

## グラフ表現による動画像内容記述からの 階層的場面構造の構築

前原 恵太 福田 慶郎 上原 邦昭

神戸大学 工学部 情報知能工学科

多様な情報を持った動画像データを取り扱う際には、グラフ表現のように表現力が高い記述形式を用いれば、柔軟な検索を行なうことが可能となる。しかし、複雑な構造を持ったグラフ表現に対する検索のコストは膨大なものになるという問題点がある。本研究で取り上げるシステム HIEROGLYPH は、R. Schank の提案した CD (概念依存) 表現を用いて動画像データベースの内容をグラフ集合として記述し、NA (Network Algorithm) と呼ばれる手法に基づく組織化を行なって、動画像の持つ階層的な場面構造を抽出することを可能にしたシステムである。ここでは、その組織化アルゴリズムについて述べ、実際の動画像データの内容記述に適用した結果抽出された場面構造をもとに、内容記述の表現形式に関する問題点について考察する。

## Hierarchical Organization of Motion Pictures with Graph-Structured Content Description

Keita Maehara Yoshio Fukuda Kuniaki Uehara

Department of Computer and Systems Engineering,  
Faculty of Engineering, Kobe University

Graph matching is a very powerful technique, but the cost of retrieval is intractable. One solution to reduce the matching cost is to organize a compact representation of the database. In this paper, we will propose the system HIEROGLYPH which can organize graph-structured description of motion pictures into hierarchical network using NA (Network Algorithm) based on the MDL principle. We will also present experimental results showing the acquisition of hierarchical scene structure from the graph-structured content description of animation data by finding subgraph isomorphism. Finally we will discuss problems about the description model of motion pictures.

# 1 はじめに

現在、計算機で取り扱われているデータは、単なるテキストから静止画像、音声、動画像へと高度化してきている。このようなデータをより簡単に利用するために、マルチメディアデータベース構築に関する研究が盛んに行なわれている。マルチメディアデータを取り扱う際には、どのような記述形式で表現するかを決定しなければならないが、動画像のように時間関係や因果関係を持った複雑なデータを記述するには、表現力の高い記述形式を用いる必要がある。

グラフ表現は、自由な構造を持つことができる柔軟な記述形式の一つであるが、属性値表現などと比較すると、検索にかかるコストが膨大になるという問題がある。この問題を解消するための一つの手法として、グラフ集合の組織化が挙げられる [1] [2]。この手法は、グラフ集合をいくつかのグラフに共通した部分グラフに基づいて組織化して、共通部分に対するマッチングを一度で済ませ、検索コストを軽減しようとするものである。本研究で取り上げるシステム HIEROGLYPH [3] は、R. Schank が提案した CD (概念依存) 表現 [4] を用いて記述された動画像データの内容を、MDL 原理 [5] に基づくネットワーク構造化手法を用いて組織化して検索効率を向上させるとともに、複数の動画像間に共通する部分構造に着目することによって、動画像の持つ階層的な場面構造を抽出することを可能とするシステムである。

## 2 グラフ集合の組織化

### 2.1 組織化の概要

HIEROGLYPH は、与えられたグラフ集合を、節点ノード、グラフノードおよび、それらの間に存在する中間ノードから構成されるネットワークへと組織化するアルゴリズムである。節点ノードは与えられたグラフ集合の各節点に対応したノードであり、グラフノードはグラフ集合の各グラフに対応したノードである。中間ノードはこれらの間を結ぶノードであり、グラフ集合の各グラフの部分グラフに対応している。グラフ集合の組織化アルゴリズムは、次に示す三つのモジュールから構成されている。

- ノード (節点ノード、中間ノード、グラフノード)

ド) を作成するモジュール

- 部分グラフの組合せを探索するモジュール
- ネットワークを完全なものにするモジュール

すべてのグラフは、節点に分解されてネットワークの入力ノードに与えられる。各節点は、同じラベルを持つ節点ノードにインスタンスとして受け渡される。これらのインスタンスが組み合わせられて元のグラフの部分グラフに対応した中間ノードが次々と作成され、最終的には入力されたグラフ自身を表すグラフノードが作成される。この時点で存在している中間ノードは、グラフノードを作成する際にできたものであり、それに対応した部分グラフが元のグラフ集合を効率的に記述するために有効であるかどうか分からない。そこで、より記述効率のよい部分グラフの組合せを発見するために、記述長がより短くなると考えられる中間ノードを付け加え、その中から探索を行なう。最後に、ネットワークに不足しているパスを付け加えるとともに、余分な中間ノードを取り除くことによって完全なネットワーク構造化が行なわれる。

### 2.2 組織化アルゴリズム

#### 2.2.1 ノードの作成

まず、節点ノードの作成について説明する。初期状態では、ネットワークには入力ノードのみが存在している。HIEROGLYPH に与えられたグラフはそれぞれ節点に分解されて、入力ノードに受け渡される (図 1)。

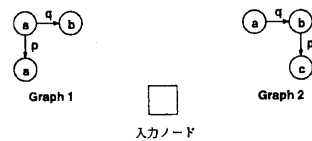


図 1: 入力ノード

入力ノードでは、受け取った節点と同じラベルを持った節点ノードに、その節点をインスタンスとして格納する。もし受け取った節点と同じラベルを持つ節点ノードがなければ、新しい節点ノードを作成し、受け取った節点を新たなインスタンスとして格納する。ここまでの処理が終了した時点で、ネット

ワークには入力ノードと節点ノードのみが存在している (図 2)。

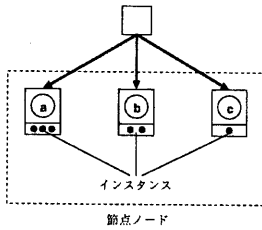


図 2: 節点ノードの作成

次に、節点ノードを二つ適当に選択し、それらのインスタンス間を結ぶ辺が元のグラフに存在しているかどうかを調べる。存在している場合には、それら二つの節点ノードから中間ノードを作成する。作成された中間ノードには、二つの親ノードから受け取ったインスタンスが、新たな一つのインスタンスとして格納される (図 3)。

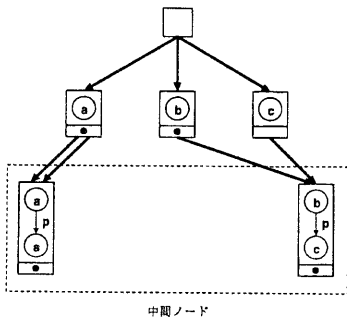


図 3: 中間ノードの作成

次に、インスタンスを持つすべてのノードの中から、先程と同様に、二つのノードを選択し、新たなノードが作成可能であればそれを作成し、インスタンスを次々と受け渡していく。以上の手順を、それ以上新たなノードが作成できなくなるまで繰り返す。この処理が終了した時点でインスタンスを持っているノードは、グラフノードのみに対応していることになる (図 4)。

ここまでの処理により、節点ノード、中間ノード、グラフノードが作成される。しかし、ここで得られた中間ノードに対応する部分グラフは、グラフ集合に含まれるグラフの部分グラフ集合の一部に過ぎな

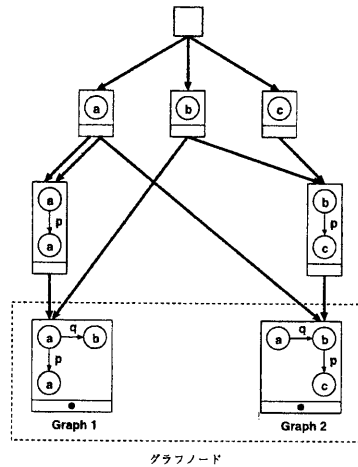


図 4: グラフノード

い。したがって、これらの中間ノードに対応した部分グラフを用いて元のグラフ集合を表現しても、それが効率的に記述されているかどうか分からない。そこで、より効率的な記述が可能となると考えられる中間ノードを新たにネットワークに付け加えて、それらの中から部分グラフの組合せを探索する必要がある。

### 2.2.2 グラフ集合の効率的な記述

グラフ集合を、その部分グラフの組合せを用いて記述する際に問題になるのは、どのような部分グラフの組合せを採用すれば、元のグラフ集合を効率よく表現できるかということである。より多くのグラフに含まれる部分グラフを用いて元のグラフ集合を表現すると、多くのグラフを同一の部分グラフを用いて記述することができるという意味で理想的である。しかし、一般に、多くのグラフに共通に含まれているような部分グラフのサイズは小さいため、この方針が効率的なものであるとはいえない。また、サイズがより大きい部分グラフを用いて元のグラフ集合を表現すると、グラフをより簡単な形で記述することができるという意味で理想的である。しかし、一般に、サイズが大きい部分グラフが共通に含まれているようなグラフの数は少ないため、この方針も効率的なものであるとはいえない。したがって、これらの方針のトレードオフをはかるためには何らか

の基準が必要となる。

本研究では、MDL原理にしたがい、グラフ集合を最も簡単に記述する部分グラフの組合せを探索するというアプローチをとり、この問題を解消している。MDL原理とは、データから未知の情報源の確率モデルを推定するための選択原理の一つとして、情報理論の枠組みで提案されたものである。MDL原理における最良のモデルとは、データを説明するモデル自身の記述長と、そのモデルを用いてデータを記述した際の記述長との和が最小となるようなモデルである。これを、グラフ集合の記述効率の問題に当てはめて考えると、グラフ集合を最も簡単に記述する部分グラフの組合せは、その中の各部分グラフを用いて元のグラフ集合を記述した際の記述長と、各部分グラフ自身の記述長との和が最小になるようなものであるということが出来る。すなわち、グラフ集合  $GSet$  に対する、部分グラフの組合せ  $SGList$  の評価関数  $Eval(SGList)$  は (1) 式で与えられる。

$$Eval(SGList) = DL(GSet|SGList) + DL(SGList) \quad (1)$$

ただし、 $DL(GSet|SGList)$  は、部分グラフの組合せ  $SGList$  を用いてグラフ集合  $GSet$  を表現した際の記述長であり、 $DL(SGList)$  は  $SGList$  自身の記述長である。この評価関数の値が小さいほど、元のグラフ集合は効率よく表現されているということになる。

### 2.2.3 中間ノードの追加

中間ノードの追加については、まず、ネットワークにグラフ集合のすべてのグラフを入力し、マッチするノードにインスタンスを格納していく。そして、すべてのノードに対して、そのノードに対応する部分グラフを用いた場合のグラフ集合全体の記述長を (1) 式にしたがって計算し、この記述長が短いものから二つを選択する。これは、(1) 式の記述長が短いノードほど、より記述長の短い新たなノードを作成する可能性が高くなると考えられるからである。具体的には、(2) 式にしたがって各ノードの選択確率  $P(Node_i)$  を計算して、記述長が短いノードを求めようとしている。

$$P(Node_i) = \frac{\frac{1}{Eval(SubGraph_i)}}{\sum_{j=1}^N \frac{1}{Eval(SubGraph_j)}} \quad (2)$$

ただし、 $SubGraph_i$  は  $Node_i$  に対応する部分グラフ、 $N$  はネットワークに含まれるノード数とする。

もし、選択された二つのノードから新たなノードが作成可能であれば、それを作成して、インスタンスを受け渡す。そして、新たに作成されたノードに対応する部分グラフを用いた場合のグラフ集合全体の記述長を計算し、すべてのノードの選択確率を再計算する。その後、更新された選択確率の下で、再びノードの選択と新たなノードの作成を繰り返し行なう。最終的に、ノードがそれ以上作成できなくなるか、あらかじめ指定されたノード数の上限を越えたら、中間ノードの追加を終了する。

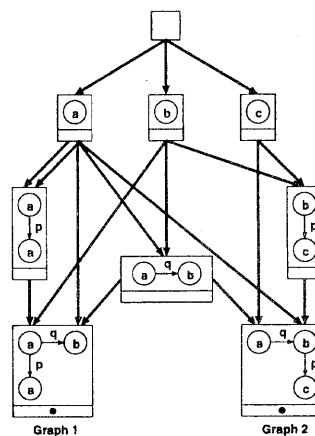


図 5: 中間ノードの追加

### 2.2.4 部分グラフの組合せ探索

次に、ネットワーク中の中間ノードに対応している部分グラフ集合から、元のグラフ集合をより効率よく表現できる部分グラフの組合せを探索する。グラフ集合を効率良く記述する部分グラフの組合せを探索するためには、ネットワーク中のノードのすべての組合せを考慮する必要があるが、その組合せの数は膨大なものとなるため、まず (2) 式の選択確率にしたがってノードを選択し、そこから可能な限り組合せを拡張していくようにしている。

具体的には、まず最初に (2) 式の選択確率にしたがってノードを選択し、そのノードに対応した部分グラフを用いてグラフ集合を表現した場合の記述長を計算する。同様に、その部分グラフを用いてグラフ集合を表現した際の、すべてのノードの選択確率

を再計算する。更新された選択確率の下で、次のノードを選択し、現在選択されているノードと合わせて二つのノードに対応した二つの部分グラフを用いてグラフ集合を表現する。この場合の記述長を (1) 式にしたがって計算した結果が、最初に求めた記述長よりも短ければ、これら二つのノードを求める部分グラフの組合せとする。

この手順を選択可能なノードがなくなるまで繰り返し、できるだけ記述長の短いノードの組合せを見つける。これを一回の探索とし、あらかじめ指定された回数だけ繰り返すことによって、より記述長の短い部分グラフの組合せを見つけ出すようにしている。なお、以下では、この部分グラフに対応する中間ノードを部分グラフノードと呼ぶ。

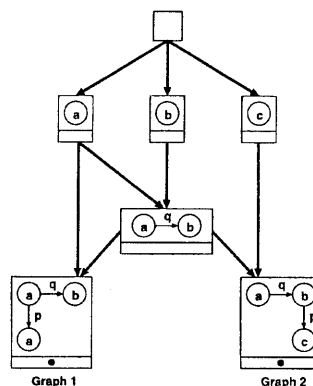


図 6: 得られたネットワーク

### 2.2.5 ネットワークの作成

この例では問題にならないが、この時点で得られているネットワークは一般には不完全である。最初に行なったグラフノードの作成は、グラフ集合に含まれるすべてのグラフをまず格納することを目的として行なったため、あるグラフノードに対して、その部分グラフノードからのパスが存在しないといったことはない。しかし、部分グラフノードの追加は、グラフ集合のより効率的な記述を目的として行なったものであり、その結果生成された部分グラフノードからグラフノードへのパスが存在しているとは限らない。そこで、節点ノード、グラフノード、および部分グラフノードの間を結ぶ最小限の中間ノードを探索し、ネットワークに加える操作が必要となる。また、部分グラフの組合せ探索の結果、不必要となった中間ノードを削除する必要がある。これらの処理を行なった結果として得られるグラフが、最終的に求めるべきグラフとなる。ここまで示してきた例から最終的に得られるネットワークは、図 6 のようになる。

### 2.3 検索アルゴリズム

ネットワークへ組織化されたグラフ集合からの検索アルゴリズムについて説明する。まず、検索したいグラフを入力ノードに渡す。入力されたグラフの各節点は、インスタンスとしてラベルの一致する節点ノードに格納される。インスタンスを持った二つの親ノードから呼び出された子ノードは、その二つ

のインスタンス間の辺を結んで自分自身のインスタンスとなるかどうかを調べ、実際にインスタンスとなっているならば格納する。新たにインスタンスを格納したノードは、さらにその子ノードを呼び出してインスタンスを受け渡していく。最終的にインスタンスがグラフノードに到達すれば、そのノードに対応したグラフを検索した結果として返す。グラフノードに到達していない場合には、インスタンスを格納しているノードに対応した部分グラフを部分的にマッチしている結果として返す。

## 3 階層的場面構造の構築

### 3.1 動画データによる実験

アニメ「まんが日本昔ばなし」の動画データの中から、登場人物が食事をしているような場面を 6 つ収集し、CD 表現による記述を行なった。CD 表現では、概念間の関係として 9 種類の因果タイプが定義されているが、本研究では、*after*, *before* といった概念を表現するために、Allen による時間空間に関する 13 種の間接関係を追加している [6]。HIEROGlyph によってこのデータの組織化を行ない、発見された部分グラフを用いて元のグラフを再現したものを図 7 に示す。この例では、誰かが飲食をし、その後で誰かが話をするという、食事のシーンにおける典型的な場面が抽出されている。また、飲食中はずっと二人の人間がシーン内に存在しているという、より広い意味での場面の記述が獲得されている。

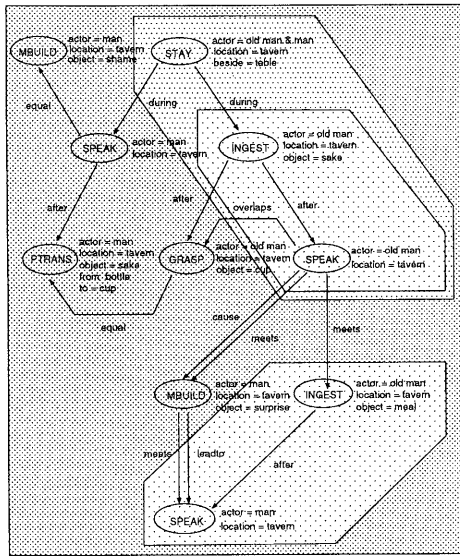


図 7: 組織化の結果

### 3.2 記述形式に関する問題点

類似した場面どうしについて考える場合、その場面内で起こる出来事の時空間関係や因果関係を厳密にとらえない方がよい場合がある。例えば、一般的な食事の場面には、食事時の会話などが含まれているが、会話と食事がどのような順序でなされていてもそれが食事の場面であることに変わりはない。現在用いている拡張された CD 表現では、因果関係や時空間関係を厳密に記述しているため、同じ食事の場面どうしであっても、動作順序の違いによってマッチしないような場合が考えられる。しかし、逆にこれらの概念関係を排除すると、全く逆の意味を持った場面がマッチしてしまう可能性も残されている。現在、このような問題を解消するために、類似した場面間の差異を不完全なグラフマッチングを導入することによって吸収するという手法の導入を検討している。

## 4 おわりに

本研究で取り上げたシステム HIEROGLYPH は、グラフ集合によって表現された動画の内容記述を、その部分グラフの組合せによるネットワークへと組

織化することによって、データの検索効率を向上させるとともに、その部分グラフが動画を構成している階層的な場面構造に対応していることを示した。

グラフ集合をネットワークへと組織化するシステムとしては NA [1] があるが、NA は逐次的なアルゴリズムであるため、常にグラフ集合の持つ共通構造を適切にとらえることができるとは限らない。逆に、HIEROGLYPH は、グラフ集合を一括処理によって組織化しており、新たなグラフが入力されたときに、既に獲得された階層構造を動的に変化させる機構が必要となるが、現時点では実現されておらず、今後の課題となっている。

動画における場面は、複数の動画に共通に含まれており、かつある程度の大きさを持った意味的な構造であるといえる。場面の階層構造を発見する際の基準として MDL 原理を用いている HIEROGLYPH は、このような条件を満たす場面構造を抽出するアルゴリズムとして適しているものと考えられる。

## 謝辞

本研究において、アニメ「まんが日本昔ばなし」を使用するにあたり、学術利用を許可して下さった愛企画センター 川内彩友美代表取締役、および大手前女子短期大学生活文化学科 浦畑育生助教授に感謝致します。

## 参考文献

- [1] Messmer, B. T. and Bunke, H.: A Network Based Approach to Exact and Inexact Graph Matching. Technical Report, IAM-93-021, University of Berne (1993).
- [2] Bunke, H. and Messmer, B. T.: Similarity Measures for Structured Representations, *Lecture Notes in Artificial Intelligence*, Vol. 837, pp. 106-118 (1994).
- [3] 福田慶郎, 上原邦昭: グラフ表現によって記述された事例集合からの MDL 基準による階層構造の発見, 情報処理学会研究会報告, 96-AI-103, Vol. 96, No. 4 (1996).
- [4] Schank, R. C. and Riesbeck, C. K., 石崎俊監訳: 自然言語理解入門 — LISP で書いた 5 つの知的プログラム —, 総研出版, pp. 12-47 (1986).
- [5] 山西健司: MDL 入門: 計算論的学習理論の立場から, 人工知能学会誌, Vol. 7, No. 3, pp. 435-442 (1992).
- [6] Allen, J. F. and Ferguson, G.: Actions and Events in Interval Temporal Logic, Technical Report, 521, University of Rochester (1994).