データベースシステム 115－11
情 報 学 基 礎 49－11
（1998. 5. 14）

イベント-アクション・モデルに基づく動画像検索システム STRIKE

牛尼 剛聡　広部 一弥　酒井 宏治
孫 魯英　渡邊 豊英

名古屋大学大学院工学研究科情報工学専攻
ushiama@watanabe.nuie.nagoya-u.ac.jp

本稿では，様々な利用者の視点にしたがってシーンを検索可能な動画像検索システム STRIKE について述べる．従来の動画像検索システムでは，検索対象であるシーンに直接キーワードなどの属性を付加し，指定された属性を持つシーンを検索する手法が一般的に用いられてきた．しかし，この手法では，キーワードの語彙やシーンの捉え方がデータベース作成者の視点に依存するために，検索対象となるシーンはあらかじめ限定されており，利用者の視点に基づいた検索を行なうことが困難であった．そこで，我々は，利用者の検索要求に基づいて動的にシーンを構成可能なイベント-アクション・モデルを開発した．本モデルでは，動画像に記録された出来事をイベントとして表現し，動画像のフレームにイベントをインデックスとして対応づけることにより，動画像の内容をイベント系列として表現する．ユーザはイベント系列パターン（アクション）を指定し，システムはイベント系列中からアクションに含まれる部分系列を抽出することによりシーン検索を実現する．

STRIKE: A Movie Retrieval System Based on Event-Action Model

Taketoshi USHIAMA　Kazuya HIROBE　Koji SAKAI
Luying SUN　Toyohide WATANABE

Department of Information Engineering, Graduate School of Engineering, Nagoya University

This paper describes a movie retrieval system that enables users to retrieve scenes according to various kinds of viewpoints. In most of traditional movie retrieval systems, contents of scene are modeled as sets of attributes (such as keywords), and user retrieves scenes using these attributes. In this approach, it is hard for users to retrieve scenes according to view of users, because these attributes are prepared based on viewpoints of database manager. We developed event-action model that enables to compose scenes dynamically according to viewpoints of user. In this model, a difference between two continuous frames in a movie is represented as an event, and a context of movie is represented as a sequence of events. For retrievals, users specifies sub-sequence of the contexts, and the system extracts specified sub-sequences and converts them to scenes.

1 はじめに

動画像は対象世界の動的な側面を直観的にわかりやすく表現可能であるため，対象世界における活動の伝達，記録，分析等を目的として，さまざまな分野で利用されている．近年，計算機上で大量の動画像を効率的に管理し，効果的に利用する要求が高まり，動画像データベース・システムに関する研究が活発化している [1]．動画像は静止画像（フレーム）の時系列であり，利用者が興味のある事柄（意味的なまとまり）を表現するフレーム部分系列をシーンと呼ぶ．動画像を利用する際には，動画像全体ではなく，特定のシーンを必要とする場合が多い．そこで，動画像データベース・システムには，シーン検索機構の提供が期待されている．

動画像を捉える視点は利用者の目的や興味によって異なる．利用者の視点の相違に基づいて，動画像中には多種多様なシーンを考えることができる．動画像データベースでは，データ共有の立場から利用者の視点に基づいて適切なシーンを検索する必要がある．シーン検索時に考慮しなければならない視点の相違として以下を考えることができる．

- **着目する実体の相違:** 動画像は対象世界上の複数の実体を表現可能である．利用者は目的に応じて，特定の実体に着目し，他の実体を無視する．同一の動画像であっても，着目する実体が異なれば，利用者が動画像中に想定するシーンは異なる．
- **活動の複合レベルの相違:** シーンが対象世界上の活動を表しているとき，活動を構成する更に細かい活動を考えることができる．例えば，野球中継におけるホームランのシーンは，投手が投球するシーン，打者がボールを打つシーン等を含む．
- **活動を捉える概念レベルの相違:** 同一の活動を複数の異なる概念レベルから捉えることができる．例えば，野球中継におけるホームランのシーンは，ヒットのシーンと考えることもできる．

これまでに提案されてきた画像検索手法は，特徴抽出に基づく手法と属性に基づく手法とに大別できる [2]．特徴抽出に基づく手法では，画像処理を用いて色やテクスチャといった物理情報を自動抽出し，抽出された物理情報に基づいた検索を行なう．一方，属性に基づく手法では，人間がキーワードなどの属性情報をインデックスとして付加し，付加された属性情報に基づいた検索を行なう．特定の応用分野を除いては，画像処理を用いて抽出可能な物理情報は抽象化レベルが低く，画像の持つ意味内容とのセマンティック・ギャップが大きい [3]．したがって，意味内容に基づいた検索のためには，属性に基づく手法を採用するのが一般的である [2]．

属性に基づく手法では，データベース作成者がデータベース作成時に画像に対して属性情報を付加し，その属性情報を用いて利用者は検索要求を表現する．しかし，データベース作成時のデータベース作成者の視点と，検索時の利用者の視点は必ずしも一致しないため，単純なキーワード・マッチングによる手法では利用者の視点を反映した検索が困難である．利用者の視点を反映した検索を実現するためには，データベース作成者が一般的かつ客観的な基準の下に基礎的な属性情報を付加し，利用者がそれらの属性情報を用いて自分の視点を表現可能なデータモデルが必要となる．

これまでに，属性に基づいてシーン検索を行なういくつかのデータモデルが提案されている [4–10]．これらの基本的なアプローチは，検索対象であるシーンに対して直接キーワード等の属性情報をインデックスとして付加するものである．すなわち，このアプローチでは，動画像が表現する対象世界上の動的な側面における意味的なまとまりを個別的なモノとしてモデル化する．対象世界を個別的なモノの集まりとしてモデル化するアプローチは，ER モデル [11] をはじめとする意味データモデル [12] やオブジェクト指向データモデル [13] 等，従来型のデータベース・システムでのデータモデルに広く採用されてきた．従来型のデータベース・システムでは対象世界の静的な側面（スナップショット）に着目する場合が多く，このアプローチは対象世界上の実体をモノとして自然にモデル化可能であった．しかし，動画像データベース・システムでは，動画像が表現する対象世界の動的な側面に着目する必要があり，それらは連続的かつ多重的であるために客観的な個別化が困難である．そのために，上記のアプローチでは利用者の視点を反映可能なデータモデルを提供することが困難である．

現在，我々は，野球中継を対象とした動画像データベース・システム STRIKE(STream data Retrieval based on Indexing with Key Events) を開発中である．本稿では，STRIKE において，利用者の多種多様な視点に基づくシーン検索を実現するデータモデルであるイベント - アクション・モデルを提案する．イベント - アクション・モデルは，シーン単位のインデキシングではなく，動画像中で発生する出来事を表すイベントをフレーム単位でインデキシングする．検索要求はイベントの系列パターンとして記述され，システムは利用者の視点に応じて動的にシーンを構成可能である．

2 関連する研究

動画像中の活動の複合レベルの相違に対処するために，シーンを時間的な包含関係に基づいて階層化する手法が提案されている [4–6]．Little ら [4] は，動画像を時間軸上で 2 段階に分割し，階層構造としてモデル化する．このモデルでは，動画像は時間軸上で排他的に分割されることを前提としているため，時間的に重なり合うシーンを表現することが困難である．さらに，階層の数が 3 つに限定されているために，3 段階以上の複合レベルを表現できない．Duda らの手法 [5] では，時間的

に重なり合うシーンから構成される任意の深さのシーン階層を許す．この手法では，任意の複合レベルにおけるシーンを表現可能である．さらに，大本らの手法 [6] は任意の深さのシーン階層に加え，属性値の概念階層知識と属性構造の包含関係を利用することにより，異なる概念レベルのシーンを表現可能である．しかし，これらの任意の深さのシーン階層を利用する上記のアプローチではシーンに関するスキーマを提供していないため，利用者が要求するシーンを検索可能であるか，また要求するシーンの検索のためにどのような属性指定を行なえば良いかを利用者が判断することは困難である．

田渕ら [7] は動画像をカットと呼ぶ原子的なシーンに分割し，カットの系列パターンを利用したシーン検索手法を提案している．例えば，大相撲放送における取組みのシーンは，「仕切り」，「対戦」，「決着」，「礼」，「力水」，「花道」というタイトルを持つカットの系列として検索する．しかし，この手法では，時間的に重なり合うカットはカット系列として扱うことができないため，カットは動画像の排他分割であることが必要である．したがって，動画像中で発生する活動が多重的である場合に対処できない．

Day ら [8] と Li ら [9] は動画像中の実体の空間的な関係に基づいてシーンを検索する手法を提案している．これらの手法はフレーム中の物体を外接矩形で近似し，矩形間の空間的な関係に基づいて原子的なシーンを定義し，それらの原子的なシーンの関係に基づいて利用者の要求するシーンを検索する．しかし，これらは動画像中の物理的な実体のみを対象としており，概念的な実体に関しては考慮していない．また，これらはシーンの繰返し等を表現する能力がないために，検索要求の表現能力が低い．

3 アプローチ

3.1 動画像とシーン候補

シーンを検索するためには，データベース・システムは動画像に含まれるシーンを列挙可能でなければならない．しかし，観測者の目的や興味によって，動画像はさまざまな視点から内容の意味的なまとまりを考えることができるため，動画像に含まれる全てのシーンを列挙することは困難である．そこで，動画像中の連続フレーム部分系列がシーンであるための必要条件を考え，必要条件を満たす系列をシーン候補とする．動画像のシーン集合はシーン候補集合の部分集合であるため，それぞれのシーン候補が指定可能であれば，利用者が要求するシーンを検索可能である．

動画像は静止画像（フレーム）の系列であり，シーンは動画像中の連続部分系列である．したがって，シーン検索とは，動画像から内容に基づいて開始フレームと終了フレームを決定する問題である．フレーム系列の中

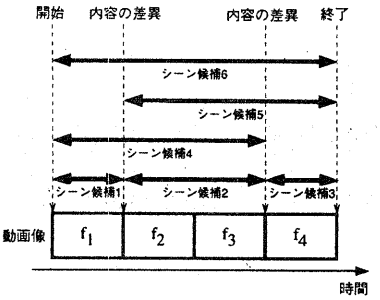

図 1: シーン候補の例

から特定のフレームを決定するためには，対象とするフレームを隣接するフレームから内容に基づいて区別可能でなければならない．したがって，動画像 $mv = \langle f_1, \cdots, f_n \rangle$ 中の任意の部分系列 $\langle f_i, \cdots, f_j \rangle$ $(1 \leq i \leq j \leq n)$ がシーンとなるために，以下の必要条件を考えることができる．

1. フレーム f_i が動画像 mv の先頭フレームである．あるいは，f_i の内容と直前のフレーム f_{i-1} の内容に差異が存在する．
2. 終了フレーム f_j が動画像 mv の最終フレームである．あるいは，f_j の内容と直後のフレーム f_{j+1} の内容に差異が存在する．

直観的に，上記の条件はシーンの開始点と終了点には動画像内容の区切りが存在していなければならないことを表している．

シーンの必要条件から，フレーム間の内容の差異に基づいてシーン候補を列挙可能である．いま，動画像 mv に含まれる連続するフレーム間の差異の個数を k $(0 \leq k \leq n-1)$ とすると，mv 中には ${}_2C_{k+2}$ 個のシーンの候補が存在する．例えば，図 1に示すように，4 フレームから構成される動画像中に 2 つの差異が存在する場合には 6 種類のシーン候補が存在する．なお，図中の長方形はフレームを表し，両端に矢印を持つ直線がシーン候補を表している．

3.2 差異の抽象化とイベント

動画像内容を捉える最も原子的な視点は，フレームを構成する画素に着目するものである．画素レベルではほとんど全ての連続フレーム間に差異が存在する．しかし，画素レベルの差異は，抽象度が低いために，必ずしも動画像の意味的な内容の差異を表現するとは限らない．そこで，差異を抽象化し，動画像中の実体に関するイベントとしてモデル化する．

実体は対象世界の静的な側面を表現する概念である．実体とは対象世界上で識別可能であり，かつ対象世界上の任意の時刻において，その状態を考えることが可能なモノである．対象世界上の実体は，物理実体と概念実体

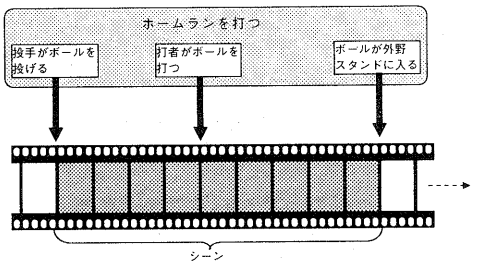

図 2: 概念図

に分類可能である．物理実体とは対象世界上の物理的な個別物体に対応づけ可能な実体である．一方，概念実体は対象世界上の物理的な個別物体に対応づけることができず，その状態が他の実体の状態に依存する実体である．例えば，野球の試合において，人物やボールなどは物理実体であり，得点やアウト・カウントなどは概念実体である．

動画像においては，同一フレーム内に存在する画素が実体の状態を表現している．したがって，動画像における連続フレーム間の画素レベルの差異は，実体レベルでは実体の特性の変化として抽象化可能である．実体の特性の変化，または実体の特性を変化させる実体間の相互作用をイベントと呼ぶ．例えば，野球の試合においては，「投手がボールを投げる」，「アウト・カウントが2になる」などはイベントである．

3.3 イベント系列とシーン検索

シーン検索では，利用者は検索したいシーンの内容を検索要求として与え，システムは与えられた内容を表現するシーンを検索結果として返す．したがって，シーン検索のために，データベース内に含まれるシーン候補の内容を表現する必要がある．我々は，シーン候補の内容をシーン候補内で発生したイベントの系列として表現する．例えば，図2に示すように，あるシーン候補内に「投手がボールを投げる」「打者がボールを打つ」「ボールが外野スタンドに入る」というイベント系列が存在する場合，このシーン候補は「ホームランを打つシーン」であると考えることができる．したがって，イベントが動画像内のフレームに対応づけられていれば，利用者がイベント系列を指定することにより，要求するシーンを検索することが可能である．

上記のアプローチに基づくシーン検索システムの基本的な構造を図3に示す．本システムは動画像を蓄えているデータベースとインデックスを蓄えているデータベースを持つ．検索時には，利用者は検索要求をアクションと呼ばれるイベント系列パターンとして指定する．インデックスとして付加されたイベント系列から，アクションに一致する部分イベント系列を取り出す．得られた部分イベント系列を開始フレーム番号と終了フレーム番号の対として表される,シーン指定に変換する．シーン指定に基づいて動画像データベースからシーンを構成し，ユーザに検索結果として返す．

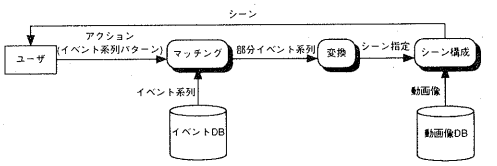

図 3: シーン検索

4 イベント-アクション・モデル

イベント-アクション・モデルは動画像中のシーンを検索するためのデータモデルである．本モデルは，1) 動画像内容の表現，2) シーン検索のための演算，3) 動画像内容のスキーマ表現，を実現するための枠組を提供する．

4.1 系列の表記

イベント-アクション・モデルを定義するために，本稿で使用する系列の表記と部分系列の概念を導入する．

集合 $S = \{a_i \mid 1 \leq i \leq n\}$ 上に $a_i \prec a_j$ $(i \leq j)$ を満たす全順序関係 $\prec$ が定義されているとき，S と $\prec$ によって与えられる系列を，$(S, \prec)$ または $\langle a_1, \cdots, a_n \rangle$ と表記する．後者の表記は系列を構成する個々の要素に着目するときに用いる．

2つの系列 $(S, \prec)$ および $(S', \prec)$ に対して，S' が S の部分集合であるとき，$(S', \prec)$ を $(S, \prec)$ の部分系列と呼ぶ．例えば，系列 $\langle a_1, a_2, a_3 \rangle$ の全ての部分系列から構成される集合は $\{\langle a_1, a_2, a_3 \rangle, \langle a_1, a_2 \rangle, \langle a_1, a_3 \rangle, \langle a_2, a_3 \rangle, \langle a_1 \rangle, \langle a_2 \rangle, \langle a_3 \rangle\}$ である．また，以下の条件を満たすとき部分系列 $(S', \prec)$ を不完全部分系列と呼ぶ．

$$a_1 \prec a_3 \prec a_2 \ (a_1, a_2 \in S', a_3 \in S, a_3 \notin S')$$

例えば，系列 $\langle a_1, a_2, a_3 \rangle$ の不完全部分系列は $\langle a_1, a_3 \rangle$ である．不完全部分系列でない部分系列を完全部分系列と呼ぶ．

4.2 動画像内容の表現

実体は文字列として表現される．同一の文字列は同一の実体であるとみなす．動画像中にはさまざまな実体を観測可能であるが，検索時に注目する実体のドメインを規定するために実体型を用いる．実体型は2項 (tn, EN) として定義する．ここで，tn は型名であり，EN は実体の集合である．実体 $en \in EN$ を tn のインスタンスと呼ぶ．野球の試合における実体の例を表1に

表 1: 野球の試合における実体の例

型名	実体
選手	"イチロー", "吉田"等
塁	"本塁", "1塁", "2塁", "3塁"
ボール	"ボール"
グラウンド	"フェア・グラウンド", "ファール・グラウンド"
観客席	"内野席", "外野席"
試合	"オリックス対ヤクルト"等
カウント	"ストライク・カウント", "ボール・カウント", "アウト・カウント", "イニング", "得点"
数値	"1", "2"等

示す．表中で「選手」「塁」「ボール」「グラウンド」「観客席」は物理実体の型であり，「試合」「カウント」「数値」は概念実体の型である．実体型「ボール」はただ一つのインスタンス"ボール"をもつ．これは，競技で利用されるボールは常に一つであるという仮定に基づいている．

イベントは2項 $(id, [en_1, \cdots, en_n])$ として表現する．ここで，id は識別子であり，en_i $(1 \leq i \leq n)$ は実体である．識別子はそれぞれのイベントを区別し，引数 $[en_1, \cdots, en_n]$ はイベントに関係する実体を表す．イベントのドメインを規定するためにイベント型を導入する．イベント型は3項 $(tn, [tn_1, \cdots, tn_n], EV)$ である．ここで，tn は型名であり，tn_i $(1 \leq i \leq n)$ は実体型の型名であり，EV はイベントの集合である．EV の要素を型 tn のインスタンスと呼ぶ．$[tn_1, \cdots, tn_n]$ はインスタンスの引数の型を指定する．すなわち，EV の要素の i 番目の引数は実体型 tn_i のインスタンスでなければならない．野球の試合におけるイベント型の例を表2に示す．なお，表中ではイベント型の表現を，イベント集合を省略した2項(イベント名,引数の型宣言)としている．

動画像内容をイベント系列として表現する．動画像内容を表現するためのイベント系列をコンテクストと呼ぶ．コンテクストを2項 $(EV, \prec)$ で表す．ここで，EV はイベントの集合であり，$\prec$ は，集合 EV と特殊記号の集合 $\{\mathfrak{c}, \$\}$ の和集合上の全順序関係であり，イベントが発生した時間的な前後関係を表す．特殊記号 $\mathfrak{c}$ はイベント系列の先頭を表し，$\$$ はイベント系列の再後尾を表す．表2で示されているイベント型が存在する場合のコンテクストの例を表3に示す．表3では，テレビ番組として放送された野球の試合における，試合開始から2打席分を対象とした．

シーン検索を行なうために，コンテクストを構成するそれぞれのイベントを動画像のフレームに対応づける．この対応づけを表す関数をインデックス関数と呼ぶ．動画像 $(F, \Rightarrow)$ とコンテクスト $(EV, \prec)$ が存在するとき，インデックス関数を以下の条件を満たす関数 ψ : $EV \longrightarrow F$ として定義する．

$\forall ev_1, ev_2 \in EV$ に対して，

$ev_1 \prec ev_2$ ならば $\psi(ev_1) \Rightarrow \psi(ev_2)$

上記の条件は，インデックスを介してフレームに対応づけられたイベント間の順序関係はフレーム間の順序関係によって保持されることを示している．なお，特殊記号 $\mathfrak{c}$ は動画像の先頭フレームに対応づけ，$\$$ は最終フレームに対応づける．

表 2: 野球の試合におけるイベントの例

イベントの型	説明
(throw,[選手, ボール])	選手がボールを投げる
(catch,[選手, ボール])	選手がボールを補給する
(drop,[選手, ボール])	選手がボールを落す
(touch-to,[選手, ボール, 選手])	選手がボールで選手に触れる
(touch,[選手, ボール])	選手がボールに当たる
(reach,[選手, 塁])	選手が塁に触れる
(bat,[選手, ボール])	選手がボールを打つ
(is-batter,[選手])	選手が打者になる
(is-out,[選手])	選手がアウトになる
(is-safe,[選手])	選手がセーフになる
(bound-in,[ボール, フィールド])	ボールが地面と接触する
(enter,[ボール, スタンド])	ボールがスタンドに入る
(strike,[試合])	ストライク
(ball,[試合])	ボール
(foul,[試合])	ファウル
(four-ball,[試合])	四球
(change,[試合])	チェンジになる
(playball,[試合])	プレイボール
(interrapt,[試合])	中断開始
(play,[試合])	試合再開
(game-set,[試合])	ゲームセット
(is-set-as,[カウント, 数値])	カウントを数値にする

以上の定義に基づき，動画像データベースを5項 (ENT, EVT, cn, mv, ψ) として定義する．ここで，ENT は実体型の集合，EVT はイベント型の集合，cn はコンテクスト，mv は動画像，ψ はインデックス関数である．

4.3 シーン検索

本手法では，コンテクストから利用者の興味のあるイベント部分系列を抽出し，インデックス関数を介してフレーム系列（シーン）に変換し，シーン検索を実現する．本節では，イベント系列中の部分系列の表現形式と，シーン検索に必要な演算を導入する．

4.3.1 アクション

イベントは識別子によって識別性を与えられているが，検索時に利用者が興味を持つのはイベントが属する型や引数の種類である．そこで，イベント集合を表すイベント・パターンを導入する．イベント・パターンは $tn[ES_1, \cdots, ES_n]$ という形式で表現する．ここで，tn はイベント型の型名であり，ES_i は実体の集合である．ES_i に実体型の型名が与えられたときは，その型

表 3: コンテクストの例

イベント	イベント型
(#1,["試合 1"])	play
(#2,["ストライク・カウント","0"])	is-set-as
(#3,["ボール・カウント","0"])	is-set-as
(#4,["イニング","1"])	is-set-as
(#5,["打者 1"])	is-batter
(#6,["投手","ボール"])	throw
(#7,["補手","ボール"])	catch
(#8,["試合 1"])	strike
(#9,["ストライク・カウント","1"])	is-set-as
(#10,["補手","ボール"])	throw
(#11,["投手","ボール"])	catch
(#12,["投手","ボール"])	throw
(#13,["補手","ボール"])	catch
(#14,["試合 1"])	ball
(#15,["ボール・カウント","1"])	is-set-as
(#16,["補手","ボール"])	throw
(#17,["投手","ボール"])	catch
(#18,["投手","ボール"])	throw
(#19,["打者 1","ボール"])	bat
(#20,["ボール","フェアグラウンド"])	bound-in
(#21,["ボール","フェアグラウンド"])	bound-in
(#22,["右翼手","ボール"])	catch
(#23,["右翼手","ボール"])	throw
(#24,["打者 1","1 塁"])	reach
(#25,["打者 1"])	is-safe
(#26,["投手","ボール"])	catch
(#27,["ストライク・カウント","0"])	is-set-as
(#28,["ボール・カウント","0"])	is-set-as
(#29,["打者 2"])	is-batter
(#30,["投手","ボール"])	throw
(#31,["打者 2","ボール"])	bat
(#32,["ボール","フェアグラウンド"])	bound-in
(#33,["投手","ボール"])	catch
(#34,["投手","ボール"])	throw
(#35,["打者 1","2 塁"])	reach
(#36,["打者 1"])	is-safe
(#37,["游撃手","ボール"])	catch
(#38,["打者 2","1 塁"])	reach
(#39,["打者 2"])	is-safe

のインスタンス集合を表すものとする．このイベントパターンは以下の条件を共に満たすイベント ev から構成されるイベント集合を表す．

1. ev はイベント型 tn のインスタンスである．
2. ev の i 番目の引数が ES_i の要素である．

例えば，reach[選手,2 塁] は選手が 2 塁に触れたことを表す全てのイベントから構成される集合を表すイベント・パターンである．なお，**top**[] はイベント集合 {c | } をを表すイベント・パターンとし，**last**[] はイベント集合 {$} を表すイベント・パターンとする．

イベント系列パターンはイベント・パターン集合上の正則表現と定義する．イベント・パターンはイベントの集合であるため，イベント系列パターンはイベント系列の集合を表現する．

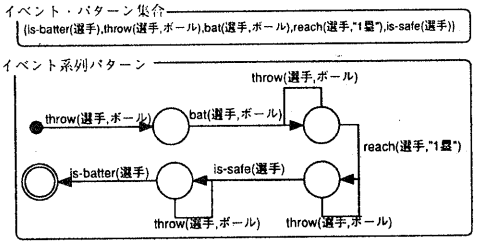

図 4: アクションの例

検索時には，イベント系列を構成するイベント全てに興味があるわけではなく，データベース内に存在する全てのイベント型を考慮してイベント・パターンを記述することは困難である．そこで，興味のあるイベントのみから構成されるイベント系列中でイベント系列パターンを指定可能とするために，アクションを導入する．アクションは 2 項 $(EVP, evsp)$ である．ここで，EVP は対象とするイベントを表すイベント・パターンの集合であり，$evsp$ はイベント系列パターンである．アクションは利用者の原始的な検索要求を表す．1 塁打を表すアクションの例を図 4に示す．図中ではイベント系列パターンを状態遷移図として表現している．黒丸が開始状態であり，二重丸が終了状態である．

4.3.2 検索演算

extract$(evs, evsp)$ はイベント系列 evs に対して，与えられたイベント系列パターン $evsp$ にマッチする完全部分系列の集合を返す演算である．例えば，表 3に示したコンテクスト（イベント系列）を evs とし，イベント系列パターン $evsp$ を throw[投手, ボール] ・ catch[補手, ボール] ・ ball[試合] としたとき，演算 extract$(evs, evsp)$ の結果はイベント系列の集合 $\{\langle(\#12, [投手, ボール]), (\#13, [補手, ボール]), (\#14, [試合1])\rangle\}$ である．

演算 project(evs, EVP) はイベント系列 evs に対して，指定したイベント・パターンの集合 EVP に含まれるイベントのみから構成される部分系列を生成する演算である．演算 project が演算結果として返す部分系列は不完全部分系列となる場合がある．例えば，表 3に示したイベント系列を evs としたとき，演算 project$(evs, \{bat[選手, ボール]\})$ の結果は，イベント系列 $\langle(\#18, [打者1, ボール]), (\#28, [打者2, ボール])\rangle$ である．ここで得られた結果は evs の不完全部分系列である．

上記の 2 演算を利用することにより，アクションが表すイベント系列集合をコンテクストから抽出することができる．すなわち，コンテクスト cn に対するアクション $(EVP, evsp)$ を用いた検索は extract(prpject$(cn, EPT), evsp)$ と実現できる．

シーンは動画像の完全部分系列である．系列内の完全部分系列を指定するためには，完全部分系列の先頭要素と最終要素によって一意に同定可能である．系列 se 中で，先頭要素 a_s，最終要素 a_e から構成される完全部分系列を sequence(a_s, a_e, se) と表記する．また，完全部分系列 se' の先頭要素を s(se') と表記し，最終要素を e(es') と表記する．イベント系列をシーンに変換する関数 scene を導入する．演算 scene(mv, ψ, ES) は集合 $\{\text{seaquence}(\psi(\text{s}(es)), \psi(\text{e}(es)), mv) \mid es \in ES\}$ を返す．ここで，mv は動画像，ψ はインデックス関数，ES はイベント系列の集合である．

シーンを前後の文脈に基づいて指定したい場合がある．たとえば，1 塁打の後に打たれたホームランのシーンを検索するときには，ホームランを表すシーン集合と，検索結果には現れない 1 塁打を表すシーン集合を検索し，シーン間の時間的な関係を利用して条件を満たすホームランのシーンを決定する必要がある．しかし，我々のアプローチでは検索するシーンに含まれるイベント系列をアクションとして指定するため，単にアクションを指定するだけでは前後の文脈を指定することができない．そこで，文脈に基づいたシーンを検索するための演算 join を導入する．この演算は二つのシーン集合の中から，指定された関係を満たす 2 つのシーンに対してシーン演算を施す．すなわち，join(op, rel, SC_1, SC_2) は $\{op(sc_1, sc_2) \mid sc_1 \in SC_1, sc_2 \in SC_2, sc_1\ rel\ sc_2\}$ を返す．ここで，SC_1, SC_2 はシーン集合であり，op はシーン合成演算であり，rel はシーン間の関係である．rel はシーン間の関係である．シーン間の関係としては，シーンは動画像中の時区間であると考えられるため，Allen[14] によって提案された時区間による表現を利用可能である [15]．シーン合成演算としては，与えられた二つのシーンのうちの前者または後者を返す演算，二つのシーンの時区間的な和または積を返す演算を考えることができる．

4.4 コンテクストの一貫性とイベント・スキーマ

イベントの間に依存関係が存在する場合がある．例えば，ボールを投げるのはボールを補球した後であり，打者がボールを打つのは投手が投球した後である．こうした依存関係はコンテクスト内のイベントが満足しなければならない制約と考えることができる．この制約を表現するために，イベント・スキーマを導入する．イベント・スキーマはアクションの集合である．コンテクストは必ずイベント・スキーマを満足しなければならない．つまり，コンテクスト cn に対するイベント・スキーマの構成要素としてアクション $(EPT, evsp)$ が定義されているとき，イベント系列 project(cn, EPT) はイベント系列集合 extract(project$(cn, EPT), evsp)$ の要素でなければならない．野球の試合におけるイベント・スキーマの例を図 5に示す．図では，アクションを状態遷移図として表現している．状態遷移図中の状態の名前はわかり易さのため便宜的に付加したものである．実際の動画像中ではカット切替や撮影範囲が限定されることなどから，動画像中には全てのイベントが記録されていない場合がある．動画像は対象世界の出来事を全て記録しているわけではない．コンテクストがイベント・スキーマを満足するためには，動画像中に記録されていないイベントを挿入しなければならない場合がある．

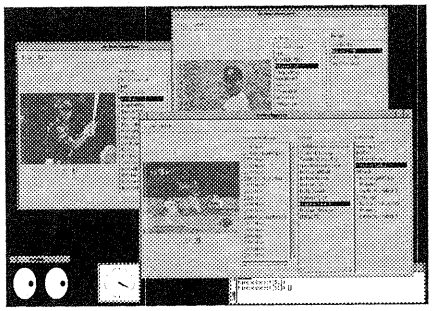

図 6: プロトタイプ・システムの実行画面

5 プロトタイプ・システム

動画像検索システムのプロトタイプを作成した．対象とした動画像はテレビ番組として放送されたものであり，計算機上に JPEG 形式に圧縮した静止画像画像の系列として保存した．コンテクストはイベント・スキーマを満たすように作成し，人手によって動画像に対応づけた．基本的なアクションとそれに対する検索処理は，JAVA 言語の関数として手続き的に記述し，あらかじめシステム内に組み込まれている．図 6にシステムの実行画面のスナップショットを示す．システムは検索用ウィンドウとシーン再生用ウィンドウを提供する．検索用ウインドウはアクションのリストを示し，利用者はリスト中の要素を選択することによりシーン検索が開始される．検索が終了すると，シーン再生用ウインドウが現れる．このウインドウには検索結果となったシーンのリストが示されており，利用者はリスト中のシーンを選択することで，検索結果を確認できる．

6 おわりに

本稿では，イベント - アクション・モデルに基づく動画像検索手法を提案した．イベント - アクション・モデルは，動画像が表現する対象世界の動的な側面をイベント系列として表現することにより，利用者のさまざまな視点に基づく検索要求に対処可能である．

本手法では，多数のイベントをインデックスとして付加する必要がある．例えば，本稿で例示したイベント型に基づいて一つの野球の試合にイベントを付加する場合，発生するイベントは数千に及び，非常にコストが高い．この問題に対処するために，現在，画像処理を利用

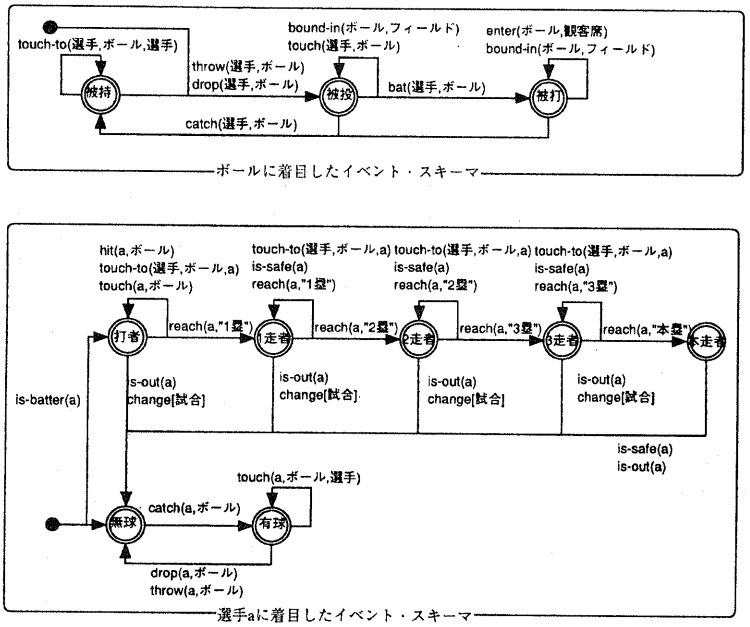

図 5: イベント・スキーマの例

したインデキシング支援機構を開発中である.

実際の動画像においては, 一般に, 多くのイベントが省略される. そこで, イベント・スキーマやヒューリスティックに基づいて省略されたイベントの自動補間機構を開発中である.

参考文献

[1] A.Elmagarmid et al.: *Video Database Systems*, Kuwer Academic Publishers (1997).

[2] Gudivada, V. and Raghavan, V.: Content-Based Image Retrieval Systems, *IEEE Computer*, Vol. 28, No. 9, pp. 18–21 (1995).

[3] 美濃導彦: 知的映像メディア検索技術の動向, 人工知能学会誌, Vol. 11, No. 1, pp. 4–9 (1996).

[4] Little, T. et al.: A Digital On-Demand Video Service Supporting Content-Based Queries, *Proc. of ACM Int. Conf. on Multimedia*, pp. 427–436 (1993).

[5] Duda, A., Weiss, R. and Gifford, D. K.: Content-Based Access to Algebraic Video, *Proc. of International Conference on Multimedia Computing and Systems*, pp. 140–151 (1994).

[6] Oomoto, E. and Tanaka, K.: OVID: Design and Implementation of a Video-Object Database System, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 626–643 (1993).

[7] 田渕仁浩, 村岡洋一: 動画像データベース中の系列データを指定する条件の不完全さを許容できる問い合わせ処理と MeSOD モデル, 信学論 (D-I), Vol. J76-D-I, No. 6, pp. 288–299 (1993).

[8] Day, Y. F., Dağtaş, S. D., Iino, M., Khokhar, A. and Ghafoo, A.: Object-Oriented Conceptual Modeling of Video Data, *Proc. of International Conference on Multimedia Computing and Systems*, pp. 98–105 (1995).

[9] Li, J. S., Özsu, T. and Szafron, D.: Modeling of Video Spatial Relationships in an Object Database Management System, *Proc. of International Workshop Multimedia DBMS*, pp. 124–133 (1996).

[10] Arisawa, H. et al.: Data Model and Architecture of Multimedia Database for Engineering Applications, *IEICE Trans. Inf. & Syst.*, Vol. E78–D, No. 11, pp. 1362–1368 (1995).

[11] Chen, P.: The Entity-Relationship Model – Toward a Unified View of Data, *ACM Trans.Database Syst.*, Vol. 1, No. 1, pp. 9–36 (1976).

[12] Hull, R. and King, R.: Semantic Database Modeling: Survey, Applications, and Research Issues, *ACM Computing Surveys*, Vol. 19, No. 3, pp. 201–260 (1987).

[13] Atkinson, M. et al.: The Object-Oriented Database System Manifesto, *Proc. of DOOD'89*, pp. 40–57 (1989).

[14] Allen, J.: Maintaining Knowledge about Temporal Intervals, *Comm. of ACM*, Vol. 26, No. 11, pp. 832–843 (1983).

[15] 増永良文: マルチメディアデータベースと時間, 情報処理, Vol. 36, No. 5, pp. 369–377 (1995).