

オブジェクトの動きを用いた動画像検索

吉光正典† 吉高淳夫† 平川正人† 市川忠男†

†広島大学大学院工学研究科

†広島大学工学部

動画像データベースにおけるシーン検索手法を提案する。動画では静止画における、縦、横の平面に時間軸を加えた空間を持つため、任意のシーンに対する検索を可能とするためには検索キーに時間的な要素を持たせる必要がある。本研究では動画像内に存在するオブジェクトの運動がこの時間的な要素を持つことに注目し、オブジェクトの動きを検索キーとした検索手法を提案する。またこの手法は、その動きを例示を用いてシステムに入力することで、ユーザの意図を柔軟に反映することができる。

Scene Retrieval by Object's Motion on Video Databases

M.Yoshimitsu†, A.Yoshitaka†, M.Hirakawa†and T.Ichikawa†

†Graduate School of Hiroshima University

†Faculty of Engineering, Hiroshima University

Kagamiyama 1-4-1, Higashi-Hiroshima 724, Japan

This paper proposes a method for retrieving scenes in terms of object's motion. Since video data has the three-dimensional space which is composed of two-dimensional plane and an additional measure for time, it is needed to select a retrieval key having a temporal element so as to make it possible to retrieve any scene in video data. In our research, based on the fact, object's motion existing in the video data has the temporal element, we propose a retrieval method which uses object's motion as a retrieval key. Here, the object's motion is given to the system by means of illustration, so this method is enabling a flexible representation of user's intention.

1 はじめに

近年、コンピューターと大容量の記憶装置の高速化に伴うデータの蓄積、処理能力の飛躍的な進歩によって、画像、音声、動画といったデータを扱うデータベースに関する研究が数多く行われている。

このような研究の中心になっているのは静止画像データベースであるが、最近になってパソコン上でも手軽に動画を扱うことができる環境が整いつつあり、動画データベースの研究も行われるようになってきた。これらの研究の中にはその検索手法についての報告もされている。

一般に、動画は静止画の集合によって表現されるため、従来の静止画像で用いられてきた検索手法をそのまま用いることは可能である。そのため従来の動画検索手法には、あらかじめ決められた検索区間をその内容を表すキーワードを用いて検索を行うものが多い [1]。

また、動画には静止画における縦、横の平面とそれに時間軸を加えた 3 次元的な空間を持っているが、任意の空間を検索するにあたって、動画像内に現れる同じ次元を持つ情報をキーとして扱うことが妥当であることから、動画像内の状態の変化をキーとした検索も行われている [2]。

しかし、従来から提案されているこれらの検索手法では、あらかじめ文字や数字等に変換された情報をキーとして検索要求を出さなければならないため、どうしてもユーザの意図が検索に反映されにくいという問題があった。

このような問題に注目した静止画像の検索においては、検索したい画像の例や略画をユーザがシステムに示すことで、システムはそこからある特徴を抽出し、同じ特徴を持つ画像を検索対象として抽出する手法が報告されている [3][4][5]。

現在、パターン認識の分野では動画像内で運動を行っているオブジェクトの抽出や追跡に関する研究が数多く行われている。この処理結果として得られたオブジェクトの運動は時間軸を持ち、動画像検索における検索キーとして、適していると考えられる。しかし、オブジェクトの運動は視覚的に認識できるもので、その情報を文字情報を用いて表現するには限界がある。

そこで本研究では、動画像検索において、動画像

内に存在するオブジェクトの動きに注目して、動きを例示で与えることでユーザの意図を柔軟に反映させ、該当する任意のシーンを検索する手法を提案する。

2 アプローチ

2.1 研究概要

本研究では、動画像データからユーザが望む任意のシーンを得るために、動画像内のオブジェクトの動きをキーとした検索手法を提案する。

また以下の事を研究の特徴として挙げる。

- 動画像内のオブジェクトの動きを、ユーザが直接例示を用いて指定することができるようにする。

これは、あらかじめシステム側で決められたキーワードを選ぶ検索手法と比べて、ユーザの意図をより正確かつ柔軟に検索に反映できることを指している。

ユーザが直接オブジェクトの動きを指定するためには、実際に指定しやすく、かつ動画像上で認識することがやさしい動きをキーとして扱う必要がある。データの特徴をよく表す対象が存在しても、ユーザの認識能力または記述能力を越えるものをキーとして検索に用いることはできない。

本研究では、動画像内に存在する画面上のオブジェクトが行う見かけの動きを扱うことにする。ここで言うオブジェクトが行う見かけの動きとは、画面上で実際に認識される変化量であり、動画像内のオブジェクトの絶対的な運動を指すものではない。

例：ある動画像内で画面自体が右方向に動いていて、その中に存在するオブジェクトが同じ右方向に、画面の移動速度よりも遅く動く時、検索空間上では左方向に移動しているものと見なす。

本研究では、上記の様に画面上で認識されるオブジェクトの動きにかかる制約を用いた検索手法について考えて行く。

2.2 chain 符号

本節では、動き(軌跡)を表すデータ形式として用いている chain_code についての説明を行う。

chain_code は Fig.1 のような 0 から 7 までの方向指数の列によって表し、ある 1 点から次の 1 点がどちらの方向へ動くかを表す。このデータ形式を用いる理由を、以下に挙げる。

- 軌跡自体が方向性を持つデータであり、単純に軌跡の形だけを扱うのではなく、その移動の向きも評価の対象にしており、軌跡の方向性に関するデータを保持する必要がある。
- 抽出データの軌跡を 1 シーン単位で検索処理に用いると、この軌跡の 1 部分がユーザが指定する軌跡と同じになる場合もある。この時、このデータ形式ならば対応する chain_code の連続した集合をとるだけでその部分的な軌跡のデータを得ることができ、検索処理に都合がよい。

3 オブジェクトの動き

本章では、本研究で取り扱うオブジェクトの動きについて述べる。ここでの説明は、対象オブジェクトの抽出と追跡がすでに行われ、対象オブジェクトが存在した各フレーム毎に、オブジェクト領域がその矩形の対角に位置する 2 点の座標によってすでに得られているものとする。

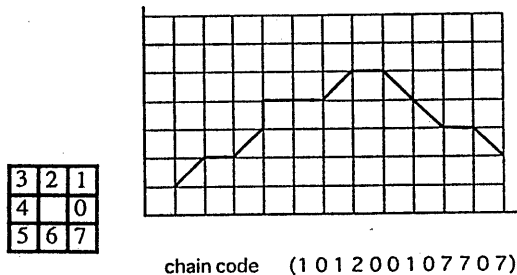


Fig.1 方向指数

3.1 オブジェクトの軌跡

第 1 の制約としてオブジェクトの軌跡を扱う。軌跡はオブジェクトの運動などの結果生じる画面上の位置変化によって得られる。先に挙げた抽出領域の中心点として、フレーム i におけるオブジェクト a の座標を P_{a_i} と表す。動画像内のある連続したフレーム上に存在するオブジェクトを時間軸の正方向に追跡していくと、その軌跡は上記の座標の集合としてデータを得ることができる。しかし、この時点ではオブジェクトの動きは点の集合であり、ユーザが確認したり、記述したりすることができるレベルではない。そこで、この連続した点の間を結び、1 本の線としてオブジェクトの軌跡を表す。

ユーザの検索に用いられるオブジェクトの運動時間の区間(以下、時間幅と呼ぶ)での軌跡を表すデータ形式は、chain 符号化された点の集合として最終的に、以下のように表される。

$$J_a = (c_1, c_2, c_3, \dots); c_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

Fig.2 にその例を挙げておく。

3.2 オブジェクトの速度

連続したフレーム間のオブジェクトの移動距離の大小によって、画面上のオブジェクトの速度を表現する。ここで扱う速度はあくまでも画面上のみかけの運動速度である。このことから、本研究で制約として用いるオブジェクトの速度が計測される時、必ず画面上でそのオブジェクトの位置的な変化を伴う。

ここで扱うのはオブジェクトの速度である。またこの制約を用いる時は必ず先に説明した軌跡の制約と併用する。これによって速度の評価の際に、先に行った軌跡の評価から、速さの向きも同時に評価できる。

ある動画像内にあるオブジェクト a のフレーム s 枚目における速度を、以下のように近似的に表すことにする。

$$v_{a_s} = \sum_{i=s-1}^{s+1} \frac{dis(P_{a_i}, P_{a_{i+1}})}{2l}$$

ここで $dis(P_{a_i}, P_{a_{i+1}})$ はオブジェクト a の P_{a_i} 、 $P_{a_{i+1}}$ 2 点間の距離であり、 $2l$ は速度を計測する区間内の

フレーム数を表す。しかし、この速度のデータをユーザが実際に検索の問い合わせに指定することは実用的でないと考えられる。そこで本研究では、3つの速度パターンで区間を分類する。なお区間を分ける点を速さの分割点とよぶ。

任意の連続した s から i までの区間が、加速、減速、等速の3つの種類の区間に分割される条件を以下に示す ($s \leq i < t$)。

$$\begin{aligned} \text{加速区間} & \quad v_{a_{i+1}} - v_{a_i} > \epsilon_v \\ \text{減速区間} & \quad v_{a_{i+1}} - v_{a_i} < -\epsilon_v \\ \text{等速区間} & \quad -\epsilon_v \leq v_{a_{i+1}} - v_{a_i} \leq \epsilon_v \end{aligned}$$

上記の式中の ϵ_v (正の数) はユーザによって可変とし、これによってユーザの検索要求に応じた区間の分割が可能となる。

この速度を用いて、任意のフレーム間 (フレーム s からフレーム t) の速度の変化率を、加速区間のとき v_{a_i}/v_{a_s} 、減速区間のとき $-v_{a_i}/v_{a_s}$ また定速区間のときは1とする。

ここで、オブジェクト a の軌跡を速度の種類によって分ける i 番目の分割点 p における速度を $v_{a_{p_i}} = (\text{分割点の位置, そこでの速度})$ とし、オブジェクト a の軌跡全体の速度を

$$V_a = (v_{a_{p_0}}, v_{a_{p_1}}, v_{a_{p_2}}, \dots)$$

で表現する。

3.3 オブジェクト領域の拡大縮小運動

動画像内の連続したフレーム間において、オブジェクトの抽出領域の大きさの変化を用いて画面上のオブジェクトの拡大縮小の動きを表現する。

あるフレーム i におけるオブジェクト a の領域の幅を w_{a_i} 、高さを h_{a_i} とする。この時、このデータを直接フレーム単位でユーザが検索の問い合わせとして用いることは、速度の場合と同様に実用的でない。そこで、対応する区間を拡大区間 (zoom-in)、不変区間 (stable)、縮小区間 (zoom-out) の3つにわけて検索に用いる。区間を分ける点を領域変化の分割点と呼ぶ。

あるフレーム s から i までの領域の縦、横の変化をそれぞれ $\frac{h_{a_i}}{h_{a_s}}$ 、 $\frac{w_{a_i}}{w_{a_s}}$ で表す。 s, i 間で領域の変化が全く見られなかった場合ともに1を取る。いま r と

して、この縦、横の変化のうちで1に近い方の値を取ることにする (ただし $r < 1$ のときは $r = -\frac{1}{r}$)。このとき、ある区間 s, i が拡大、縮小、不変になることは次式で与えられる。 ($s \leq i < t$)

$$\begin{aligned} \text{拡大区間} & \quad r > 1 + \epsilon_z \\ \text{縮小区間} & \quad r < -(1 + \epsilon_z) \\ \text{不変区間} & \quad -(1 + \epsilon_z) \leq r \leq 1 + \epsilon_z \end{aligned}$$

ϵ_z は速度の場合と同じで、区間を分割する際の許容量である。また、ある区間の変化率は、拡大、縮小区間の時は r 、不変区間では1とする。

分割点の位置 i とその時の領域の大きさを組にして、それぞれの分割点上のオブジェクトの大きさ $z_{a_{p_i}}$ を保持する。これを用いてオブジェクト a の時間幅全体の拡大縮小運動は以下のように表すことにする。

$$Z_a = (z_{a_{p_0}}, z_{a_{p_1}}, z_{a_{p_2}}, \dots)$$

4 評価処理

本研究では、ユーザの意図を検索によりよく反映させるために、ユーザが検索したいオブジェクトの動きを例示でシステムに与えるようにしている。このような問い合わせは、従来のテキスト型のデータベースで行われていた問い合わせと異なり、問い合わせ条件に完全に一致する答が出る可能性はきわめて低い。従って、何らかの基準に照して類似度を測る必要がある。

本章では以下に軌跡と速さの評価を行う手順を説明する。

4.1 chain 符号による軌跡のマッチング

ここでは、動画像データから抽出された軌跡とユーザが提示した軌跡との類似度の評価を行う動的計画法の最経路問題を適用したマッチング手法について説明する。以下、まず最初にマッチングを評価する評価値 N_i を求めた後、これを最小にするような写像区間を求める手法を説明する。そのあとでもう1つの評価の基準となる写像の均等性を示すパラメーター N_u の求め方とその意味について説明する。

まず動画データから抽出された軌跡のデータ D ($size_D$ 個の chain_code から成る) とユーザから与えられた軌跡のデータ M ($size_M$ 個の chain_code から成る) が、それぞれ ($0 \leq i \leq size_M$), ($0 \leq j \leq size_D$) で

$$D = (d_1, \dots, d_{size_D}); d_j \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

$$M = (m_1, \dots, m_{size_M}); m_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

となるような chain 符号列で表されているとし、 M を D の連続した部分区間へ写像することを考える。ただし仮定として、 M の大きさ $size_M$ は M が写像される D の部分区間よりも小さいこととし、また写像の前後でデータの順序が保たれていて、 M をフレーム s から t までのすべての D の符号に写像するものとする。

いま、 M が D の chain_code d_s から d_t までの部分区間に写像されるとし、この写像を行う関数を f_{md} とすると

$$d_j = f_{md}(m_i) \quad (j \in \{s, s+1, \dots, t-1, t\}) \\ i \in \{1, 2, \dots, size_M\}$$

またこのとき、先程説明したように、マッチングに関して次の制約が満たされるものとする。

$$size_M \leq t - s + 1$$

次に $dis(a, b)$ を Fig.1 中の 2 つの方向指数 a, b ($a, b \in \{0, \dots, 7\}$) がどれくらい離れているかを表す関数とし、以下のように定義する。

$$dis(a, b) = \min(|a - b|, ||a - b| - 4|)$$

このとき M が D に写像される時にかかるマッチングコスト n_i は、写像された d_s から d_t までのすべての点と、写像前の対応する M の要素との差の総和によって次のような式で表すことができる。

$$n_i = \min \sum_{j=s}^t dis(d_j, f_{md}^{-1}(d_j)); (f_{md}^{-1}(d_j) \in M) \\ (j \in \{s, s+1, \dots, t-1, t\})$$

ただし、ユーザが要求を完全な形でシステムに与えることは不可能であり、写像区間が大きくなればそれに応じてコストも大きくなるのが予想される。

そこで実際に評価に用いるコストは、 n_i をその写像が行われる区間の長さで割り、

$$N_i = n_i / (t - s)$$

とする。

このコスト N_i を最小とするような s, t, f_{md} を求めることで、最適なマッチングを行うことができる。関数 f_{md} を求めるための方法を Fig.3 を用いて説明する。Fig.3 には、写像元のデータ M の対応する番号を縦軸 I に、写像先のデータ D に対応する番号を横軸 J に取った図が描かれている。その写像を示す経路は s から t まで写像されたとき、対応する 1 から斜線で記された連続した経路を通して $size_M$ まで到達する。

ここでマッチングを行う軌跡は、時間と逆に進むことはない。また、前に述べた仮定から m は拡大方向にしか写像されない。

このことから、写像後の chain_code D から写像前の chain_code M の対応する、符号の番号を求める関数 g_{dm} は以下のようになる。

$$i = g_{dm}(j); \quad (j \in \{s, s+1, \dots, t-1, t\}) \\ i \in \{1, 2, \dots, size_M\}$$

上記の関数を用いると、Fig.3 のように J 軸上の任意の $j-1, j$ において $j-1$ から j へ至る経路は下記のように 2 つ存在する。

$$g_{dm}(j) = g_{dm}(j-1) + \Delta; \quad (\Delta \in \{0, 1\})$$

実際にこの 2 つの経路のうち、コストの低い経路を選びながら任意の区間を探索して、最もコストの合計が小さい経路を求める。

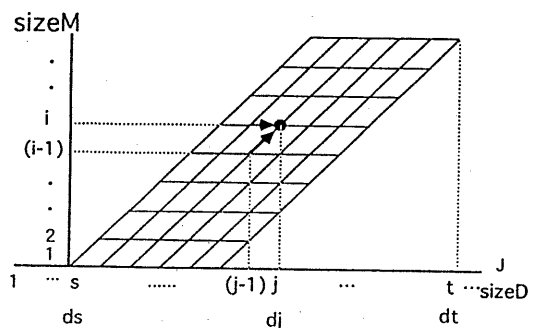


Fig.3 軌跡の写像

これまで、コストを最小にしてその経路を導き出す手法を述べてきた。この手法は、柔軟な写像が行えるようにモデルデータの chain_code の並びのみを考慮した手法である。そのため、モデルの各部分の写像において、それぞれの拡大率が一定である保証が無い。その結果、コストだけを評価した場合、各部分が均等な倍率で写像が行われていないために、実際のモデルと全く似ていない軌跡が評価される場合がある。

そこでこの写像が均等な倍率で行われているかどうかを判断するパラメータを加えて、前述のコストによる評価値と併用してマッチングを評価する。

具体的には、 M が d_s から d_t までに写像された場合、次のように n_u を決める。

$$n_u = \sum_{j=s}^t \left| \frac{\text{size}_M}{(t-s)} * (j-s) - g_{dm}(j) \right|$$

この値も、写像区間が広がるにつれて大きくなることが予想されるため、同じように写像区間で割り、以下のようにする。

$$N_u = n_u / (s-t)$$

これを各部分の写像の均等性を示すパラメータとして評価に用いる。

4.2 速度のマッチング

ここでは、ユーザが入力した速度に関するデータと動画像内から実際に抽出されたデータを、3.2で述べた速度のデータ形式を用いて評価を行う手法を示す。

3.2で述べたように、速度の制約を軌跡の制約と併用する。その時の動画像データから抽出された軌跡のデータと、ユーザから与えられた軌跡のデータを、5.1で用いた D 、 M とする。またそれぞれの軌跡に対応する速度は3.2で述べた速度のデータ形式を用いて、それぞれ以下のように表す。

$$\begin{aligned} V_D &= (V_{D_0}, V_{D_1}, \dots, V_{D_K}) \\ V_{D_k} &= (f_d(k), v_{d_k}); k \in \{0, 1, \dots, K\} \\ V_M &= (V_{M_0}, V_{M_1}, \dots, V_{M_L}) \\ V_{M_l} &= (f_m(l), v_{m_l}); l \in \{0, 1, \dots, L\} \end{aligned}$$

$f_d(k)$ 、 $f_m(l)$ はそれぞれ k 、 l 番目の速度の分割点が軌跡の chain_code D と chain_code M 上の何番目の code に対応しているかを示す関数である。

V_D 、 V_M それぞれの連続した2つの分割点間の区間は3.2で述べた基準に従って、加速、減速、等速の3種類の区間に分けることができる。ここで V_M の分割は ε_v を用いて分割してよいが、 V_D の分割は、この ε_v を写像された倍率に応じて変換し、その上で分割を行う。

S_{D_k} を分割点 V_{D_k} 、 $V_{D_{k+1}}$ 間の区間、また S_{M_l} を分割点 S_{M_l} 、 $S_{M_{l+1}}$ 間の区間を表すとすると分割区間の集合は以下のようになる。

$$S_D = (S_{D_0}, S_{D_1}, \dots, S_{D_{K-1}})$$

$$S_M = (S_{M_0}, S_{M_1}, \dots, S_{M_{L-1}})$$

上記のすべての区間は加速、減速、等速の3種類いずれかの性質を持つ。

そこで第1の評価として、モデルの軌跡を抽出データの軌跡に写像した際、 S_M とそれが写像された S_D 上の区間との類似度を計算する。

モデルデータの i 番目の速度の分割点は、5.1の関数 g を用いて軌跡のマッチングの際に抽出データにおける軌跡の $g_{dm}^{-1}(f_m(i))$ 番目の符号に写像される。 $h(i) = g_{dm}^{-1}(f_m(i))$ とすると、類似度 H を以下の式で表す。

$$H = \frac{1}{h(L) - h(0)} \sum_{i=0}^{L-1} (h(i+1) - h(i)) * R_s(S_{M_i})$$

関数 R_s は、 S_{M_i} とそれが写像される $h(i)$ から $h(i+1)$ 番目の軌跡に相当する区間を S_d から見つけ出し、その類似度を計算する関数であるとする。

この関数によって得られる類似度は写像元のモデルの分割区間の種類に応じて以下の様な基準で行われる。

加速区間：写像された区間の加速区間、等速区間の割合 (0 から 1) を評価する。

減速区間：写像された区間の減速区間、等速区間の割合 (0 から 1) を評価する。

等速区間：写像された区間の等速区間の割合 (0 から 1) を評価する。

例えば S_{M_i} が加速区間であり、写像先の区間が加速、減速、等速区間をそれぞれ $1/3$ ずつ含んでいた時の類似度は $2/3$ となる。

類似度 H は、この得られた割合に写像された区間の大きさ (写像された chain_code の長さ) の重みを掛け、モデルのすべての分割区間において得られたこの値を足して、全写像区間の大きさ (写像された全区間の chain_code の長さ) で割った値である。

次に第 2 の評価として、モデルの各分割区間における変化率とそれらが写像された各区間における変化率を比較して得られる類似度を用いる。この類似度 T は以下の様な式で求められる。

$$T = \frac{1}{L-1} \sum_{i=0}^{L-1} R_t(S_{M_i})$$

ここで関数 R_t は分割区間 S_{M_i} の変化率とそれが写像される $h(i)$ から $h(i+1)$ の間の変化率を比較し、その類似度を計算する関数であるとする。 r_1 、 r_2 を、それぞれ 2.2 で述べたような方法で求められたある区間の写像前、写像後の変化率とすると、この関数 R_t によって得られる類似度は写像されるモデルの分割区間の種類によらず $|r_1 - r_2|$ であり、 T はすべての区間の類似度を求め、写像するすべての区間の数で割った 1 区間あたりの平均の変化率の差とする。

5 条件の例示による指定

本研究では、検索にユーザの意図をよりよく反映させるため、条件の直接の例示入力を許している。本章では、その具体的な入力方法について述べる。

軌跡：ユーザは、検索したいオブジェクトの軌跡をマウスによって記述する。システムは、単位時間ごとにマウスの座標をサンプリングして、そのデータを 3.1 のオブジェクトの軌跡を表すデータに変換する。画面上にはその軌跡がユーザ確認用に表示される。

速度：軌跡を描く際の、マウスのカーソルの移動速度によって、検索したいオブジェクトの速度をシステムに示す。システムはそのサンプリングされた座標の間隔から 3.2 で示した 3

種類の区間に、各点の速度を用いて分解する。画面上には分解された結果を色分けして表示し、ユーザの確認を求める。そのあとオブジェクトの速さを表すデータに変換する。

拡大縮小：軌跡と併用する場合は軌跡の時間幅を、単体で用いる場合は新たに時間幅を設定し、まずこの時間軸上で分割点を設定する。そのあと各分割点ごとに、オブジェクトの変化前と変化後の領域の大きさをマウスをもちいて描いた矩形の大きさを示す。これからシステムは 3.3 で述べた変化率を抽出し、区間の分割を行う。画面上には時間軸と、もし軌跡の制約と併用する場合には軌跡上の分割点の位置も表示する。

6 実験

本研究では軌跡と速度の制約を評価するシステムを実際に構築しており、提案する手法の有効性について評価をおこなっている。本章では紙面の関係から軌跡を用いた検索にかぎって、その実験結果を Fig.4 に示すこととした。

実験で用いた動画はフレームレート 15 (フレーム/秒) で、ユーザが記述するモデルは 20 (フレーム/秒) でデータのサンプリングを行った。Fig.4 にマッチング結果を示す。同図 (a) は動画内から抽出したオブジェクトの軌跡を表している。また (b)、(c)、(d) はそれぞれユーザが記述したモデルであり、それらの (a) へのマッチング処理が行われた結果を (e)、(f)、(g) に示す。 N_u の値がある規定値を越えないものの中でコストが最小になる写像区間を、図中に太線で表している。なお (e)、(f)、(g) については規定値を 1.5 にしているが、結果 (h) は (d) について規定値 1.0 の下で評価しなおした結果である。この場合、(h) は評価値こそ少し増えてはいるが、結果 (f) よりもモデルの特徴をよく表した結果になっている。

実験システムでは、規定値 N_u とコストの最大値をユーザに決めさせ、その規定値以下のマッチングを行い、なおかつそのコストが最大値を越えないデータを検索結果としてユーザに返すようにしている。

7 まとめ

本研究では動画像内に存在するオブジェクトの動きの制約を用いた検索において、その動きをユーザの例示によって記述する手法について述べた。これにより、従来の動きを用いた検索に比べてユーザの意図をより詳しく検索に反映することが可能となる。また、実際に運動の軌跡と速度のマッチングに関する検索システムを構築し、その評価を行った。

今後の課題としては、その他の制約部のシステムの構築とその評価を行うことが挙げられる。なお、このシステムは現在 MS-WINDOWS 上で C 言語を用いて構築中である。

参考文献

- [1] Warren Sack and Marc Davis "IDIC: Assembling Video Sequences from Story Plans and Content Annotations", Proc of the International Conference on Multimedia Computing and Systems", pp.30-36, 1994
- [2] 安部伸治 他, "状態の時間変化をキーとする動画像検索手法", 信学論, Vol.J75-D-II, No.3, pp.512-519, 1992
- [3] Elisabetta Binaghi, Isabella Gagliardi, and Raimondo Schettini, "Indexing and Fuzzy Logic-Based Retrieval of Color Images", Proc of the IFIP 2nd Working Conference on Visual Database Systems, pp84-97, 1991
- [4] T. Gevers and A.W.M. Smeulders, "Indexing of Images by Pictorial Information", Proc of the IFIP 2nd Working Conference on Visual Database Systems, pp98-105, 1991
- [5] 加藤俊一 他, "マルチメディア商標・意匠データベース TRADEMARK", 信学技報, PRU88-9, pp.31-38, 1988









			
(a)	(b)	(c)	(d)
			
0.141593 N_u = 0.380531	1.050000 N_u = 0.387500	0.156716 N_u = 1.485075	0.163793 N_u = 0.931034
(e)	(f)	(g)	(h)

Fig.4 実験結果