を用いて実験を行った結果, PDA で 2.8 倍, ノート PC で 1.6 倍程度までバッテリー持続時間を延長できることを確認した. また, 実用上十分な精度で, 再生希望時間

図 6 提案手法の使用によるバッテリー持続時間の変化 (PDA)
Fig. 6 Actual battery lives when using proposed method on the PDA.図 7 提案手法の使用によるバッテリー持続時間の変化 (ノート PC)
Fig. 7 Actual battery lives when using proposed method on the laptop PC.

を満足できるようにバッテリー持続時間を制御できることを確認した.

本論文では, 無線 LAN を使用する環境を想定したが, そのほかにも第三世代携帯電話を用いる場合が考えられる. 携帯電話では現在位置に応じて電波の出力が変動するため, 電波の出力の変動による電力消費量の変化を予測する, あるいは, 各時間帯での平均出力を見積もることで提案システムを適用可能であると考えられる. また, 本論文では動画のデコード処理に Xscale, Pentium などの汎用プロセッサを使用することを想定して電力消費モデルを構築した. 一方, DSP やハードウェアアクセラレータを用いた場合には, 特定の処理が高速化および低消費電力化されるが, 消費電力が画素数やビットレートに対して線形に増加するという性質は変わらないと考えられるため, 提案モデルにおける定数値の変更で対応できると考えられる.

本論文では, トランスコードプロキシと携帯端末間の帯域の変動については考慮しなかった. このような

問題に対しては、変換後の動画パラメータ値をストリーミング再生開始時の1回のみ行うのではなく、帯域の変動を常時監視しておき、実バッテリー残量と予測バッテリー残量との差がある閾値を超えると、動画パラメータ値を再計算し、再計算後のストリームを受信、再生することにより対応できると考えられる。そのための機構の実装および評価は今後の課題である。

参 考 文 献

- 1) Simunic, T., Benini, L., Acquaviva, A., Glynn, P. and De Micheli, G.: Dynamic voltage scaling for portable systems, *Proc. 38th Design Automation Conf. (DAC2001)*, pp.524-529 (2001).
- 2) Pering, T., Burd, T. and Brodersen, R.: Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System, *Proc. Driven Microarchitecture Workshop*, pp.107-112 (1998).
- 3) Lahiri, K., Raghunathan, A. and Dey, S.: Battery-Efficient Architecture for an 802.11 MAC Processor, *Proc. 2002 IEEE Int'l Conf. on Communications (ICC2002)*, pp.669-674 (2002).
- 4) Yip, K.W. and Ng, T.S.: Fast Power Control, Transmit Power Reduction and Multimedia Communications over WLANs, *Proc. 1st Int'l Conf. on Information Technology and Applications (ICITA2002)* (2002).
- 5) Chang, J.H. and Tassiulas, L.: Energy Conserving Routing in Wireless Ad-hoc Networks, *Proc. IEEE INFOCOM 2000*, Vol.1, pp.22-31 (2000).
- 6) Xu, R., Li, Z., Wang, C. and Ni, P.: Impact of Data Compression on Energy Consumption of Wireless-Networked Handheld Devices, *Proc. 23rd Int'l. Conf. on Distributed Computing Systems (ICDCS'03)*, pp.302-311 (2003).
- 7) Agrawal, P., Chen, J.C., Kishore, S., Ramanathan, P. and Sivalingam, K.: Battery power sensitive video processing in wireless networks, *Proc. IEEE PIMRC'98*, Vol.1, pp.116-120 (1998).
- 8) Thomas, S.: MPEG Digital Video-Coding Standards, *IEEE Signal Processing Magazine*, Vol.14, No.5, pp.82-100 (1997).
- 9) The Berkeley MPEG Player.
<http://bmrc.berkeley.edu/research/mpeg/>
- 10) MJPEG Tools. <http://mjpeg.sourceforge.net/>
- 11) NOFLUSHD - An idle-disk daemon.

<http://noflushd.sourceforge.net/>

(平成 16 年 4 月 21 日受付)

(平成 17 年 2 月 1 日採録)



玉井 森彦 (学生会員)

2002 年岡山県立大学情報工学部情報システム工学科卒業。2004 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。現在、同大学院博士後期課程在学中。マルチメディア通信システム、分散処理方式に興味を持つ。



安本 慶一 (正会員)

1991 年大阪大学基礎工学部情報工学科卒業。1995 年同大学大学院博士後期課程退学後、滋賀大学経済学部助手。1997 年モンテリオール大学客員研究員。2002 年より奈良先端科学技術大学院大学情報科学研究科助教授。分散システム、マルチメディア通信システムに関する研究に従事。IEEE/CS 会員。



柴田 直樹 (正会員)

2001 年大阪大学大学院基礎工学部研究科情報数理系専攻博士後期課程修了。2001 年より奈良先端科学技術大学院大学情報科学研究科助手、2003 年より滋賀大学経済学部助教授。分散システム、マルチメディア通信システム、ITS、遺伝的アルゴリズム等の研究に従事。



伊藤 実 (正会員)

1979 年大阪大学大学院基礎工学部研究科博士前期課程修了。1979 年より大阪大学基礎工学部助手。1986 年より大阪大学基礎工学部講師。1989 年より大阪大学基礎工学部助教授。1993 年 4 月より奈良先端科学技術大学院大学情報科学研究科教授。データベース理論、オブジェクト指向データベースのアプリケーション、遺伝的アルゴリズム等の研究に従事。電子情報通信学会、ACM、IEEE 各会員。