インデックス付きライブ映像ストリームによる動的番組生成と配信

角谷和俊[†1] 川口知昭[†2] 土居明弘[†3]
赤迫貴行[†4] 田中克己[†5]

本論文では，インターネットによる放送を用いてビデオ映像の編集，配信，およびフィルタリングをおこなう方式の提案を行う．まず，ビデオ映像に対してリアルタイムにインデックスを付与することで内容記述を行なう方式について述べる．また，ネットワーク環境において配信されたビデオ映像をフィルタリングするための番組スケルトンを提案し，複数のライブ映像が同時に配信された場合の優先順位の決定と制御方法について検討する．さらに，実験環境としてマルチキャスト・ネットワークを用いて設計したプロトタイプシステム ScoopCast について述べる．

Dynamic Video Production and Delivery from Indexed Live Video Stream

KAZUTOSHI SUMIYA ,[†1] TOMOAKI KAWAGUCHI ,[†2] AKIHIRO DOI ,[†2]
TAKAYUKI AKASAKO [†4] and KATSUMI TANAKA [†5]

In this paper, we propose a new method for indexing, delivering, and filtering live video streams in the Internet broadcasting environment for grass root users. We discuss the scheduled program skeleton to describe fixed scenario of broadcast programs. In addition, we propose the dynamic program producing mechanism based on a degree of broadcast priority to avoid conflicts from unspecific number of providers. We implemented a prototype called ScoopCast which can deliver video stream using multicast networks.

1. はじめに

現在，インターネットやデジタル衛星放送を用いたデータ配信方式[1]により，ネットワークを介してビデオ映像やテキストデータをリアルタイムに配信する技術が注目を集めている．また，デジタルビデオの普及により，ビデオ映像の編集は非常に容易なものになってきている．ネットワーク環境においては，10MbpsのCATVのサービスが開始され，ADSLやVDSLといったサービスの準備も進められている．一方，モバイル環境においても同様に，携帯電話などを用いた1Mbpsを超える高速ネットワークの実験が始まっている．

ネットワーク環境の整備に伴い，ユーザはデータを単に受信するだけでなく，自らビデオ映像やテキストなどの様々なデータを配信することが可能となってきている．もし，各ユーザが各々のビデオ映像を配信することが可能になれば，非常に大量の新鮮な映像がネットワーク上に存在することになる．さらに，もしリアルタイムに映像を送信する仕組みがあれば，ユーザ個人の映像をライブでネットワークに配信することが可能となる．従って，たくさんの個人のライブ映像を動的に合成してリアルタイムに新しいビデオ映像を作成することが可能であれば，ビデオ映像放送の新しいパラダイムとなりうる可能性がある．

我々の研究で扱う映像は，テレビ局で作成されたテレビ番組での映像と限定はしていない．一般のテレビ番組はあらかじめ決められたシナリオに基づいて専門のディレクターにより製作される．一方，我々が提案するビデオ映像 (以下，**番組**と呼ぶ) は，アマチュア

†1 神戸大学都市安全研究センター 都市情報システム研究分野
Research Center for Urban Safety and Security, Kobe University
†2 NTT 西日本 法人営業本部
Business Communications Headquarters, NTT-West
†3 NTT 西日本 神戸支店法人営業部
KOBE Branch Corporation Sales Dept., NTT-West
†4 NTT コミュニケーションウェア株式会社 技術開発部
NTT Communicationware corporation
†5 神戸大学大学院 自然科学研究科 情報メディア科学専攻
Graduate School of Science and Technology, Kobe University

が撮影した生のビデオ映像から動的に作成される映像ストリームである．一般に，ユーザのビデオ映像を集めて編集する行為は番組の作成・演出であると考えられる．例えば，マラソン中継において，沿道の観戦者がビデオカメラを持っていれば，ある種のスクープ映像を撮影することができるかもしれない．スクープ映像には予期せぬ出来事が含まれており，ユーザはこの出来事をリアルタイムに伝えたいという意志を持っていると考えられる．すなわち，一般ユーザは，作成・編集の作業さえ簡単であれば，ユーザ独自の番組を作成する動機を持っていると考えられる．

本論文では，インターネットやデジタル衛星放送などのネットワーク環境におけるライブ映像ストリームのインデックス付け，配信，およびフィルタリングに関するフレームワークを提案する．すなわち，専門家でない一般ユーザがビデオ映像をリアルタイムに処理するための方式およびシステムを提供することが本研究の目的である．従って，本研究の要件は以下の通りである：

- オリジナルのビデオ映像に対してリアルタイムにインデックスを付与する機能および配信機能
- オリジナルのビデオ映像と付与されたインデックスに基づく動的番組作成機能
- 配信されてくるマルチ映像ストリームの自動的フィルタリング機能

基本的な手順は以下の通りである．

(1) オリジナルのビデオ映像は，内容を分析しインデックスが付与される．

(2) インデックスが付与されたビデオ映像は，インデックス付きビデオストリームとしてネットワークに配信される．

(3) 配信された複数のインデックス付きビデオストリームからビデオ映像の一部分を選択することにより動的に番組が生成され，リアルタイムに配信される．

(4) 番組視聴者は配信された番組を視聴する．

現在，我々は提案する手法に基づき，ScoopCast と呼ばれるプロトタイプシステムを作成している[2)~4)]．

以下，本論文の構成を示す．2 章では本研究の動機とアプローチについて述べる．3 章ではライブビデオ映像に対するインデックス付与について議論し，4 章ではリアルタイムビデオ記述モデルについて述べる．5 章では優先度に基づく動的番組生成方式について述べる．6 章ではプロトタイプシステム ScoopCast の実装について述べ，7 章では結論と今後の課題について述べる．

2. 動機とアプローチ

高速ネットワークやデジタルビデオ機器の発達により，一般ユーザによるビデオ映像は広範囲にわたって流通している．ビデオ映像は他のメディアに比べて情報量が多く，重要なメディアである．情報の価値という意味では，ライブ映像は特に利用価値が高いと考えられる．その中で特に価値のあるビデオ映像は**スクープ映像**となる可能性がある．一方，通常のテレビのニュース番組では，ビデオ映像素材をあらかじめ編集してから放送するために，たとえリアルタイムに映像が提供されても放送することは一般的には不可能である．我々のアプローチの基本的コンセプトは以下の通りである．

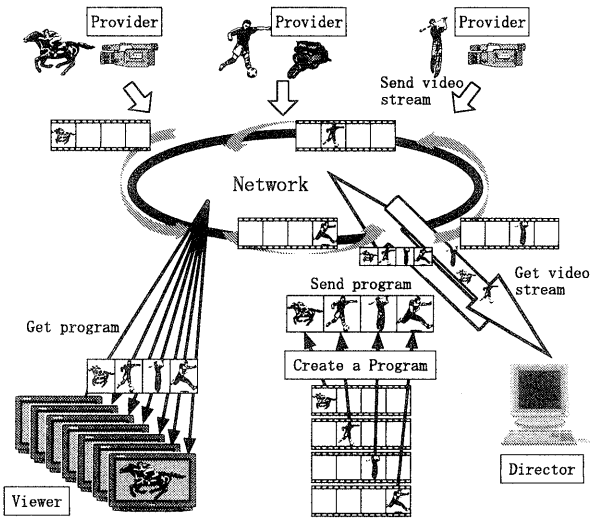

図 2 ScoopCast の概要

(1) 映像提供者は，スクープ映像を撮影した場合，そのビデオ映像にキーワード付与による内容記述を行う．同時に，インデックス付けされた映像ストリームをネットワークに配信する．

(2) 番組編集者は，どのような条件の映像ストリームを番組映像として採用するかを，キーワードを用いてあらかじめ設定しておく．キーワードの照合により条件を満たす映像ストリームは，ライブ映像番組として合成されリアルタイムに送信される．

(3) 視聴者は番組を受信する．

我々のアプローチでは，上記の (1) ～ (3) の 3 種類

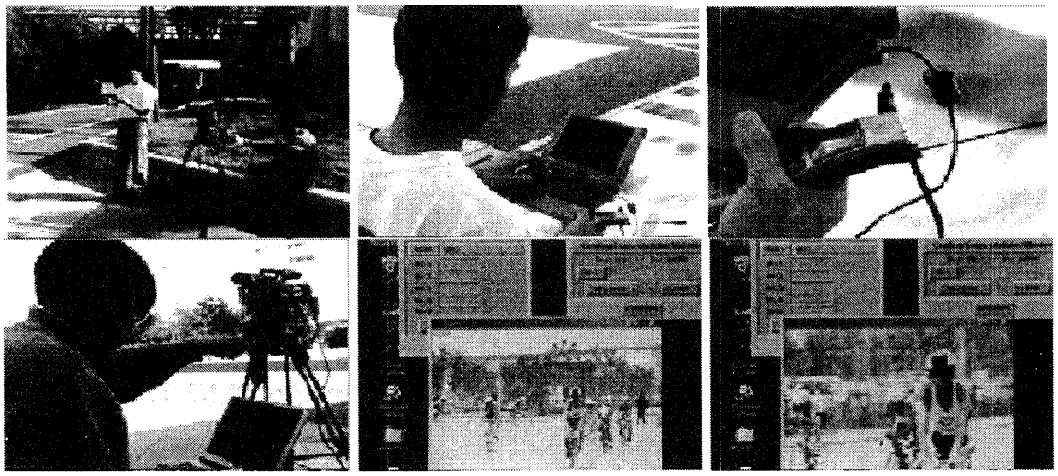

図 1 ScoopCast の実験環境

の立場のユーザが存在しており☆，それぞれ**プロバイダ** (provider)，**ディレクタ** (director)，**視聴者** (viewer) と呼ばれる．もちろん，映像提供者および番組編集者は，専門家という意味ではなく，アマチュアのユーザを含めて想定している．また，本研究で扱う映像ストリームは 2 種類に分類される:

通常番組映像 (scheduled program video) 番組構成上，基本となる映像ストリームのこと．

スクープ映像 (scoop video) 常時存在するのではなく，突発的に発生するスクープ性の高い映像ストリームのこと．緊急性が要求されるため，通常番組映像の途中に挿入される．

図 2 は ScoopCast の概要を示している．プロバイダ，ディレクタ，および視聴者はそれぞれネットワークに接続している．プロバイダはインデックス付きビデオ映像をネットワークに送信する．ディレクタは目的に応じたビデオ映像のみを受信し，編集を行った上でネットワークに配信する．図 1 は ScoopCast の実験環境を示している．

ScoopCast は様々な種類のアプリケーションに適用可能である．例えば，スポーツ中継，地域の祭りなどのイベントのリアルタイムな映像配信である．このようなアプリケーションにおいては，映像対象の状況は時間とともに変化する．マラソン中継においては，リタイアや故障など予期せぬ出来事 (event) が起こる可能性がある．しかし，従来の番組放送では，広範囲にわたるすべてのレースの模様を把握することは不可能である．もし，沿道の観戦者がビデオカメラを用いて出来事を撮影し，その映像を簡単にネットワークに送信することが出来れば，そのビデオ映像は価値あるスクープとして配信することが可能である．

プロバイダがライブ映像を送信する時に必要な機能は以下の 2 つである：1) リアルタイム映像送信機能，2) リアルタイム内容記述機能．一方，ディレクタが映像を編集する場合には，想定する番組を記述するための機能が必要である．次章では，これらの機能について詳しく述べる．

3. ライブ映像ストリームのためのインデックス生成

3.1 ビデオ・インデックス

複数のビデオ映像の内容を同時に分析し，映像ストリームとして編集することは一般的には困難である．特に，ライブ映像の場合は，いつ何処からビデオ映像が配信されてくるかが分からないため，瞬時に配信されてきたビデオ映像の価値を判断することが難しい．そこで，配信されてくるビデオ映像の価値や必要性を判断するために，ビデオ映像に含まれる内容を記述する方法が必要になる．自然言語を用いてビデオ映像の内容をシーン毎に記述することは可能であるが，現実的なアプローチであるとは言えない．

ビデオ映像に対するインデックス付けについては，様々な方法が提案されている．例えば，スクリプト言語を用いる方法 [5),6)]，シナリオ言語を用いる方法 [7)~9)] である．これらの方法は蓄積されたビデオ映像に対しては非常に効果的であるが，リアルタイムにビデオ映像を処理する目的には適用できない．また，PRAJA プレゼンス環境において開発された ActionSnaps システム [10)] は，半自動でライブイベントを要約しビデオ映像に内容記述を行うことが可能である．しかし，

☆ 同一ユーザが複数の立場になることも可能である．

現段階では複数のビデオ映像を扱うことが出来ない．

従来のビデオデータベースの研究では，ビデオ映像の内容記述について，いくつかの方法が提案されている．Shahraray らは，映像に付与されているクローズドキャプションをそのシーンの内容記述として用いる方法を提案している [11]．また，Ariki らは，ニュース映像に含まれるテロップなどの文字情報を画像認識により抽出する方法を提案している [12]．しかしながら，これらの方法は既に編集済みの映像とその付加情報に基づくために，ライブ映像の内容記述には適用できない．

我々は 2 種類のライブ映像のためのリアルタイム内容記述方式を提案する．一つは，音声認識技術を用いてビデオの撮影者が映像に注釈をつける方式である．もう一つは，イベント発生に伴い，そのイベントを GUI を用いて瞬時に入力することにより映像にイベントの記述を付与する方式である．これらの 2 つの方式は，撮影の対象に応じて，より適切な方式を選択することが可能である．

3.1.1 音声認識による内容記述

音声認識による内容記述は，リアルタイムにキーワードを入力することが可能である点で，テキスト入力やオフライン編集に比べて効率が良い．我々は音声認識技術を用いてライブ映像の内容記述を行うツールを開発している [13]．

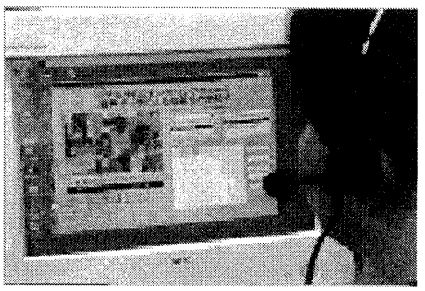

図 3 音声認識による内容記述

ユーザは，現在の状況をマイクを通して音声により説明する．これは，従来の放送番組では実況中継のアナウンサーの役割に相当する．音声認識ソフトウェアにより，ユーザが発話した内容からキーワードを抽出する．図 3 はヘッドホンマイクを用いて内容を記述する実験の様子を示している．本ツールは音声認識ソフトウェアとして IBM の ViaVoice98 を用いており，ツール本体は Java により実装されている．

筆者らは，ビデオ映像撮影中にある事象が発生した場合，その事象をキーワードとして記述するためのリアルタイム内容記述ツールを開発している [14]．図 4 は野球の試合のための内容記述ツールの画面例を示している．試合中の事象 (例えば，ヒット，三振，ホームランなど) が進行状況に応じて入力されていく．このツールでは GUI(キーワードボタンをマウスでクリックする等) を用いて各々の事象の入力を行う．また，試合に関連する情報 (選手の情報，過去の成績など) をネットワーク経由で取得しブラウザで表示することも可能である．

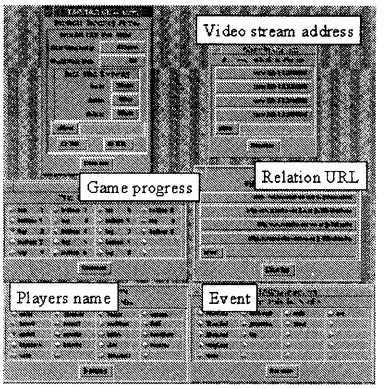

図 4 内容記述ツール

3.2 時制インデックス

3.1 節では，キーワードによる内容記述について述べたが，ライブ映像ストリームでは時間に関するインデックスが重要である．例えば，キーワードがビデオ映像のどの時間に付与されたかなどである．この時間のことをトランザクション時間 (transaction time) と呼ぶ．トランザクション時間は，ある事象をシステムに登録した時間である．

一方，付与されたキーワードが有効である時間 (区間) のことを有効時間 (valid time) と呼ぶ．言い換えれば，ビデオ映像と有効時間の組により，映像対象の状況を表現していることになる．

4 章では，ビデオ映像のための 3 種類のインデックスについて述べる．4.1 節では，インデックスの構成要素，4.2 節では，3 種類のインデックス型，4.3 節では，各インデックス型の有効時間の推定方法について述べる．なお，キーワードはユーザがイベントツールや内容記述ツールを用いて入力を行なうものであり，時制インデックスに関してはシステムにより自動的に計算される．

4. リアルタイム・ビデオ映像内容記述モデル

4.1 ビデオ映像のためのインデックス

ライブ映像ストリームのための記述モデルを提案する．インデックスはビデオ映像のある区間の内容記述に相当する．ビデオ映像 V における，インデックス I_V は以下の式で表される．

$$I_V = \{r | r = (d, dt, tt, v), v \subseteq V\} \quad (1)$$

ここで，r はインデックスを構成する要素，d は内容 (キーワード)，dt は記述型，tt はトランザクション時間，v は有効時間である．

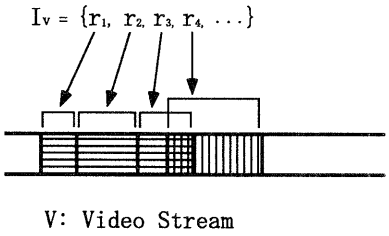

図 5 ビデオ映像 V におけるインデックス I_V

記述の型には**イベント型**と**状況型**の 2 つの種類がある:

イベント型: ビデオ映像の中の事象の記述のために用いられる．例えば，野球の試合であれば，ストライク，ヒット，ホームラン，ファインプレーなどである．

状況型: 状態の記述のために用いられる．例えば，ストライク・カウント，ボール・カウント，得点などである．

イベント型は，更に 2 つの型 (**原子イベント型**，**複合イベント型**) に分けられる．原子イベント型は単一の事象を表すのに対し，複合イベント型は複数の事象から構成される事象を表す．例えば，「三振」はある打者の打席の事象 (ストライク，ファイル，空振りなど) から構成される事象であり，複合イベントである．

トランザクション時間は，ある事象がシステムに登録された時間である．有効時間は，インデックスが有効である区間である．有効時間 v は，以下のように表される．

$$v = [t_s, t_e] \quad (2)$$

t_s は有効時間の開始時刻，t_e は終了時刻である．

4.2 記 述 型

イベント型記述 d_e は以下のように表される．

$$d_e = (object, role, event) \quad (3)$$

ここで，$ojbect$ は実体，$role$ は役割，$event$ は事象である．例えば，野球の例では，("$McGwire$", "$batter$","$home\ run$") はイベント型の記述である．

一方，状況型記述 d_s は以下のように表される．

$$d_s = (situation, value) \quad (4)$$

ここで，$situation$ は記述対象の状態，$value$ は数値型データである．例えば，野球の例では，("$strike\ count$", "2") などのように表される．

4.3 有効時間の区間推定

インデックスの有効時間 (区間) はシステムによって自動的に計算される．区間推定の方法は，記述型により異なる．

4.3.1 原子イベント型

原子イベント型のインデックスレコード r_{e_i} の有効時間 $r_{e_i}.v$ は以下のように計算される．

$$r_{e_i}.v = [r_{e_{i-k}}.tt + l, r_{e_i}.tt + l] \quad (5)$$

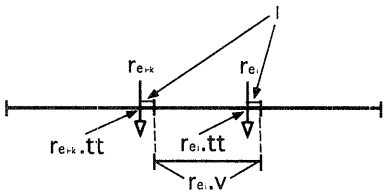

図 6 原子イベント型インデックスの有効時間

ここで，$r_{e_i}.tt$ は，このインデックスレコードのトランザクション時間，$r_{e_{i-k}}.tt$ は一つ前のインデックスレコードのトランザクション時間である．l はタイムラグである☆ (図 6).

4.3.2 複合イベント型

複合イベント型のインデックスレコードは，いくつかの原子イベントレコードの集合として表される．記述対象の性質を表現するために，有限オートマトンを用いて，複合イベント型のインデックスレコードの有効時間を計算する方法を提案する (図 7).

複合イベント型の記述における事象 ($event$) の種類ごとに，決定性有限オートマトンを用意する．複合イベント型のインデックスレコードが生成されると，その記述における $event$ に対応する決定性有限オートマトンが起動し，過去のインデックスレコード系列における記述系列を入力記号列として遡りながら走査する (図 7)．その際，状態遷移と同時に，入力記号を表す記述に対応する有効区間を出力する．この時，記述系列が受理されたとき得られる出力系列は，各記述に対

☆ タイムラグはシステムの処理時間などにより発生する．基本的に一定である．

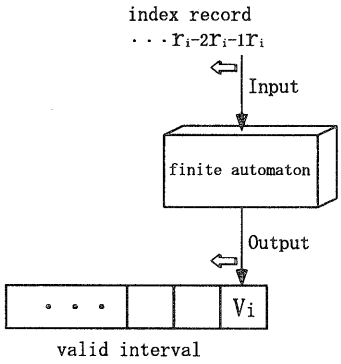

図 7 複合イベント型の有効時間を区間推定のための有限オートマトン

する有効区間の集合となる．

有効区間導出オートマトン M を以下のように定義する．M は，出力として「受理する」,「受理しない」を表す"1","0"以外に，有効区間を考える必要があるため，ミーリー型順序機械における出力記号の概念を導入する．ここでは，入力記号をインデックスレコードにおける記述とし，出力記号を入力記号としての記述を含むインデックスレコードの有効区間としている：

$$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0, F) \quad (6)$$

Q ：状態の有限集合

Σ ：入力記号の有限集合

入力記号は記述 a で表現する．

$\Sigma = \{a_1, a_2, \cdots, a_n\}$

Δ ：出力記号の有限集合

出力記号は，入力記号である記述を持つインデックスレコードの有効区間 v で表現する．v は映像データ全体を表す V の部分集合である．

$\Delta = V\ (v \in V)$

δ ：状態推移関数

M の現在の状態 $p(\in Q)$ とそれへの入力 $d(\in \Sigma)$ のすべての組み合わせに対して，次の時点の状態 $q(\in Q)$ を $\delta(p, a) = q$ により一意的に定める．

λ ：出力関数

M の現在の状態 $p(\in Q)$ とそれへの入力 $a(\in \Sigma)$ のすべての組み合わせに対して，次の状態へ推移する間に出力する出力記号 $v(\in \Delta)$ を $\lambda(p, a) = v$ により一意的に定める)．この場合，入力 a に一致する記述を持つインデックスレコード r の有効区間 $r.v$ が出力となる．

$q_0(\in Q)$ ：初期状態

$F(\subseteq Q)$ ：最終状態の集合

ここで，野球中継映像を例として，複合イベント「三振」を含むインデックスレコードの有効区間を決定するための，時区間群を求める過程について述べる．

図 8 は，複合イベント「三振」を含む複合イベント型のインデックスレコードが生成されると起動する有効区間導出オートマトンである．このオートマトンにおいて定義されている入力記号 $a_i(i = 0, 1, \cdots, 15)$ は以下の通りである．

(1) $a_0 = (*, バッター, 打席に入る)$
(2) $a_1 = (*, バッター, 空振り)$
(3) $a_2 = (*, バッター, 打った)$
(4) $a_3 = (*, バッター, ファウル)$
(5) $a_4 = (*, ピッチャー, 投げた)$
(6) $a_5 = (ボール, 投球, ストライク)$
(7) $a_6 = (ボール, 投球, ボール)$
(8) $a_7 = (ストライクカウント, 0)$
(9) $a_8 = (ストライクカウント, 1)$
(10) $a_9 = (ストライクカウント, 2)$
(11) $a_{10} = (ボールカウント, 0)$
(12) $a_{11} = (ボールカウント, 1)$
(13) $a_{12} = (ボールカウント, 2)$
(14) $a_{13} = (ボールカウント, 3)$
(15) $a_{14} = (ボールカウント, *)$
(16) $a_{15} = (バッター, *)$

ここでは記述において，項の値として記号 $*$ が定められているものがあるが，これはその項においてはどんな値をも取り得ることを意味する．すなわち，a_0 において，*object* は「松井」であっても「イチロー」であってもかまわない．

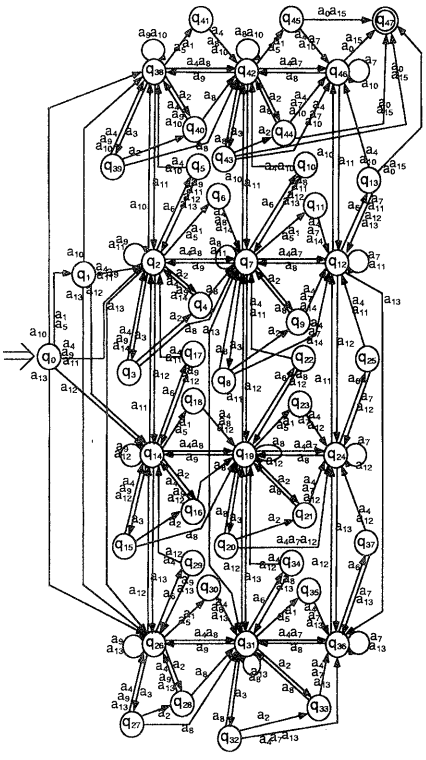

図 8 複合イベント「三振」の有限オートマトン

複合イベント「三振」を含むインデックスレコード r_k が生成された時点での，インデックスレコード系列 $r_0 \cdots r_{k-1} r_k$ において，図 8 のオートマトンに走査され得るインデックスレコードの記述系列を以下のものとする．

(1) $r_{k-30}.d$ = (バッター, 八木)
(2) $r_{k-29}.d$ = (野村, *, 投げた)
(3) $r_{k-28}.d$ = (*, *, ストライク)
(4) $r_{k-27}.d$ = (ストライクカウント, 1)
(5) $r_{k-26}.d$ = (ボールカウント, 0)
(6) $r_{k-25}.d$ = (ストライクカウント, 1)
(7) $r_{k-24}.d$ = (ボールカウント, 0)
(8) $r_{k-23}.d$ = (*, *, 投げた)
(9) $r_{k-22}.d$ = (*, *, ストライク)
(10) $r_{k-21}.d$ = (ストライクカウント, 2)
(11) $r_{k-20}.d$ = (ボールカウント, 0)
(12) $r_{k-19}.d$ = (ストライクカウント, 2)
(13) $r_{k-18}.d$ = (ボールカウント, 0)
(14) $r_{k-17}.d$ = (ストライクカウント, 2)
(15) $r_{k-16}.d$ = (ボールカウント, 0)
(16) $r_{k-15}.d$ = (野村, *, 投げた)
(17) $r_{k-14}.d$ = (*, *, ボール)
(18) $r_{k-13}.d$ = (ストライクカウント, 2)
(19) $r_{k-12}.d$ = (ボールカウント, 1)
(20) $r_{k-11}.d$ = (*, *, 打った)
(21) $r_{k-10}.d$ = (ストライクカウント, 2)
(22) $r_{k-9}.d$ = (ボールカウント, 1)
(23) $r_{k-8}.d$ = (*, *, 投げた)
(24) $r_{k-7}.d$ = (*, *, ボール)
(25) $r_{k-6}.d$ = (ストライクカウント, 2)
(26) $r_{k-5}.d$ = (ボールカウント, 2)
(27) $r_{k-4}.d$ = (ストライクカウント, 2)
(28) $r_{k-3}.d$ = (ボールカウント, 1)
(29) $r_{k-2}.d$ = (野村, *, 投げた)
(30) $r_{k-1}.d$ = (*, *, 空振り)
(31) $r_k.d$ = (*, *, 三振)

これは，1998 年 5 月 22 日のプロ野球阪神対横浜 (甲子園) の毎日放送ラジオの実況内容を基にしている．ここでも，記述の項において記号 * が用いられているが，この場合は情報の欠如を意味する．つまり，記述が生成される段階で情報の欠落の問題が発生している．また，情報の重複がいたるところで起こっている ($r_{k-27}.d$ に対する $r_{k-25}.d$ など)．また，$r_{k-3}.d$ は明らかに正しくない記述である．

インデックスレコード r_k が生成されると，前述のオートマトンが起動する．そして，以下のような状態推移を行なう．

(1) $q_0 \stackrel{a_1}{\Longrightarrow} q_1$ $(equals(r_{k-1}.d, a_1) = true)$
(2) $q_1 \stackrel{a_4}{\Longrightarrow} q_2$ $(equals(r_{k-2}.d, a_4) = true)$
(3) $q_2 \stackrel{a_{11}}{\Longrightarrow} q_2$ $(equals(r_{k-3}.d, a_{11}) = true)$
(4) $q_2 \stackrel{a_9}{\Longrightarrow} q_2$ $(equals(r_{k-4}.d, a_9) = true)$
(5) $q_2 \stackrel{a_{12}}{\Longrightarrow} q_{14}$ $(equals(r_{k-5}.d, a_{12}) = true)$
(6) $q_{14} \stackrel{a_9}{\Longrightarrow} q_{14}$ $(equals(r_{k-6}.d, a_9) = true)$
(7) $q_{14} \stackrel{a_6}{\Longrightarrow} q_{17}$ $(equals(r_{k-7}.d, a_6) = true)$
(8) $q_{17} \stackrel{a_4}{\Longrightarrow} q_2$ $(equals(r_{k-8}.d, a_4) = true)$
(9) $q_2 \stackrel{a_{11}}{\Longrightarrow} q_2$ $(equals(r_{k-9}.d, a_{11}) = true)$
(10) $q_2 \stackrel{a_9}{\Longrightarrow} q_2$ $(equals(r_{k-10}.d, a_9) = true)$
(11) $q_2 \stackrel{a_2}{\Longrightarrow} q_4$ $(equals(r_{k-11}.d, a_2) = true)$
(12) $q_4 \stackrel{a_{14}}{\Longrightarrow} q_2$ $(equals(r_{k-12}.d, a_{14}) = true)$
(13) $q_2 \stackrel{a_9}{\Longrightarrow} q_2$ $(equals(r_{k-13}.d, a_9) = true)$
(14) $q_2 \stackrel{a_6}{\Longrightarrow} q_5$ $(equals(r_{k-14}.d, a_6) = true)$
(15) $q_5 \stackrel{a_4}{\Longrightarrow} q_{38}$ $(equals(r_{k-15}.d, a_4) = true)$
(16) $q_{38} \stackrel{a_{10}}{\Longrightarrow} q_{38}$ $(equals(r_{k-16}.d, a_{10}) = true)$
(17) $q_{38} \stackrel{a_9}{\Longrightarrow} q_{38}$ $(equals(r_{k-17}.d, a_9) = true)$
(18) $q_{38} \stackrel{a_{10}}{\Longrightarrow} q_{38}$ $(equals(r_{k-18}.d, a_{10}) = true)$
(19) $q_{38} \stackrel{a_9}{\Longrightarrow} q_{38}$ $(equals(r_{k-19}.d, a_9) = true)$
(20) $q_{38} \stackrel{a_{10}}{\Longrightarrow} q_{38}$ $(equals(r_{k-20}.d, a_{10}) = true)$
(21) $q_{38} \stackrel{a_9}{\Longrightarrow} q_{38}$ $(equals(r_{k-21}.d, a_9) = true)$
(22) $q_{38} \stackrel{a_5}{\Longrightarrow} q_{41}$ $(equals(r_{k-22}.d, a_5) = true)$
(23) $q_{41} \stackrel{a_4}{\Longrightarrow} q_{42}$ $(equals(r_{k-23}.d, a_4) = true)$
(24) $q_{42} \stackrel{a_{10}}{\Longrightarrow} q_{42}$ $(equals(r_{k-24}.d, a_{10}) = true)$
(25) $q_{42} \stackrel{a_8}{\Longrightarrow} q_{42}$ $(equals(r_{k-25}.d, a_8) = true)$
(26) $q_{42} \stackrel{a_{10}}{\Longrightarrow} q_{42}$ $(equals(r_{k-26}.d, a_{10}) = true)$
(27) $q_{42} \stackrel{a_8}{\Longrightarrow} q_{42}$ $(equals(r_{k-27}.d, a_8) = true)$
(28) $q_{42} \stackrel{a_5}{\Longrightarrow} q_{45}$ $(equals(r_{k-28}.d, a_5) = true)$
(29) $q_{45} \stackrel{a_4}{\Longrightarrow} q_{46}$ $(equals(r_{k-29}.d, a_4) = true)$
(30) $q_{46} \stackrel{a_{15}}{\Longrightarrow} \underline{q_{47}}$ $(equals(r_{k-30}.d, a_{15}) = true)$ ← 受理

一連の状態推移において，記述の重複は，(19) などのように同じ状態へ推移させることで解決している．また，記述の誤りについては，重複して生成される状況型の記述を利用して，然るべき状態へ推移するようにしている．ここでは，(3) の時点では状態 q_2 であるが，(5) において状態 q_{14} に推移させることで誤りを補正している．

4.3.3 状況型

状況型のインデックスレコード r_{e_i} の有効時間 $r_{s_i}.v$ は以下のように計算される．

$$r_{s_i}.v = [r_{s_i}.tt + l, r_{s_{i+k}}.tt + l] \quad (7)$$

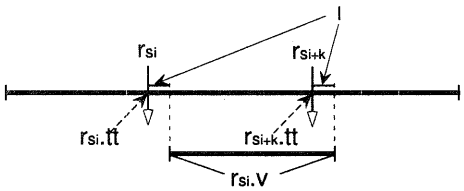

図 9 状況型インデックスの有効時間

ここで，$r_{s_{i+k}}.tt$ は，このインデックスレコードのトランザクション時間，$r_{s_i}.tt$ は *situation* が同じで，かつ *value* の値が違うインデックスレコードの中で，時間的に一番近いインデックスレコードのトランザクション時間である (図 9)．なお，l はタイムラグである．例えば，あるインデックスレコードが，(”*strike count*”,”2”) であるとすると，*situation* が同じで，

かつ *value* の値が違うインデックスレコードは，(”*strike count*”,”1”) である．

5. 動的番組生成

5.1 配信優先度

ディレクタは複数のライブ映像から意図に適した映像を選択し，編集して番組を作成する．本研究で扱う映像ストリームが 2 種類あることは前述した．すなわち，通常番組映像とスクープ映像である．ディレクタはどのタイミングでどのような映像を放送するかをあらかじめ決めておく．言い換えると，通常番組映像により番組を構成し，予期せぬスクープ映像が配信されてきた場合は，通常番組映像をスクープ映像に自動的に差し替えて放映する．

一般的にはスクープ映像の方が，通常番組映像よりも優先度が高いと考えられる．もちろん，スクープ映像は時間が経つにつれて，その価値は変化する (基本的には減少する)．我々は，スクープ映像の配信優先度を計算する方式を提案する．

本論文で提案する方式は，映像の付与されたインデックス中のキーワードの出現時間および出現回数に基づいている．言い換えると，同じキーワードを持つインデックス同士であっても，そのインデックスが付与された時刻によって，そのインデックスが示す映像の優先度が違うことがある☆．また，同じキーワードを持つインデックス同士であっても，そのキーワードがビデオ映像の最初から何回使用されたかによって，優先度が違うことになる．以下に重要度と新鮮度の例を示す：

(重要度の例) マラソンレースの場合,「先頭集団」の映像はスタート直後やゴール直前においては，非常に高い重要度を持つと考えられる．しかし，レースの中盤においては，スタート直後やゴール直前ほどは高くないと考えられる．一方,「棄権」や「アクシデント」などは，もしこの事象が起こった場合は，どの時点であろうと常に高い重要度を持つと考えられる．

(新鮮度の例) マラソンレースの場合,「通過点 A」の映像が繰り返し放映されると，出現回数の増加とともに，その価値は減少すると考えられる．一方,「先頭集団」の映像は常に新鮮度が高く，新鮮度は減少しないと考えられる．

時間 t における**配信優先度** $P(d,t)$ は，重要度 $P_i(t)$ と新鮮度 $P_f(n)$ によって計算される．

$$P(d,t) = w_1 P_i(d,t) * w_2 P_f(d,n) \quad (8)$$

ここで，t は評価を行う時刻，d はインデックス中のキーワード，n はキーワードの繰り返し回数，P_f は**重要度**の評価関数，P_f は**新鮮度**の評価関数である．w_1,w_2 は重みである．

5.1.1 重 要 度

一般的に，重要度は番組の構成上，時間経過によって変化する．重要度の評価関数は以下のように定義できる：

$$P_i(d,t) = D \times T \rightarrow [0,1] \quad (9)$$

ここで，D はすべての可能な内容記述の集合，T はすべての時刻の集合である．図 10 は「先頭集団」と「アクシデント」の重要度の定義の例を示している．

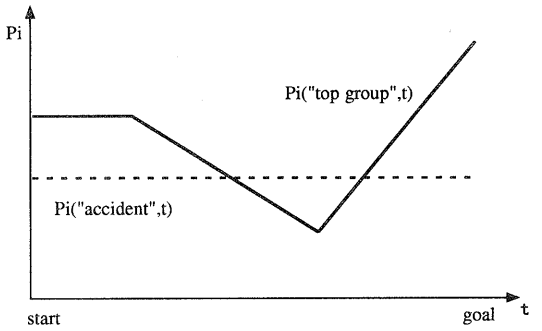

図 10 重要度の例

重要度の評価関数の定義は，番組提供者 (本研究ではディレクタ) の意図を反映するものである．従って，対象とする分野や演出方法によっても違うため，一般的な評価関数の定義を与えることはできない．図 10 に示す評価関数も一例である．

5.1.2 新 鮮 度

一般的に，新鮮度は時間の経過とともに減少すると考えられる．すなわち，繰り返し発生する事象は，注目度が低下すると考えられる．従って，新鮮度の評価関数は以下のように定義できる：

$$P_f(d,n) = D \times N \rightarrow [0,1] \quad (10)$$

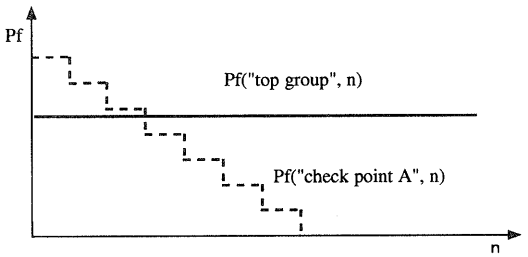

図 11 新鮮度の例

☆ もちろん，重要度が一定の場合もある．

ここで，D はすべての可能な内容記述の集合，N はすべての事象の発生回数の集合である．図 11 は「先頭集団」と「通過点 A」の新鮮度の定義の例を示している．

新鮮度の評価関数の定義も，重要度の場合と同様に番組提供者 (本研究ではディレクタ) の意図を反映するものである．従って，対象とする分野や演出方法によっても違うため，一般的な評価関数の定義を与えることはできない．図 11 に示す評価関数も一例である．

5.2 配信優先度の結合計算

一般的に，事象は個別に発生すると考えられるが，場合によっては同時に発生することも考えられる．例えば，「選手 X」と「棄権」という記述が時間的に同時に起こることがある．この場合は「選手 X が棄権」という意味であるので，2 つの記述を結合した配信優先度を計算する必要がある．配信優先度の結合は以下の式で表される☆：

$$P(a\ b,t) = \alpha * P(a,t) + \beta * P(b,t) \qquad (11)$$

ここで，$\alpha + \beta = 1$ である．

5.3 配信シナリオ

通常番組映像は，あらかじめ決められたシナリオに基づいて構成される．以下により，通常番組映像が表される：

$$S = \{s_1, s_2, \ldots, s_n\} \qquad (12)$$

$$s_i = (D_i, [t_s, t_e)) \qquad (13)$$

ここで，D_i はある内容記述，s_i はあらかじめ決められた通常番組映像の一つである．t_s, t_e はそれぞれ開始時刻，終了時刻である．

ディレクタは通常番組映像を組み合わせることで，一つの番組を作成する．通常番組映像が放映されている途中にスクープ映像が配信された場合は，現在放映中の通常番組映像の配信優先度とスクープ映像の配信優先度を比べて，大きいほうを放映する．もし，スクープ映像が配信されない場合は，ディレクタが構成した通常映像のみが放映されることになる．プロトタイプシステムにおいては，配信シナリオは番組スケルトンとして実装されている．番組スケルトンについては，次章で詳しく述べる．

ここで提案している番組配信方式は，選択肢として複数の映像が与えられたときに，あらかじめ設定された条件からリアルタイムに映像を選択する枠組みを与えるものである．言い換えると，映像データベースからの動的検索機構を実現するものである．

☆ この計算式は発見的であるため，すべての場合に適用することは出来ない．しかし，優先度の結合はアプリケーションに依存するので，アプリケーションごとに定義することになる．

6. プロトタイプシステム ScoopCast

6.1 番組スケルトン

番組は番組スケルトン (Scheduled Program Skeleton: SPS) で記述される．番組スケルトンは，映像素材と放映時間との対の集合からなるタイムテーブルである．基本的には，ディレクタはどのような映像を持ったプロバイダがいるのか，スクープが発生する時間は何時か，あるいはスクープの内容を知らない．従って，番組スケルトンのどの区間には，どのような内容のスクープを挿入可能であるかのみを記述しておく．番組スケルトンは 2 つの要素から構成されている：

- 内容記述：キーワード，場所など
- 制御記述：挿入可能区間，ビデオ映像の品質 (画像サイズ，フレームレート) など

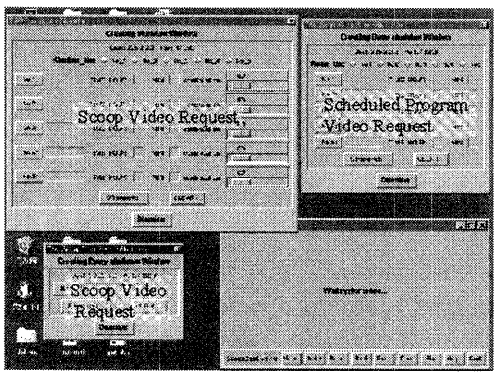

図 12 番組スケルトンのユーザインタフェース

図 12 は番組スケルトンのユーザインタフェースを示している．左側はスクープ映像記述ウィンドウであり，右側は通常映像記述ウィンドウである．

6.2 マルチキャスト・ネットワークを用いたインデックス付き映像の配信

マルチキャスト・ネットワークは不特定多数のユーザがビデオ映像の送受信を行うのに適した環境である．同時に複数のユーザにビデオ映像を配信する場合を考えると，通常のネットワークではトラフィック量が莫大になる．しかし，マルチキャスト・ネットワーク環境であれば，トラフィック量の問題が解決されている．近年，マルチキャスト・ネットワークは Vic(video conference tool)[15] などのビデオ会議システムにおいても注目を集めている．

本研究では，ビデオ配信にはマルチキャスト方式を用いている．この方式では，プロバイダ，ディレクタ，および視聴者がそれぞれネットワークにアクセスし，ビデオ映像を受信および送信する．配信されるデータ

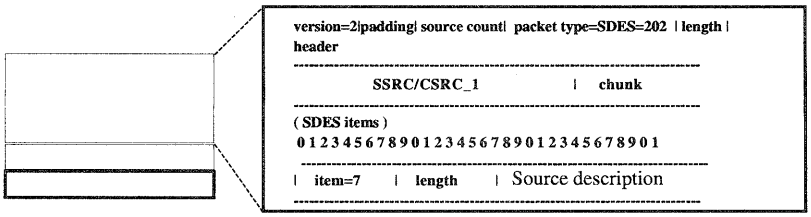

図 13 RTCP パケットの SDES エレメント

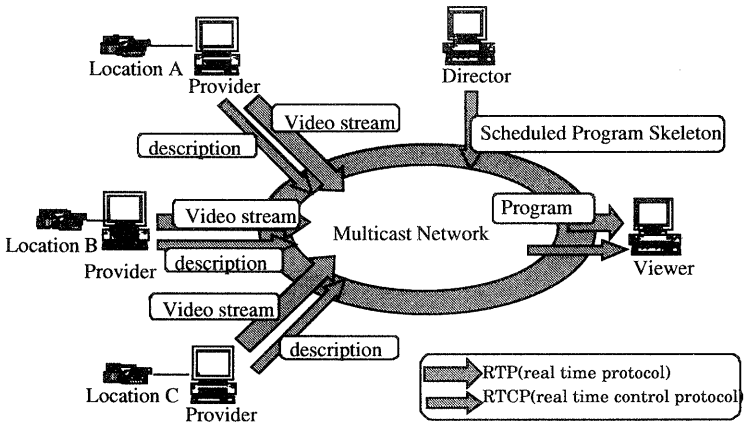

図 14 システム構成

は，映像だけでなく，付与されたインデックスも同時に配信する．インデックスの配信にはRTCPパケット(Real-time Control Protocol packets)[16]を用いる．

一般的には，マルチキャスト・ネットワークではRTP (Real-time Transport Protocol) が用いられており，RTCPパケットが常時転送されている．従って，インデックスをパケットに埋め込むことが出来れば，常時どこでも参照することが可能である．我々は，RTCPパケットのSDES (source description) エレメントをインデックス記述に用いている．図13は，RTCPのSDESを示している．

6.3 システム概要

我々は，これまでに述べた手法によるプロトタイプシステムを開発している．複数のプロバイダから配信されたビデオ映像は，ディレクタが記述した番組スケルトンに基づいて，配信優先度が計算され配送される．システムの概要は以下の通りである：

- 番組スケルトンによる配信優先度の計算とスクープ映像の自動配信
- ライブ映像のリアルタイム配信処理
- ディレクタから配信された複数のビデオ映像のフィルタリング

システムの実装では，ディレクタは番組スケルトンをマルチキャスト・ネットワークに配信する．各プロバイダ側では配信されてきた番組スケルトンを保持しておく．プロバイダの配信機器は，保持している番組スケルトンと内容記述を比較し，ライブ映像をディレクタに自動送信する．ディレクタはRTCPを，プロバイダはRTPを用いている．システムの概要を図14に示す．

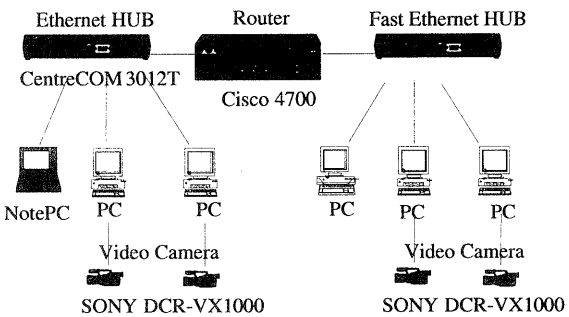

図 15 機器構成

ScoopCastではLawrence Berkeley National Laboratory の Network Research Group が開発したVic[17]を用いている．番組スケルトンはSDESにより実装されている．SDESパケットは，定期的にネット

ワークに送信されるパケットであり，アプリケーション独自に拡張することが可能である．Vic における SDES の利用は，送信元の名前および電子メールアドレス等の情報を付与することが目的となっており，ビデオ映像自身の内容記述や制御のためには用いられていない．本システムでは，Vic に以下の 2 つの機能を追加し拡張を行っている．

- ディレクタによる番組スケルトンの SDES パケット送信
- プロバイダによる SPS パケット受信

ディレクタの番組スケルトン送信機能として，番組スケルトン作成用ユーザインタフェースを Tk ウィジェットを利用して作成している．入力された番組スケルトンは SDES パケットに整形され，ネットワークに配信される．プロバイダの受信機能として，番組スケルトンを制御信号として理解し，条件に従って動作する機能を追加した．番組スケルトンの時間管理は OS のシステムコールを利用している．図 15 に機器構成を示す．

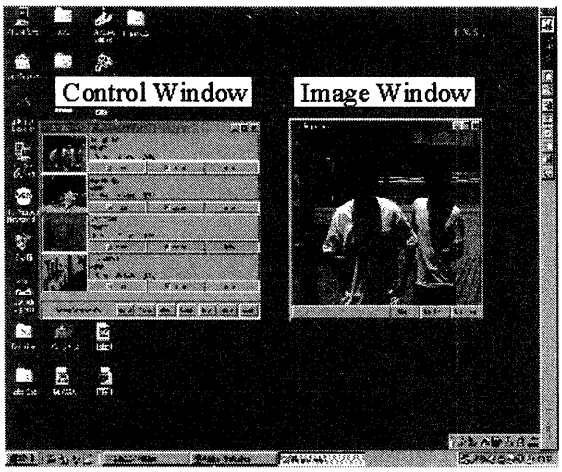

図 16 ビューワーの画面例

視聴者は，ビューワーを用いて，ディレクタの記述した番組スケルトンによって動的に編集された映像ストリームを視聴する．図 16 にビューワーの画面例を示す．視聴者は，マルチキャスト・ネットワークにより，複数のディレクタから配信されるいくつかの映像を選択的に視聴することも可能である．

6.4 評　　価

現在，ScoopCast システムについてはプロトタイプ 1.0 版の開発を完了してる．システムの評価実験については，全ての項目については検討を終えていないが，以下の方針で評価を行なう予定である．

評価実験については，大きく 2 つの項目に分けられる．一つはリアルタイムインデックス生成，もう一つは番組スケルトンを用いた番組の生成 (映像の切り替え) である．

リアルタイムインデックス生成については，4.3.2 節の例に用いた，1998 年 5 月 22 日のプロ野球「阪神対横浜」の毎日放送ラジオでの 1 回裏の実況音声 (約 10 分間) を用いて，音声認識と内容記述についての評価実験を行った．この実験であらかじめ登録しておいたキーワードの数は 108 個である．また，生成されたインデックスは 59 個であった．認識の結果としては，生成されたインデックスのうち約 85%が正しく，残りの約 15%は誤認識であった．この実験では，一般のアナウンサーの実況を入力としているが，インデックスの生成を前提とした実況を用いれば，さらに認識率が向上すると考えられる．

動的番組生成については，番組スケルトンの動作を確認したのみであり，広域ネットワーク環境での実験は完了していない．ただし，マルチキャスト・ネットワークでの動作確認は完了している．

7. おわりに

本論文では，専門家でないユーザがインターネットによる放送を用いて，ビデオ映像の編集，配信，およびフィルタリングをおこなう環境の提案を行った．ビデオ映像配信のための番組スケルトンを設計し，複数のライブ映像が同時に配信された場合の優先順位の決定方法と制御方法を開発した．また，提案する方式を検証するために，マルチキャスト・ネットワーク上に実験環境を構築し，プロトタイプシステム ScoopCast の開発を行った．ScoopCast はマルチキャスト・ネットワークを用いた映像配信のフレームワークの実験環境として構築されている．現バージョンのプロトタイプでは，リアルタイムなインデックス生成，および動的番組生成について基本的な動作確認を完了している．今後の課題としては，インデックス生成の定量的な評価，および広域ネットワークでの映像配信実験が挙げられる．

今後，ネットワークがさらに普及し，家庭内での高速ネットワークの利用が可能になることが予想される．また，ビデオカメラ，携帯端末などのデジタル機器が一層普及することにより，新しい形態のビデオ映像ストリーム配信が現実のものになると考えられる．

謝辞 本研究において初期の段階からの議論，およびプロトタイプ作成にご協力いただいた神戸製鋼所の木俵豊氏，日本電気株式会社の飯嶋亨氏に深く感謝致

します．著者の一部（田中）は本研究において，一部，文部省重点領域研究（課題番号 08244103）および，日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」によっています．ここに記して謝意を表します．

参考文献

1) 角谷和俊, 宮部義幸: 放送型情報配信のためのモデルとシステム, 情報処理学会論文誌, Vol. 40, No. SIG8(TOD4), pp. 141–157 (1999).
2) 川口知昭, 土居明弘: マルチキャストネット通信における映像と付加情報の同期送信方式, 電子情報通信学会春季全国大会, pp. B7–146 (1997).
3) 川口知昭, 土居明弘: マルチキャストネット通信における映像と付加情報の同期表示方式, 電子情報通信学会秋季全国大会, pp. B7–76 (1997).
4) Sumiya, K., Kawaguchi, T., Doi, A., Tanaka, K. and Uehara, K.: ScoopCast: Dynamic Video Production and Delivery from Indexed Live Video Stream, *Proc. of International Computer Science Conference*, IEEE Hong Kong, Lecture Notes in Computer Science 1749, Springer-Verlag, pp. 133–145 (1999).
5) Harada, K., Tanaka, E., Ogawa, R. and Hara, Y.: ANECNODE: A Multimedia Storyboarding System with Seamless Authoring Support, *Proc. of ACM Multimedia'96*, pp. 341–351 (1996).
6) Baecher, R., Rosenthal, A. J., Friedlander, N., Smith, E. and Cohen, A.: A Multimedia System for Authoring Motion Pictures, *Proc. of ACM Multimedia'96*, pp. 31–42 (1996).
7) Kamahara, J., kaneda, T., Ikezawa, M., Shimojo, S., Nishio, S. and Miyahara, H.: A New Scenario Language for Analysis and Synthesis of the TV news Program, *Proc. of the 4th International Conference on Multimedia Modeling'97*, pp. 85–88 (1997).
8) Ueda, H., Hayashi, M. and Kurihara, T.: DeskTop TV Program Creation - TVML (TV program Making Language) Editor -, *ACM Multimedia'98 State of the Art Demos* (1998).
9) Hayashi, M., Ueda, H. and Kurihara, T.: TVML (TV program Making Language) - Automatic TV Program Generation from Text-based Script -, *Proc. of Imagina'99*, pp. 31–42 (1999).
10) Jain, R. and Gupta, A.: PRAJA Presence Technology, *PRAJA white paper* (1998).
11) Shahraray, B. and Gibbon, D. C.: Automated Authoring of Hypermedia Documents of Video Programs, *Proc. of ACM Multimedia'95*, pp. 401–409 (1995).
12) Ariki, Y. and Teranishi, T.: Indexing and Classification of TV News Articles based on Telop Recognition, *Proc. of 4th Int'l Conference on Document Analysis and Recognition '97*, pp. 422–427 (1997).
13) 赤迫貴行, 飯嶋亨, 角谷和俊, 田中克己: 映像データのリアルタイム内容記述方式とその実装, 電子情報通信学会データ工学ワークショップ (DEWS'99) 論文集 (1999).
14) 川口知昭, 土居明弘, 角谷和俊, 田中克己: ScoopCast：ライブ映像ストリームによる動的番組編集システム, 情報処理学会研究報告, 98-DBS-116 (1998).
15) MacCanne, S. and Jacobson, V.: Vic: A Flexible Framework for Packet Video, *Proc. of ACM Multimedia'95*, pp. 511–522 (1995).
16) RTP: RTP/RTCP:RFC1889.
17) LBL: *http://www-nrg.ee.lbl.gov/*.

(平成 11 年 9 月 22 日受付)
(平成 11 年 12 月 27 日採録)

(担当編集委員 **加藤俊一**)

角谷 和俊（正会員）

1988 神戸大学大学院工学研究科修士課程修了．同年松下電器産業株式会社入社．ソフトウェア開発環境，マルチメディアデータベース，データ放送の研究開発に従事．1998 神戸大学大学院自然科学研究科博士後期課程（情報メディア科学専攻）修了．1999 神戸大学都市安全研究センター都市情報システム研究分野（工学部情報知能工学科 兼担）講師．博士（工学）．情報処理学会データベースシステム研究会幹事．ACM, IEEE Computer Society, 映像情報メディア学会，地理情報システム学会，システム制御情報学会各会員．

川口 知昭（正会員）

1989 大阪産業大学工学部機械工学科卒業．同年日本電信電話株式会社入社．ネットワーク設計・構築，VOD システムの研究開発に従事．マルチキャスト通信，マルチメディアデータ配信，インターネット放送，ネットワーク管理に興味を持つ．電子情報通信学会会員．

土居 明弘

1995 神戸大学工学部電気工学科卒業．同年日本電信電話株式会社入社．プライベートネットワークシステム設計・構築に従事．映像圧縮、映像配信、マルチメディアデータベースに興味を持つ．

赤迫 孝行

1997 神戸大学工学部情報知能工学卒業，1999 同大学大学院修士課程修了．同年 NTT コミュニケーションウェア株式会社入社．データベース，ハイパーメディアに興味を持つ．

田中 克己（正会員）

1974 京都大学工学部情報工学科卒業，1976 同大学大学院修士課程修了．1979 神戸大学教養部助手，1986 同大学工学部助教授．1994 同大学工学部教授 (情報知能工学専攻)．1995 同大学大学院自然科学研究科 (現在，情報メディア科学専攻) 専任教授，現在に至る．工学博士．主にデータベースの研究に従事．96 年度より通信・放送機構「次世代デジタル映像通信の研究開発」の研究統括責任者，文部省科研費重点領域研究「分散発展型データベースシステム技術の研究」の研究代表者，神戸マルチメディアインターネット協議会会長，人工知能学会，IEEE Computer Society，ACM 等各会員．