

シーン適応な高速動画像分散トランスコーディング

三部 靖夫[†] 大上 貴充[†] 渡辺 真太郎[†]
中村 太一^{††} 若宮 直紀^{†††}

シーンチェンジを境界とするように分割して、符号方式や符号レートを変換するシーン適応な動画像の分散トランスコーディングについて提案する。シーン適応にトランスコードすることで、分割点で画質劣化が生じた場合でもこの劣化が知覚されにくくなる。このことにより、詳細な符号化レート制御が行えないような様々な符号方式への変換も、分散化にともなう画質劣化を生じずに高速に行うことが可能になる。分散化の方式として固定長割当てと可変長最適割当ての2つの割当てに基づいてシーン適応に分散化する方式について検討し、分割数を少なくした可変長割当てに基づく方式が有利であることを示す。このシーン適応分散トランスコーディングは、シーンチェンジ間隔が長い映像ほど処理時間は長くなるが、この間隔が比較的長いドキュメンタリ映像のデータを用いたシミュレーション実験から、10台のPCで分散処理することにより処理時間をおよそ1/5程度に短縮できることを示す。

Scene Adaptive Distributed Video Transcoding

YASUO SAMBE,[†] TAKAMITSU OHGAMI,[†] SHINTARO WATANABE,[†]
TAICHI NAKAMURA^{††} and NAOKI WAKAMIYA^{†††}

This paper presents scene-adaptive distributed video transcoding, which alters an original coded video file into various video coding formats and rates with employing abrupt scene change detection. The transcoding system divides the video file into small segments along the time axis and transcodes them in parallel, in order to decrease the total transcoding time. Parallel transcoding along time axis without video coding rate-control among segments usually suffers from quality degradation around the segmented cut points in the transcoded video, because of a lack of information such as the coding complexity of the previous segment. Scene adaptive transcoding can decrease temporal video quality degradation perceived by human observers. This provides more flexibility in terms of video encoder used in the transcoder and makes possible to change the video file into various kinds of formats. We investigate how segment transcoding should be distributed to obtain the shortest total transcoding time. Simulation results show that the scene-adaptive distributed transcoding on 10 PCs can decrease the transcoding time by a factor of about 5.

1. はじめに

多様なブロードバンドアクセスネットワークや、CD-ROM や DVD といった大容量蓄積媒体の普及により、一度符号化された動画像符号ファイルを他の符号化レートや符号方式へ変換して配信したり、あるいは、複数種類変換されたファイルを蓄積格納しておき、ユー

ザが再生時に復号ソフト、回線速度、および端末の再生能力に応じた動画像を選択可能なようにしたりするという要求が増えてきている。

このような要求にこたえるために、近年、符号化レートや符号化方式を高速高品質に変換する研究がさかんになってきている。たとえば、文献 1), 2) では MPEG-2 から MPEG-2、あるいは MPEG-4 ソフトウェアでリアルタイムにトランスコード可能な方式の提案がされており、文献 3) では H.261/H.263 へ高速にトランスコードする方式の提案がなされている。これらは、動きベクトルや DCT 係数の再利用による処理量の削減や、バッファ遅延を考慮した制御を行うことにより高速で高品質なトランスコーディングを実現している。

[†] 株式会社 NTT データ技術開発本部
NTT DATA Corporation

^{††} 東京工科大学コンピュータサイエンス学部
School of Computer Science, Tokyo University of Technology

^{†††} 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

一方、多種類同時に高速符号変換するための方式として、筆者らは、時間軸方向に分割して分散処理する分散トランスコーディングシステムの提案を行っている^{4),5)}。一般に、MPEGなどの動画像符号化では相対的に画像の冗長性が高いことを利用して圧縮し、さらに、それまでの符号化処理で得られた実符号量と目標符号量との差や複雑度といった統計情報を基に、引き続き画像の符号化の困難さを推定しながら画質が一定になるよう制御している。したがって、単純に、時間軸方向に分割して分割点前の符号制御情報を使わずに独立して符号化すると、分割点で画質劣化を生じてしまう。このため文献4)では、変換前のソース符号情報から分割点での符号制御情報を推定したレート制御を行うことで、分割点で画品質が不連続にならないようにする方式を提案している。

しかしながら、符号化アルゴリズムの詳細が公開されていなかったり、その制御手段が提供されていないような符号化方式への変換では、分割点で画質連続性を保証するといった符号制御を行うことができない。そこで本稿では、分割点がシーンの切替え箇所のみになるようにするシーン適応な分散トランスコーディング方式を提案する。このようにすることで、分割点で画品質が不連続になったとしても、この劣化が知覚されにくくなることを利用する。

シーンチェンジで短時間の画質劣化が生じた場合の視覚特性は明らかでないが、通常映像シーン中における短時間のランダム雑音や偽輪郭に対しては、劣化低減の効果のあることが知られている⁶⁾。シーンチェンジのように直前の画像と無関係な箇所であれば、さらに、この劣化マスキング効果は高くなると想定される。また、MPEGなどの動き補償を行う符号化では、このマスキング効果を利用することと同時に、直前画像との差異が大きい画像で動き補償が有効に機能しないため、イントラフレームをシーン適応に入れた高効率な符号制御方式が検討されている^{7),8)}。さらに、シーンの切れ目でフレーム内符号化画像が入っていると、ここでシーンを切り出しやすくなるなど動画像編集が容易になるというメリットがある。これらの理由により、最近の実用に供せられている動画像エンコーダはほとんど、シーンの切替り画像を自動的に検出し、この画像をイントラ符号化する機能を有している。本稿で対象にする動画像符号化はこのようなエンコーダによって再符号が可能なものであるとする。

以下、2章でシーン適応な分散トランスコード方式の概要について述べ、3章でこの方式による性能見積りを行う。4章で、この性能見積りをシミュレーション

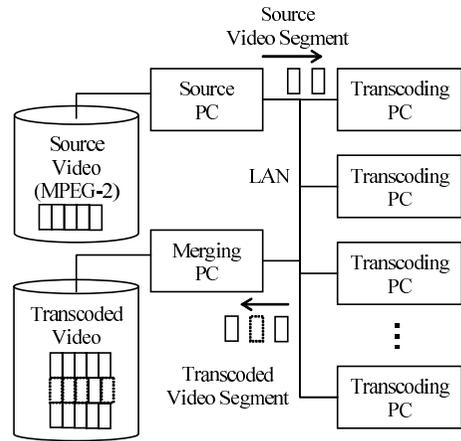


図1 分散ビデオトランスコードシステムの構成
Fig. 1 Configuration of distributed video transcoding.

によって検証し考察を述べ、最後に5章でまとめと今後の課題について述べる。

2. シーン適応分散ビデオトランスコードの概要

2.1 分散ビデオトランスコードの概要

分散トランスコードシステムは図1に示すようにソースPC、マージPC、および複数のトランスコードPCとを高速LANで接続した構成をとる。

ソースPCは動画像と音声が多重化されたMPEG-2 PS (Program Stream) の映像ファイルを格納している。ソースPCは、このMPEG-2 PS ファイルの動画像と音声とを分離し、利用できるトランスコードPCの数に応じて、動画像ファイルを適当な長さに分割してトランスコードPCへ転送する。音声の変換は高速に処理できるため、分割せずに1つのトランスコードPCで実行する。以下、分散処理は動画像の変換のみを対象とする。動画像を時間軸方向に分割したファイルを動画像セグメントと呼ぶ。動画像セグメントは、独立して処理をしやすいようにGOP (Group of Pictures) を単位として切り出すようにする。さらに、ソースPCは動画像セグメントを送る際に、このセグメントを再符号化する方式や、解像度、および目標符号化速度などの変換パラメータを送る。

トランスコードPCでは、送られてきた動画像セグメントを指定された符号化方式や符号化速度に変換して、マージPCへ転送する。トランスコードPCの機能構成を図2に示す。セグメントの受信が完了した後、復号器 (Decoder) がこのセグメントをデータを順次読み出して復号化を行い画像を1枚ずつFilterへ送る。復号画像は、Filterでパラメータ指定された

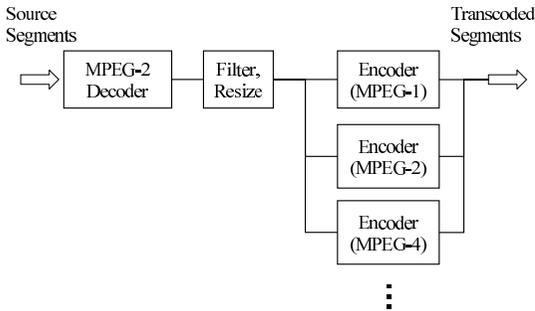


図2 トランスコード PC の構成
Fig. 2 Block diagram of transcoding PC.

解像度に変換された後、符号化器 (Encoder) へ順次送られ再符号化される。このようにしてトランスコード PC は、受け取ったセグメントに対して、セグメント内に記述されているパラメータで指定される解像度変換や、符号化方式、および符号化速度の数だけ複数種類トランスコードした動画セグメントを作成してマージ PC へ送る。このとき、復号処理は 1 回しか行わないため、変換種類が多くなるほど効率が増す構造となっている。

最後にマージ PC で、符号変換された動画セグメント集めて結合し、これと音声の符号変換済みファイルとを多重化する。

2.2 シーンチェンジ検出

シーンチェンジを検出する方法としては、従来から、映像編集や検索のための代表画像を抽出することなどを目的として各種提案がなされている^{9)~11)}。しかしながら、これらの方法を分散トランスコードのセグメンテーションに用いることは適当ではない。なぜなら、使用する符号化器のシーンチェンジ検出のアルゴリズムやこのアルゴリズムで用いられる閾値が一般には公開されていないために、この符号化器と同じ位置でシーンチェンジを検出できるとは限らないためである。セグメントの最初の画像は、PC ごとに独立して再符号化することから、動き予測するための直前参照画像がないためにフレーム内符号化される。したがって、符号化器の判定とは異なる画像をシーンチェンジと判定してここでセグメントを切り出すと、全体としてフレーム内符号化画像が増えてしまうことになる。フレーム内符号化画像は動き予測を行わないために、通常、動き補償符号化画像より数倍多い符号量が割り当てられる。余分なフレーム内符号化画像が増えると、他の画像の符号量割当てを下げたことになり、長い期間にわたって画質劣化が生じる。したがって符号化器と同じ精度でシーンチェンジ検出を行って分割することが必要である。

そこで、一度トランスコードした後の符号化データから、符号化器がシーンチェンジと見みなしたフレームを推定することによりシーンチェンジ検出を行うことにする。そして、2.3 節で述べるようにこのシーンチェンジ画像を新たな境界とするように、各セグメントのシーンチェンジまでの MPEG-2 データをトランスコード PC 間で送受信し、再符号化を行う。このため、各セグメントの最初のシーンチェンジまでのソース動画は 2 度符号変換されることになる。

シーンチェンジ画像の推定には、トランスコード PC の符号化器がシーンの切替りで自動的にフレーム内符号画像を挿入する機能を利用する。具体的には、相続くフレーム内符号化画像の間隔が再符号化時に指定した間隔値と異なる場合に、この画像をシーンチェンジ点であるとする。この方式では、指定した間隔と同じところにフレーム内符号画像が出現した場合には、正確な判定はできない。この場合には、安全側に立ってシーンチェンジ点ではないと判定する。仮にシーンチェンジであっても、再符号時に符号化器でシーンチェンジであると判定され、イントラフレームが自動的に挿入されるためである。

このようにして測定したシーンチェンジ間隔の分布を図 3、図 4、図 5、図 6 に示す。この測定は、Microsoft 社の Windows Media 9 を使って、フレーム内符号間隔を 100 秒と指定して符号化した出力から、実際に生成されたフレーム内符号間隔を求めることで得たものである。Windows Media ではフレーム内符号した画像をキーフレームと呼ぶ。図 3~図 6 から生成されたフレーム内符号間隔の平均がせいぜい 8 秒程度であり、指定した 100 秒に比べてかなり小さいことから、生成されたフレーム内符号画像のほぼすべてが、符号化器がシーンチェンジと判定して生成したものと見える。測定に用いた映像はすべて 1 時間の長さで、比較的シーンチェンジの多いアニメ映像から、シーンチェンジの少ない自然ドキュメンタリー解説映像までの 4 種類 (Animation, Variety, Football, Documentary) を選んだ。それぞれの図には、サンプルから算出した平均値に従う指数分布をあわせてプロットしている。一般的に、シーンチェンジ間隔がどのような分布に従うかは明らかでないが、この実験結果からは 1~2 秒程度と短い時間にかかなり偏った分布であり、指数分布で近似できることが分かる。

1 秒以下のシーン間隔が多いのは、デゾルブやワイプ、あるいは画像全体の動きが比較的激しいところで、連続してイントラフレームが入っているためであり、通常の意味でのシーンチェンジとは異なる。しかしな

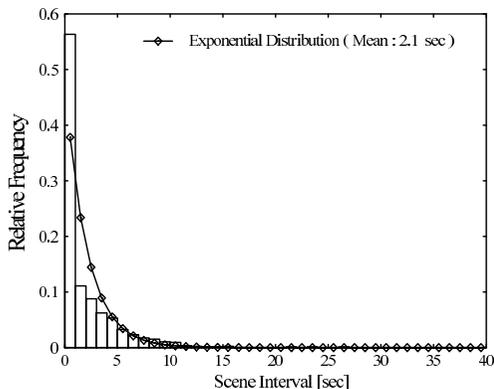


図 3 アニメ映像のシーンチェンジ間隔の測定結果
Fig. 3 Scene interval of animation video.

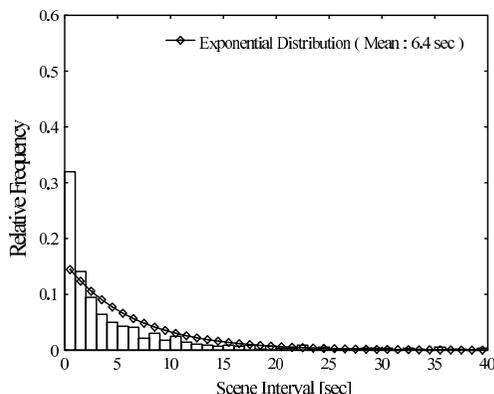


図 5 サッカー中継映像のシーンチェンジ間隔の測定結果
Fig. 5 Scene interval of football game video.

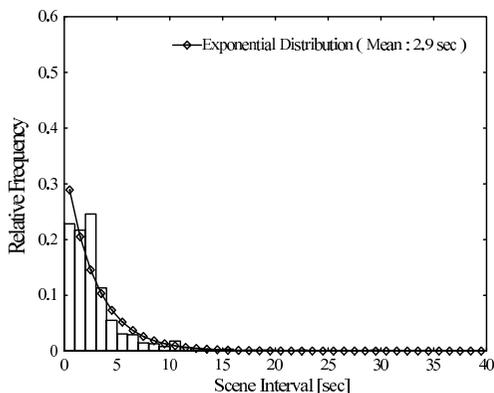


図 4 パラエティ番組映像のシーンチェンジ間隔の測定結果
Fig. 4 Scene interval of variety show video.

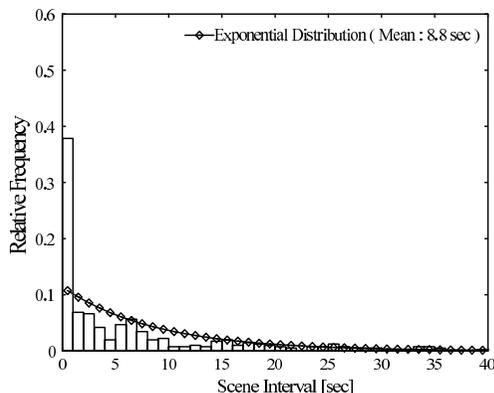


図 6 自然ドキュメンタリー映像のシーンチェンジ間隔の測定結果
Fig. 6 Scene interval of documentary video.

がら、このような箇所では分割化にともなう画質劣化が生じても知覚されにくいことと、このような符号化器を用いて分割せずに再符号化したのと同じ符号出力が得られるようにするために、ここをシーンチェンジであると判定する。

2.3 シーンチェンジ処理

動画セグメント内でシーンチェンジを検出したトランスコード PC は、最初のシーンチェンジまでのデータを、直前セグメントをトランスコードしている PC へ再転送する。この再転送データを受け取った PC は、現在、トランスコードしているセグメントに続けて再転送されてきたデータの符号変換を行う。このように次のシーンチェンジまで続けて再符号化することで、画品質が不連続になりうる箇所はシーンチェンジ直後のみになる。

このフローを図 7 の (a) に示す。セグメント内にシーンチェンジを見つけたトランスコード PC (P_{i-1}) は最初のシーンチェンジまでのソース符号データを、

直前のセグメントをトランスコードしている PC (P_i) に送る。 P_i では、現在のセグメントで見つかった最初のシーンチェンジ画像から P_{i-1} より送られてきたデータの最後の画像までを変換して、マージ PC へ送る。

セグメント内にシーンチェンジがなかった場合は、図 7 の (b) に示すように、このセグメントすべてを直前の PC へ再転送するとともに、直後の PC から再転送されてきたシーンチェンジまでのデータを、再度、前の PC (P_{i+1}) へ送る。

これらのデータ再転送をソース PC から行わずにトランスコード PC から行うのは、このシステムの性能ボトルネックがソース PC の MPEG-2 PS ファイルの分離速度にあるため⁴⁾、再転送負荷をトランスコード PC 側へ移して分散化し台数スケラビリティを向上させるためである。

2.4 PC への割当て方法

セグメントの割当て法としては、実装が容易な固定

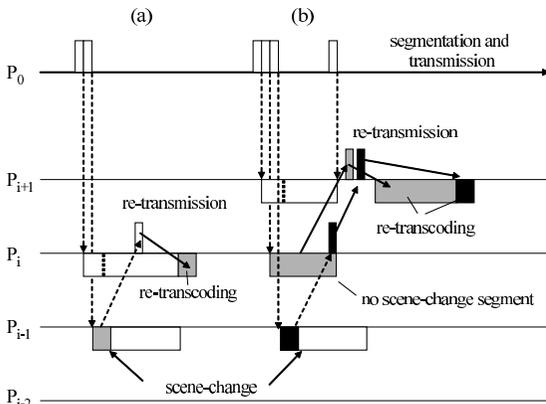


図 7 シーンチェンジを考慮した分散符号変換のフロー

Fig. 7 Process flow of scene-adaptive distributed video transcoding.

長割当て (以下 FL) 方式⁴⁾ と, Divisible Load Theory^{12),13)} に基づく可変長割当て (以下 DL) 方式の 2 つを検討し, この両者の比較を行う. DL 方式は, ジョブが任意の量に分割可能で, かつ, その分割単位の処理順序に制約がない場合に適用できる可変長の最適割当て方式であり, 分割数が少なくても全体の処理時間が最小値に近くなるという特徴がある. 各方式とも, ソース PC が前節で述べたシーンチェンジ処理を行わない場合の最小処理時間となる割当てを行い, これと並行して, 各トランスコード PC 間でシーンチェンジ処理にともなう再トランスコードを行う.

まず, 多重分割の DL 方式 (Multi-Installment Load Distribution) について文献 12) の記法に従って説明する.

図 8 に分散トランスコードシステムのモデルを示す. トランスコード PC の番号と繰り返し転送の番号は, 文献 12) と同じく, 後の解析式で添字の扱いが煩雑なものにならないように一般的なものと逆の順序でつけており, P_m から P_1 へ順番に, n 番目から 1 番目の転送を行うものとしている. また, ここで扱うトランスコード PC は, セグメントデータの送受信処理とトランスコード処理とが並列に実行可能なようにマルチスレッド化がなされているものとする. したがって, 文献 12) における通信用フロントエンドプロセッサがある場合のモデルを適用する.

ソース PC (P_0) から, 分離性能 zT_{cm} で MPEG-2 PS ファイルから動画情報分離されて動画セグメントが切り出されて送信され, トランスコード PC の P_m から順に P_1 までの転送を n 回繰り返す. ここで, z はこのソース PC の分離速度に反比例する定数で, T_{cm} はある仮想的な標準 PC でソース映像

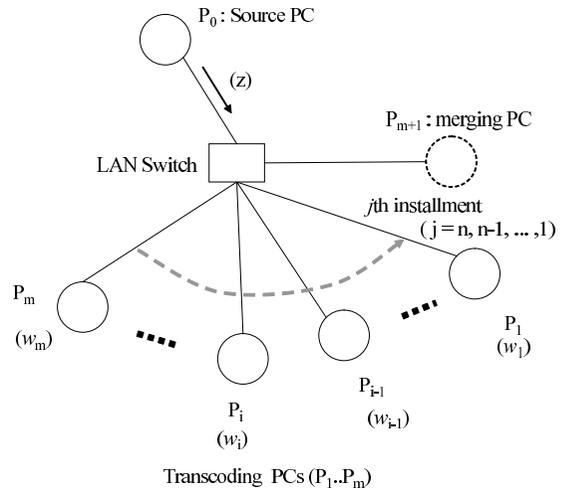


図 8 Divisible Load Theory に基づく分散符号変換システムのモデル

Fig. 8 System model for performance estimation.

ファイルを全部分離するのに要する時間であり, これらの積 zT_{cm} はソース PC で映像ファイルを全部分離する時間となる. 各トランスコード PC (P_i) についても, それぞれの処理速度に反比例する定数を w_i とし, ある仮想的な標準 PC で映像ファイルを全部トランスコードするのに要する時間を T_{cp} とすると, P_i でこのソース動画ファイルをトランスコードするのに要する時間は $w_i T_{cp}$ となる.

処理タイミングを図 9 に示す. ソース PC (P_0) は, 各トランスコード PC へ順にソース符号データを $\alpha_{i,j}$ の割合で分割したセグメントを転送する. ここで, i は PC の番号で, j は転送回数の番号である. トランスコードの終わったセグメントはマージ PC (P_{m+1}) へ送られ, 順次結合される. このマージ処理は主にタイムコードの振り直し処理であり, この速度はマージ PC のディスクの I/O 速度で決まる. 一般的にビデオトランスコードは数 Mbps 以上の高品質な動画ファイルから 1 桁から 2 桁程度小さい符号速度の映像ファイルへ変換する. したがって, トランスコード後のデータのマージ処理は分離処理やトランスコードに要する時間に比べて十分に小さい. そこでマージ時間を無視し, ソース PC から送る全セグメントの変換時間が最小になるように割当てを行うことにする.

再トランスコード処理を行わない場合の変換時間が最小になるのは, P_1 から P_m の処理時間がすべて同じになるときである¹²⁾. すべてのトランスコード PC の処理時間が同じになるには, まず最後の転送 ($j = 1$) において,

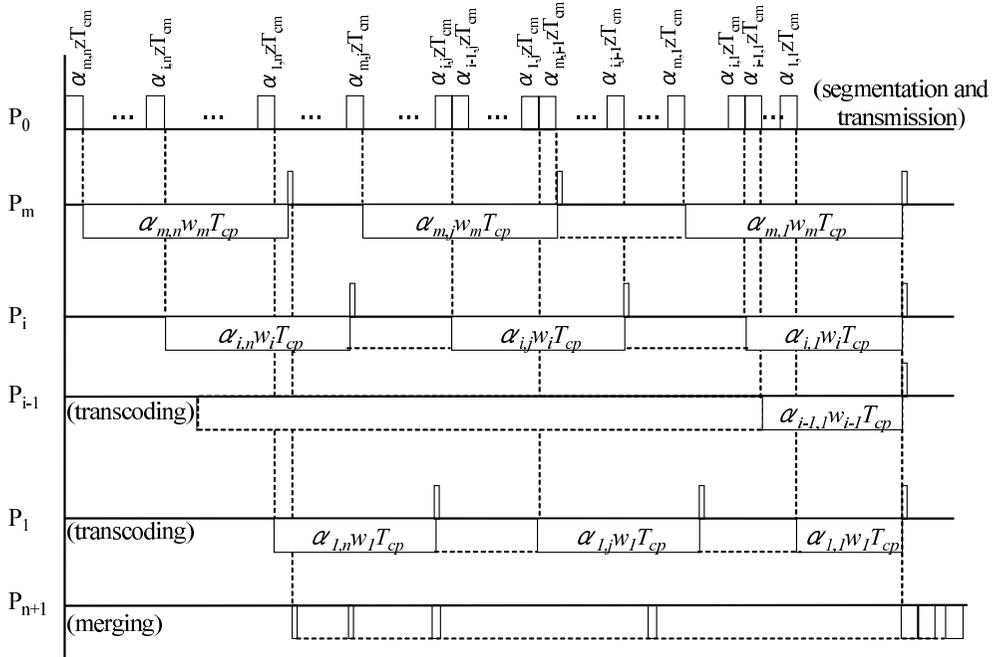


図 9 Divisible Load Theory に基づく分散符号変換のタイミング
Fig. 9 Timing diagram based on Divisible Load Theory.

$$\alpha_{i,1} w_i T_{cp} = \alpha_{i-1,1} (z T_{cm} + w_{i-1} T_{cp}),$$

$$i = 2, 3, \dots, m \quad (1)$$

となり、それ以前の転送 ($j = 2, \dots, n$) において

$$\alpha_{i,j} = \left\{ \sum_{k=1}^{i-1} \alpha_{k,j} z + \sum_{k=i}^m \alpha_{k,j-1} z \right\} \left\{ \frac{\delta}{w_i} \right\},$$

$$i = 1, \dots, m; \text{ and } j = 2, \dots, n \quad (2)$$

となる必要がある。ここで $\delta = T_{cm}/T_{cp}$ である。式 (1) を変形すると、

$$\alpha_{i,1} = \alpha_{1,1} \left\{ \prod_{k=1}^{i-1} \frac{w_k + z\delta}{w_{k+1}} \right\} \quad (3)$$

である。また、 $\alpha_{i,j}$ は分割の割合であるので、

$$\alpha_{1,1} + \dots + \alpha_{m,n} = 1 \quad (4)$$

が成り立つ。これらから $\alpha_{i,j}$ について解けば、トランスコード時間が最小となる割当てが求まる。このとき、全体のトランスコード時間 T_{DL} は、

$$T_{DL} = \sum_{j=1}^n \sum_{i=1}^m \alpha_{i,j} z T_{cm} + \alpha_{1,1} w_1 T_{cp} \quad (5)$$

となる。

以上のようにして $m = 10, n = 3, w_i = 2.0$ のときに求められる $\alpha_{m,n}$ で割り当てたときのタイミング図を図 10 に示す。この図から、最初の割当て $\alpha_{10,3}$ が最も小さく、順次大きく割り当てられることが分

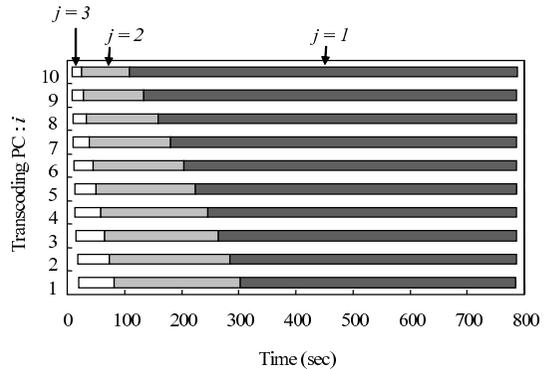


図 10 DL 方式での割当て例
Fig. 10 An example of timing diagram based on DL method.

る。分割数を大きくすると、この最小割当て量が急速に小さくなる。図 11 に 1 時間の動画をトランスコードする場合に、この最小割当て量がどのようになるか計算した結果を示す。通常の MPEG-2 符号化では 1GOP は 0.5 秒程度とされる。1GOP より小さく割り当てると分割されたセグメントは独立に復号できなくなってしまう。このため、これ以下に分割することはできない。図 11 から、PC10 台のときで $n = 5$ 、PC5 台のときで $n = 4$ 以上にすると、最小割当て量が 0.5 秒より小さくなることが分かる。このように DL 方式では、最小割当て量の制約から分割数を大き

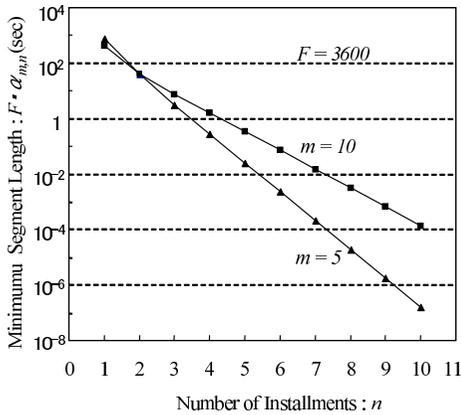


図 11 DL 方式での最小割当て量

Fig. 11 Minimum segment allocation of DL method.

くすることはできない。

一方、すべての転送回で等しい長さのセグメントをラウンドロビンで送る固定長方式 (FL) の場合の処理時間 T_{FL} は、すべてのトランスコード PC の性能が同じで $w_i = w$ であるとする、初めのセグメントが最後に送られる PC (PC_m) へのセグメント転送時間遅れと、この PC でのトランスコード処理時間を加えることにより⁴⁾,

$$T_{FL} = \frac{zT_{cm}}{n} + \frac{wT_{cp}}{m} + nT_{OH} \quad (6)$$

となる。ここで T_{OH} はセグメントごとに要する画質調整のための数 GOP 分の重複トランスコード時間と再符号レート制御処理時間である。

3. 性能見積り

全体の処理時間は、前節で述べたソース PC から割り当てられたセグメントをすべて処理する時間に、シーンチェンジにともなう再トランスコードに要する時間を加えたものになる。この再トランスコード量はシーンチェンジ間隔と分割回数に依存し、さらに各 PC ごとに大きく異なる。したがって、全処理時間に加える再トランスコード時間は、 m 台ある PC の中で最もこれに要する時間が大きいものになる。そこで、まず、1 つの PC における再トランスコード量の確率分布を求め、次に、これが m 台あるときの最大値を求めることにより、全処理時間を求める。

3.1 再トランスコード量の見積り

1 つの動画ファイルにおいて、シーンチェンジ間隔は 2.2 節で述べたように平均 a 秒の指数分布に従うと仮定し、すべてのフレームで同じ分布が成り立つものとする。すなわち、任意の分割点から次のシーン

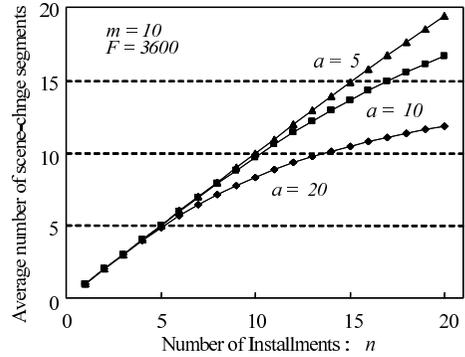


図 12 シーンチェンジが存在するセグメントの数

Fig. 12 Average number of video segments having scene-changes.

チェンジまでの間隔が平均 a 秒の指数分布に従うと仮定する。ソース PC から受け取ったセグメント内にシーンチェンジがあると、このセグメントの後ろの映像を処理しているトランスコード PC から再転送されるソース符号データを続けてトランスコードすることになる。

FL 方式の場合には、トランスコード PC に長さ d のセグメントが n 個渡されたとき、この n 個のうち、1 つ以上のシーンチェンジを含むセグメントの個数 b は

$$b = n\{1 - \exp(-\frac{d}{a})\} \quad (7)$$

となる。1 時間 ($F = 3,600$ 秒) の動画像に対する、この b の計算結果を図 12 に示す。PC の台数 m に対してセグメントの長さは $d = F/mn$ となる。分割数 n が大きくなってセグメント長 d がシーンチェンジ間隔 a に比べて 2 倍程度以下 ($a = 10$ のとき $n \approx 9$, $a = 20$ のとき $n \approx 18$) になってくると、シーンチェンジを含む確率が小さくなり b の伸びは小さくなる。DL 方式の場合は、式 (7) において n 倍となるのではなく、 PC_i ごとに異なり、各割当て回 j の割当て長 $\alpha_{i,j}F$ に対する確率を足し合わせたものになる。

あるトランスコード PC での再トランスコード量の総和 x は、シーンチェンジを含むセグメント b 個の指数分布の和となり、scale parameter が a で、shape parameter が b のガンマ分布になる¹⁴⁾。したがって、再トランスコード量の総和 x の確率密度関数 $f(x)$ は、

$$f(x) = \frac{(x/a)^{b-1} \exp(-x/a)}{a\Gamma(b)} \quad (8)$$

となる。

この再トランスコード量 x の平均は ab 、分散は a^2b となり、処理する動画像のシーンチェンジ間隔 a に対

して再トランスコード量が比例して大きくなり変動も大きくなる．また，分割数 n にも，ほぼ比例して大きくなり，変動も大きくなってしまふことから，分割数を大きくすることは全体の処理時間を増加させてしまふことが分かる．一方，分割数を少なくしてセグメント長を長くすると，各トランスコード PC への最初のセグメントデータの転送遅延が大きくなってしまふ．したがってこのデータ転送遅延と再動画画像変換量の間にはトレードオフがあり，全体の処理時間を最短にするためには，最適な分割数を決定する必要がある．

3.2 全処理時間の見積り

全体の処理時間は，2.4 節で述べたソース PC から割り当てられたセグメントをすべて処理する時間に，シーンチェンジにともなう再トランスコードに要する時間が最も大きい PC の再トランスコード処理時間を加えたものになる．前者は，FL 方式では式 (6)，DL 方式では式 (5) である．後者の， m 台のトランスコード PC のうち最大となる時間は，順序統計によって求めることができる．すなわち，各トランスコード PC(i) の再トランスコードの処理時間が確率密度関数 $f(x)$ に従う確率変数になるとすると，この $f(x)$ から m 回の試行で得られるサンプル X_1, X_2, \dots, X_m の中の最大値の確率密度関数 $g(x)$ と分布関数 $G(x)$ は，それぞれ

$$g(x) = m[F(x)]^{m-1}f(x) \tag{9}$$

$$G(x) = [F(x)]^m \tag{10}$$

となる¹⁵⁾．ここで $F(x)$ は $f(x)$ の分布関数である．

PC が 10 台あるときの，FL 方式における，再トランスコード量の最大値分布を図 13 と図 14 に示す．図 13 はシーンチェンジ間隔の短い映像 ($a = 5$) に対して，図 14 はシーンチェンジ間隔の長い映像 ($a = 20$) に対して計算したものである．これらの結果から，分割数 n を大きくすると再トランスコード量の最大値とその変動が大きくなってしまふことと，セグメント長がシーンチェンジ間隔と同程度となるところまで n を大きくするとこの再トランスコード量の最大値の変化は小さくなっていくことが分かる．

DL 方式のときは，各 PC と各転送回ごとにセグメント長が異なり，各 PC の再トランスコード量の総和 x の分布の式 (8) において shape parameter b が異なるため，式 (9) で最大値の分布を求めることはできない．しかしながら，ここでは近似的にすべて PC の割当てが PC₁ に等しいとして式 (9) で求めることとする．DL 方式では，図 11 から推測されるように， n が 3 程度より大きくしても，きわめて短いセグメントが増えるだけで，この短いセグメントにシーンチェン

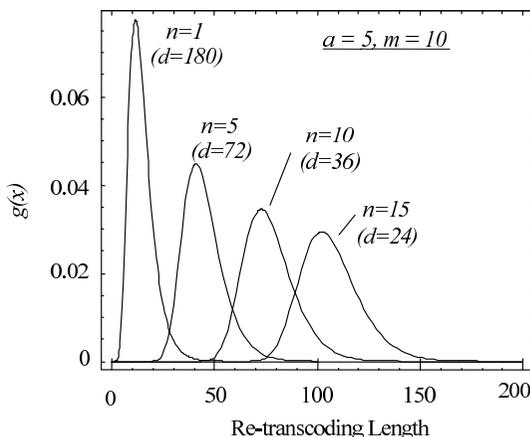


図 13 再動画画像変換量の最大値分布のシーンチェンジ平均間隔依存性 1

Fig. 13 Maximum re-transcoded video length distribution : $a = 5$.

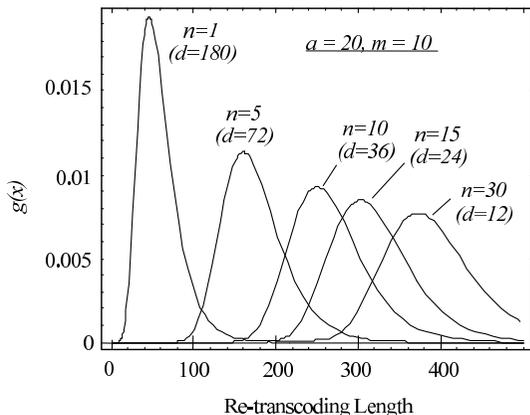


図 14 再動画画像変換量の最大値分布のシーンチェンジ平均間隔依存性 2

Fig. 14 Maximum re-transcoded video length distribution : $a = 20$.

ジが存在する確率はほとんどないことから，再トランスコード量は大きく変化しない．

各 PC の性能 w_i が等しく $w_i = w$ であるとするとき，この再トランスコード量の最大値の分布から求める平均値と w の積をとることで m 台で n 分割したときの再トランスコードに要する平均時間が求まる．これと，再トランスコードをしないときの処理時間とを足し合わせることで，全体の処理時間の平均値を見積もることができる．後者は，FL 方式では式 (6) であり，方式では，式 (5) である．このようにして求めた全処理時間の数値計算結果を図 15 と図 16 に示す．この計算で使用したパラメータは表 1 のとおりである．トランスコード PC1 台でトランスコードす

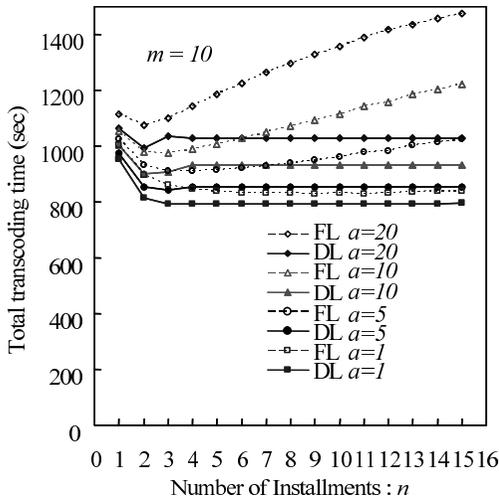


図 15 全処理時間の数値計算結果：転送繰り返し回数依存性
Fig. 15 Analytical performance estimation.

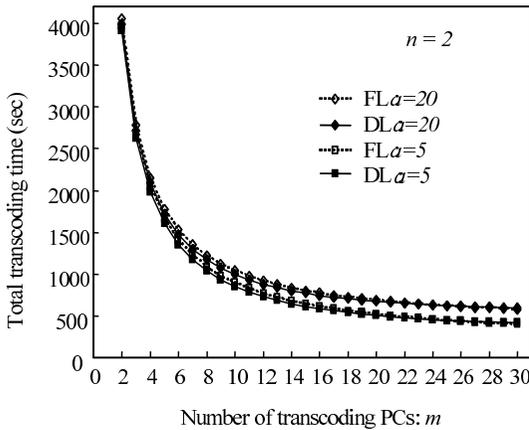


図 16 全処理時間の数値計算結果：台数効果
Fig. 16 Analytical performance estimation.

表 1 処理時間計算に使用したパラメータ
Table 1 Parameter for performance estimation.

ソース符号化ファイル	
ファイル長 F	3,600 sec
符号レート R_s	8.25 Mbps
平均シーンチェンジ間隔 a	1, 5, 10, 20 sec
ソース PC	
de-multiplexing 速度 z, T_{cm}	1,288 sec (100 Mbps に相当)
分割数 n	1, 2, 3, ..., 10
トランスコード PC	
動画像変換速度 w_i, T_{cp}	2.15, 3,600 sec
台数 m	10
セグメントオーバーヘッド T_{OH}	0.1 sec

ときの処理時間は wT_{cp} であり、この計算パラメータでは 7,740 秒となる。図 15 は PC を 10 台使用したときの、FL 方式と DL 方式のそれぞれについて、

シーンチェンジ間隔 a と分割数 n を変えて計算した結果である。DL 方式については、 $n = 5$ 以上は最小割当てが 0.5 秒以下となって、通常の MPEG-2 符号ファイルでは割当て不可能であるが、傾向をみるために $n > 5$ の計算結果も示してある。この図から、シーンチェンジ間隔が大きくなるにつれて処理時間が大きくなるのが分かる。また DL 方式の方が FL 方式よりも処理時間が短いことが分かる。シーンチェンジ間隔 a が 1 秒程度と小さいときは分割数を大きくすると処理時間が若干小さくなるものの、 a がそれより大きい動画の場合は、分割数を 3 以上にすると FL 方式では処理時間が増加してしまう。DL 方式では分割数 n はおおむね 2 のときに最小となり、これより n を大きくしても処理時間は変わらない。DL 方式で n を大きくしても処理時間が変わらないのは、 n を増加してもシーンチェンジ間隔 a より短いセグメントが増えるだけで、再トランスコード量が増加しないためである。

図 16 は、この分割数 n を 2 と固定したときの、PC の台数 m による処理時間削減の効果を示している。PC の台数 m が 15 台程度までの間であれば DL 方式の方が 5~10% ほど処理時間が短くなるが、これ以上の台数であればどちらも同じ値に近づくことが分かる。

各 PC の性能が大きく異なる場合の割当てについては、式 (9) をそのまま適用することができない。この場合の解析的な求め方は今後の課題である。

4. シミュレーション実験結果と考察

前章での見積りと最適な分割数を検証するため、まず指数分布に従うシーンチェンジ間隔データを用いてシミュレーションを行った。

シミュレーションに使用したモデルを図 17 に示す。このシミュレーションでは、解析値の導出で無視した LAN の輻輳やマージ PC でのマージ処理時間も加えたモデルとしている。ソース PC から切り出されたセグメントは LAN Switch を通ってトランスコード PC へ順次転送され、トランスコードとシーンチェンジ検出を行う。シーンチェンジ処理にともなう再転送データは再び LAN Switch を通って、2.3 節で述べたように前の PC へ送られる。LAN Switch の交換能力は 500 Mbps とし、転送に要する時間は指数分布に従うものとしてモデル化した。また、マージ PC は、60 Mbps のマージ速度で処理する能力があったとした。これら以外のパラメータは、表 1 と同じである。

固定長セグメント割当ての場合の結果を図 18 に、Divisible Load Theory に基づく割当ての場合の結果

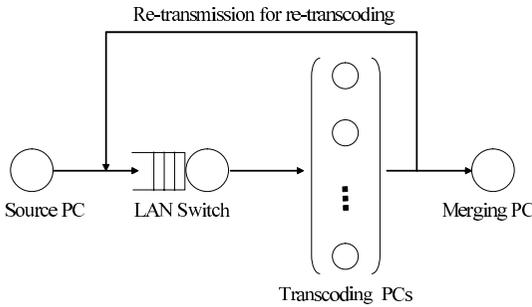


図 17 シミュレーションモデル
Fig. 17 Simulation model.

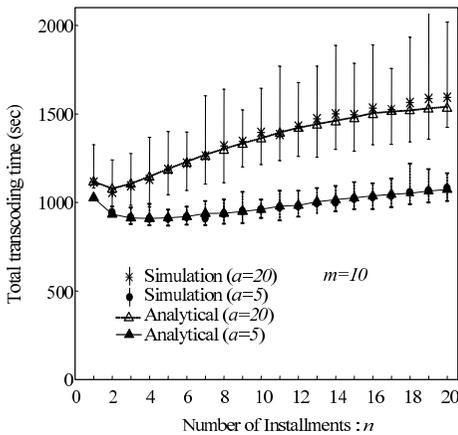


図 18 FL 方式の速度性能シミュレーション結果
Fig. 18 Performance simulation results of FL method.

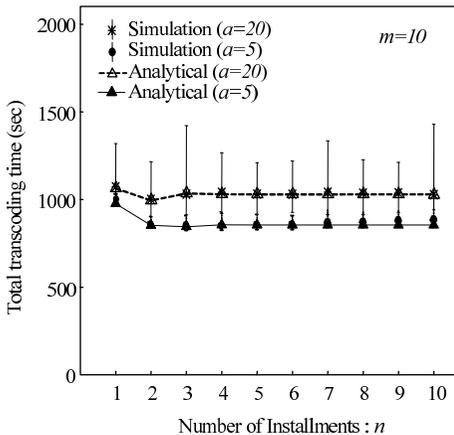


図 19 DL 方式の速度性能シミュレーション結果
Fig. 19 Performance simulation results of DL method.

を 図 19 に 示 す . い ず れ も 分 割 数 n を 変 え た と き に 100 回 の 試 行 で 得 ら れ た 最 大 値 , 最 小 値 お よ び 平 均 値 を 示 し て お り , 3.2 節 で 求 め た 平 均 処 理 時 間 も あ わ せ て 載 せ て い る . こ れ ら の 結 果 か ら , 解 析 的 に 求 め た 平 均 処 理 時 間 が シ ム ュ レ ー シ ョ ン で 得 ら れ た 平 均 処 理 時 間 と よ く 合 っ て い る こ と , お よ び チ ェ ン ジ 間 隔 a が 大

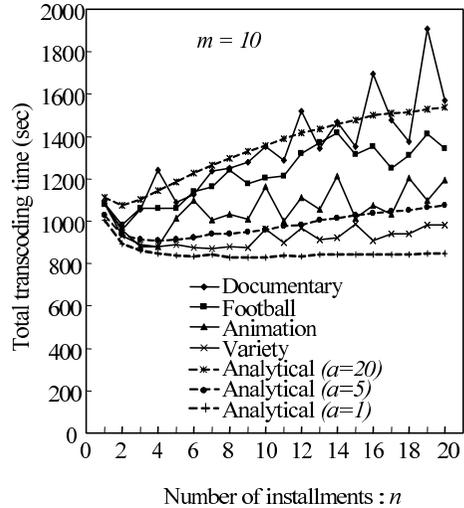


図 20 シーンチェンジログを用いた FL 方式の速度性能シミュレーション結果
Fig. 20 Performance simulation results of FL method using scene interval data.

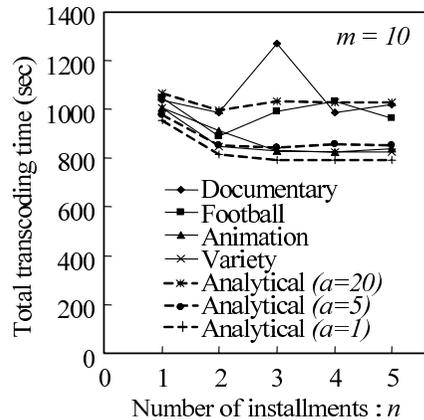


図 21 シーンチェンジログを用いた DL 方式の速度性能シミュレーション結果
Fig. 21 Performance simulation results of DL method using scene interval data.

き くな る と , 3.1 節 で 述 べ た よ う に 処 理 時 間 と そ の 変 動 の 両 方 が 大 き くな る こ と が 分 か る . ま た , FL 方 式 で は 分 割 数 が 大 き くな る に つ れ て , 前 節 で 述 べ た よ う に 処 理 時 間 変 動 が 大 き くな る が , DL 方 式 で は , こ の 変 動 は 分 割 数 に は ほ と ん ど 依 存 し な い こ と が 分 か る .

次 に , 2.2 節 で 求 め た 動 画 像 の シ ーン チ ェ ン ジ 間 隔 を 使 っ て シ ム ュ レ ー シ ョ ン し た 結 果 を , 図 20 と 図 21 と に 示 す . 3.2 節 で 求 め た シ ーン チ ェ ン ジ 間 隔 が $a = 5$ と $a = 20$ の と き の 数 値 計 算 結 果 も あ わ せ て プ ロ ッ ト し て い る . こ の 結 果 か ら , シ ーン チ ェ ン ジ 間 隔 a が 2 秒 程 度 と 小 さ い Variety や Animation で は , 解 析 値 で 求 ま る 平 均 値 と 近 い も の , a が 大 き い Documen-

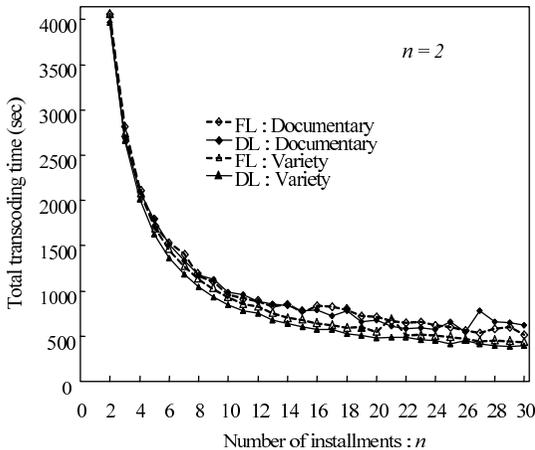


図 22 シミュレーションによる台数効果の比較

Fig. 22 Performance vs. number of transcoding PCs.

tary や Football (それぞれ 6.4 秒, 8.8 秒) では, n が 3 以上のとき, a が倍以上の 20 秒のときの解析値に近くなっていることが分かる.

これは, 図 6 から分かるように a が大きい動画では指数分布で見積もるよりも長いシーンチェンジ間隔の箇所が多かったためと考えられる. シーンチェンジ間隔の長い割合が指数分布より多くなってくると, n が大きいと, 解析値で予想されるものより性能の悪化を招く結果になることが分かる. このため, あらかじめ厳密にシーンチェンジ間隔の分布を知ることができないという条件では, 分割数を小さくして 2 としておくのがよいといえる. また, このように分割数が小さい場合は, FL 方式はデータ遅延時間が大きく, Documentary を除けば, DL 方式より 50~60 秒ほど大きな処理時間となっている.

次に $n = 2$ としたときの, PC 台数を増やしたときの効果を映像 Documentary と Variety を用いて調べた結果を図 22 に示す. それぞれの映像での処理時間は, 上記の分割数 n 依存性と同じく, 図 16 の解析解での $a = 20$ と $a = 5$ のときほぼ同じである. この図から多少の変動はあるものの PC 台数を増やすことにより, 処理時間を減少させることができ, 分割数 n を増やすより効果の大きいことが分かる. また, シーンチェンジ間隔の小さい Variety で DL 方式の方が FL 方式よりも小さい処理時間となっている. このことから DL 方式の方が有利であることが分かる.

1 台の PC でトランスコードするときの処理時間は表 1 の w_i と T_{cp} の積であり, このシミュレーションでは 7740 秒となる. これに対して, 10 台の PC で分散符号変換したときでおよそ 1/5 に, 20 台の PC のときで 1/10 程度にできることが分かる. シーン適応

に行わない場合には 1/6 の時間短縮がはかれる⁴⁾ のに対して, 処理時間短縮の割合は下がるが, その分, 分割点での画品質劣化が少なく, より高品質な符号変換を実現できる可能性を有している. 今後の課題として, 提案方式での画品質評価を実施し, この画品質劣化低減の効果を明らかにする必要がある.

5. おわりに

本稿では, シーンチェンジを境界とするようにトランスコードを行うことによるシーン適応ビデオトランスコーディング方式について述べた. この方式は分割点での画質不連続性が知覚されにくいところで分割することに特徴があり, これにより, 詳細な符号化のレート制御が行えないような様々な符号方式への変換も, 分散処理によって高速化することが可能である. 負荷分散の方式としては, 固定長割当てと Load Divisible Theory に基づく可変長割当ての 2 つの割当て方法について比較検討した. 検討した結果をシミュレーション実験によって検証し, Load Divisible Theory に基づくセグメント割当てが有利であり, 分割数は 2 程度で最適となることを明らかにした. また, シーンチェンジ間隔が長い映像ほど処理時間が長くなってしまふものの, シーンチェンジ間隔の比較的長いドキュメンタリ映像からのデータを用いた実験でも, 10 台の PC へ分散化しておよそ 1/5 程度の処理時間にすることができることを示した.

今後の課題としては本方式を実装した実験システムで映像種類を増やして画品質評価および性能検証を行うことや, 異なる性能の PC を組み合わせた場合の最適な割当て法, および, さらなる性能向上のため, ソース PC 側で簡易な処理によってある程度の精度でシーンチェンジ検出を行うことで再符号変換量を抑え, 組合せ最適化手法を用いる割当てを行うなどの検討があげられる.

参考文献

- 1) 笠井裕之, 富永英義, 花村 剛, 亀山 渉: 低遅延 MPEG-2 ビデオトランスコーダ符号量制御方式, 信学論 (B), Vol.J83-B, No.2, pp. 151-164 (2000).
- 2) 阿久津隆史, 木全英明, 清水 淳, 八島由幸, 小林直樹: 低ビットレート用 MPEG-2 リアルタイムソフトウェアトランスコーダ「Trampeg」の開発, 信学論 (D-II), Vol.J84-DII, No.6, pp.1084-1093 (2001).
- 3) Shanableh, T. and Ghanbari, M.: Heterogeneous Video Transcoding to Lower Spatio-

Temporal Resolutions and Different Encoding Formats, *IEEE Trans. on Multimedia*, Vol.2, No.2, pp.101–110 (2000).

- 4) 三部靖夫, 渡辺真太郎, 于 冬, 中村太一, 若宮直紀: 分散処理による高速ビデオトランスコーディングシステム, マルチメディア, 分散, 協調とモバイル (DICOMO2003) シンポジウム, pp.409–412 (2003).
- 5) Sambe, Y., Watanabe, S., Yu, D., Nakamura, T. and Wakamiya, N.: A High Speed Distributed Video Transcoder for Multiple Rates and Formats, *2003 International Technical Conference on Circuits/Systems Computers and Communications (ITC-CSCC2003)*, pp.921–924 (2003).
- 6) Ohtsuka, S. and Kosugi, M.: Quality evaluation of locally impaired pictures, *Proc. SID, SID*, pp.19–24 (1991).
- 7) Farin, D., Mache, N. and With, P.H.N.: A software-based high-quality MPEG-2 encoder employing scene change detection and adaptive quantization, *IEEE Trans. Consumer Electronics*, Vol.48, No.4, pp.887–897 (2002).
- 8) Pyun, J.Y., Kim, Y., Ko, S.J. and Song, H.: Scene-adaptive frame-layer rate control for low bit rate video, *IEICE Trans. Fundamentals*, Vol.E86-A, No.10, pp.2618–2622 (2003).
- 9) 長坂晃朗, 田中 謙: カラービデオ映像における自動索引付け法と物体探索法, 情報処理学会論文誌, Vol.33, No.4, pp.543–550 (1992).
- 10) Gargi, U., Kasturi, R. and Strayer, S.H.: Performance characterization of video shot change detection methods, *IEEE Trans. Circuits Syst. Video Technology*, Vol.10, No.1, pp.1–13 (2000).
- 11) 鈴木賢一郎, 中嶋正臣, 坂野 鋭, 三部靖夫, 大塚作一: 動き方向ヒストグラム特徴を用いた映像データからのカット点検出, 電子情報通信学会論文誌, Vol.J86-D-II, No.4, pp.468–478 (2003).
- 12) Bharadwaj, V., Ghose, V.M.D. and Robertazzi, T.G.: *Scheduling divisible loads in parallel and distributed systems*, IEEE Computer Society Press (1996).
- 13) Robertazzi, T.G.: Ten reasons to use divisible load theory, *IEEE Computer*, Vol.36, No.5, pp.63–68 (2003).
- 14) Jain, R.: *The art of computer systems performance analysis*, John Wiley & Sons. Inc. (1991).
- 15) 塩見 弘: 信頼性工学入門, 丸善 (1982).

(平成 16 年 1 月 8 日受付)

(平成 16 年 10 月 4 日採録)



三部 靖夫 (正会員)

1986 年大阪大学基礎工学部卒業 . 1988 年同大学院基礎工学研究科博士前期課程修了 . 同年日本電信電話 (株) 入社 . 1988 年より NTT データ通信 (株) . マルチメディア通信システム, 動画像トランスコーディングの研究に従事 . 工学博士 . ACM, IEEE, 電子情報通信学会, 画像電子学会各会員 .



大上 貴充

1998 年富山大学工学部卒業 . 2000 年同大学院理工学研究科博士前期課程修了 . 同年 (株) NTT データ入社 . 電子透かし, 動画像トランスコーディングの研究に従事 .



渡辺真太郎

1994 年東京工業大学理学部情報科学科卒業 . 1996 年奈良先端科学技術大学院大学博士前期課程修了 . 同年 NTT データ通信 (株) 入社 . 現在 (株) NTT データ技術開発本部にて画像処理, 映像処理の研究に従事 . 電子情報通信学会会員 .



中村 太一 (正会員)

1972 年千葉大学工学部電気卒業 . 1974 年同大学院修士課程修了 . 同年日本電信電話公社電気通信研究所入所 . 以来, 画像通信システムの研究に従事 . 1988 年より NTT データ通信 (株) にて映像通信システム, Web システムの研究に従事 . 2003 年より東京工科大学コンピュータサイエンス学部教授, 現在に至る . Web マイニング, 映像分散トランスコーディングおよびプロジェクトマネジメントの研究に従事 . 工学博士 . 電子情報通信学会, IEEE, プロジェクトマネジメント学会各会員 .



若宮 直紀（正会員）

1992年大阪大学基礎工学部中退．

1994年同大学院基礎工学研究科博士前期課程修了．

1996年同後期課程修了．

同年より同大学院基礎工学研究科助手．

1997年同情報処理教育センター助手．

1999年同大学院基礎工学研究科講師．

2002年より同大学院情報科学研究科助教授．

マルチメディア情報システム，センサネットワーク等に関する研究に従事．

工学博士．ACM，IEEE，電子情報通信学会各会員．
