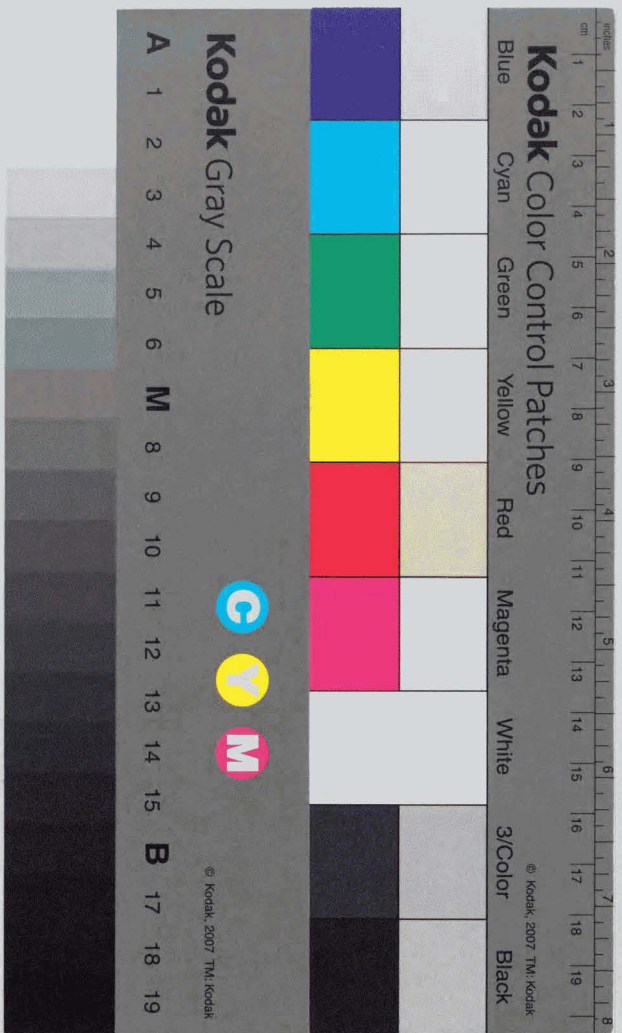


実世界データベースにおける
視覚情報のモデル化と操作

1998年12月

富井 尚志



①

学位論文

実世界データベースにおける
視覚情報のモデル化と操作

指導教官 有澤 博 教授

横浜国立大学大学院
工学研究科
電子情報工学専攻

富井 尚志

1998年12月22日

要旨

本論文では、映像や物体の形状データなどの視覚情報を基にして、実世界をデータベース化する手法について述べる。

データベースシステムの目的は、「実世界の情報を計算機上に表現し、効率良く管理・蓄積・検索する」ことであるといえる。そのためには、データモデルという概念的・理論的枠組を仲介にして、実世界の情報を計算機上のデータにマップする必要がある。ここで実世界の情報として、「対象世界を目で見た」視覚情報をデータベース化する対象にすることは、エンジニアリング分野をはじめとする様々な応用分野において有効である。1980年代以降の計算機技術の発展により、グラフィック画像やビデオ映像などのいわゆる視覚データを計算機上で取り扱うことが容易になってきた。映像についていえば、アナログビデオ信号を計算機に取り込むためのアップコンバータの登場と、2次記憶装置の大容量・高速化、さらに中央処理装置の高速化によるラスタデータの処理性能向上によって、ビデオ信号を完全なデジタルデータとしてディスク上に蓄積し、メモリ上で画像処理を行なうことが現実となった。加えて、映像をデジタルデータとして取り扱う D1 規格や D1 データを計算機に取り込む装置である DVDA (Digital Video Disk Array) の登場により、カメラでとらえた映像をさらに高解像度で情報をロスすることなく取得することが現在では可能となっている。また、コンピュータグラフィックス (Computer Graphics: CG) の世界では、Open GL などの高度な API の発展と、専用のレンダリングハードウェアによって、現実には存在しない空間—すなわち仮想空間をあたかも現実であるがごとく再現することが可能となってきている。そこで本研究では、映像や物体の形状データといったマルチメディアデータを基に、視覚情報データベースの構築を試みた。

視覚情報に基づいて現実世界をデータベース化し、トータルな視覚情報システムを構築するするためには次の2つの面からの考察と、手法の確立が必須である。その一方は、実世界から取得されたデータと、そのデータから抽出された知識をどのような概念モデルによって計算機上にモデル化するかという、データ表現手法である。もう一方は、モデル化された知識を基に、どのようなデータ操作を行うことが可能なかという、データ操作手法である。そこで本論文では、まず本研究の基盤となるデータモデル論について述べた上で、その操作言語について述べる。この二つを柱として、映像を基にしたデータベースを実際に構築し、その検証を行なった。また、映像データベースシステムによって得られた成果を多視点映像へと拡張し、単一視点による映像データベースよりも現実の世界に近い情報蓄積が可能な、現実世界データベースの構築を試みた。

本論文は、6章で構成されている。

1章では、本研究の背景と目的について述べる。

2章では、実世界を計算機上に表現するために、データモデルに関する考察を行なう。データベースの概念的枠組であるデータモデルとしてはAISモデルを用いた。AISモデルは関数型データモデルの一つであり、実世界のあらゆる情報を、主体とそれらの間の2項関係に分解して表現するデータモデルである。AISモデルによって対象世界の情報の概念的な関係だけでなく、映像や、映像を構成するフレーム、フレームに映り込んだ被写体などのマルチメディアデータを基にして対象世界を表現し、蓄積することができた。このデータモデルではフラットな表現方法をとるため、ある特定の視点からの見方だけでなく、2項関係のパスをたどることだけで様々な関連を参照することが可能であることを示した。

3章では、AISモデルによって表現された情報に対するデータ操作言語であるMMQLを提案する。AISモデルは関数をプリミティブとしたデータベースアクセスメソッドを持つ。MMQLは関数型言語であるFPを基にして設計されているため、関数型データモデルであるAISモデルの操作系として良くマッチする。MMQLを用いることで、AISモデルによるデータベースの検索操作とは、データベースの一部を切り出してある特定の視点から構造化して新たなデータを創り出すことであると考えられることができる。これによって、データベースの構成はシンプルに、データ操作はad hocかつ複雑に表現することが可能となる。またMMQLは宣言的な関数型言語に基づくため、並列処理抽出の容易さや、事前に評価時間を概算して検索結果を得られるまでの時間を評価できるなどの特徴を持ち、データベース操作言語として有効である。

4章では、映像データベースの構築と操作について述べる。映像に映り込んだ被写体などの主題を対象とした内容検索(Content Based Retrieval)を実現するために、映像データの取得からモデル表現、映像データベースのデータ操作手法など、システムアーキテクチャの観点も含めて議論する。マルチメディアデータは、データ構造や操作方法がメディアごとに異なるが、MMQLを用いることでメディア依存処理を関数として埋め込んで用いることができるため、例えば検索結果として特定の被写体にスポットライトが当てられるなどの強調加工がなされた映像を得ることができる。これによって、検索操作でデータベースには存在しない新しい情報を得ることが可能となる映像データベースシステムを構築して、その有効性を示した。

5章では、視覚情報を基にした実世界データベースについて述べる。映像を単一視点からの一本のムービーだけではなく、多視点画像 + 物体の3次元形状へと拡張することで、より現実の世界に近い実世界データベースの構築を試みた。ここで実世界をモデル化するために、「意味情報」と個別のデータとの仲介役を担うmediatorを導入することを提案する。本手法で用いたmediatorは、人間の頭部の形状をサンプルとして用い、目や鼻、口といった意味づけがなされている。このmediatorにより、個別の形状データ間に意味の統一を図ることができるため、例えば鼻の体積の平均値を求めるなどの従来の形状データだけでは不可能であった集約演算が可能となる。さらに、このmediatorは、簡略化されたポリゴン形状であるため、時空間質問を高速に評価することが可能である。

最後に6章で結論を述べる。本研究の結論として、

- 関数型モデルに基づくデータモデルによって、様々な見方のできるデータベース設計

を実現した

- FPに基づく操作言語によって、マルチメディアデータベースの操作を実現した
- 内容に基づく検索が可能な映像データベースを実現した
- 映像データベースを拡張し、3次元的な空間をデータベース化するためのモデルを提案した
- 時空間質問処理について考察を行なった
- 物体形状の意味と個別の形状データとを対応づけるmediatorを提案し、時空間質問処理を高速に実現できることを示した

という成果が得られた。

目次

1	序言	1
1.1	本研究の背景と目的	1
1.2	本論文の構成	5
2	関数型データモデル AIS とマルチメディアデータモデリング	7
2.1	マルチメディアとデータモデル論	7
2.2	AIS モデル	10
2.3	AIS ダイアグラム	12
2.4	まとめ	14
3	関数型データ操作言語 MMQL	15
3.1	データ操作言語	15
3.2	オブジェクト式	17
3.3	MMQL の基本構成要素	23
3.4	まとめ	31
4	映像データベースの構成と実現	33
4.1	視覚情報としての映像データと映像データベース	33
4.2	映像モデリング	35
4.3	AIS モデルによる映像データベースの構成	37
4.4	映像データベースのデータ操作	39
4.4.1	様々な検索例	44
4.5	映像データベースの試作と評価	45
4.5.1	データベース構成システムアーキテクチャ	48
4.5.2	映像データベース試作システム	49
4.6	まとめ	50
5	実世界データベースの構成と実現	53
5.1	視覚情報を基にした実世界のデータベース化と時空間データベース	53
5.2	mediator を用いた実世界モデリング	55
5.2.1	シーンの構成要素	55

5.2.2 シーンデータベースの構築	58
5.3 AIS モデルによる実世界データベースの構築	64
5.3.1 シーンデータベースのスキーマ	64
5.4 時空間質問処理	67
5.4.1 空間オペレータによる質問処理	67
5.4.2 質問処理に関する定量的評価	73
5.5 実世界データベースのデータ操作	74
5.6 まとめ	75
6 結言	79
謝辞	82
発表論文	83
参考文献	87

Chapter 1

序言

この章では、本研究の背景となる研究や製品開発の動向について述べ、本研究の目的および、その方針の根拠について説明する。また、本論文の構成について、簡単な説明を行う。

1.1 本研究の背景と目的

「百聞は一見に如かず」という諺がある。旧来より人間は伝聞によって知識を補うよりも、自らの視覚によって得られる情報から、より多くの確実な知識を得てきている。計算機システムが、人間が知識を得、その知識を蓄えるために用いられるのであるとすれば、そこで視覚情報を取り扱えることがどれだけ有効であるかは想像に難くない。

1980年代以降の計算機技術の発展により、グラフィック画像やビデオ映像などのいわゆる視覚データを計算機上で取り扱うことが容易になってきた。映像についていえば、アナログビデオ信号を計算機に取り込むためのアップコンバータの登場と、2次記憶装置の大容量・高速化、さらに中央処理装置の高速化によるラスタデータの処理性能向上によって、ビデオ信号を完全なデジタルデータとしてディスク上に蓄積し、メモリ上で画像処理を行なうことが現実となった。加えて、映像をデジタルデータとして取り扱う D1 規格や D1 データを計算機に取り込む装置である DVDA (Digital Video Disk Array) の登場により、カメラでとらえた映像をさらに高解像度で情報をロスすることなく取得することが現在では可能となっている。また、コンピュータグラフィックス (Computer Graphics: CG) の世界では、Open GL などの高度な API の発展と、専用のレンダリングハードウェアによって、現実には存在しない空間—すなわち仮想空間—をあたかも現実であるがごとく再現することが可能となってきている。

従来計算機で取り扱われてきた文字・数値型のデータとは異なり、視覚データをはじめとして、自然界の音声や MIDI の合成音源など、視覚・聴覚に依存する形態をしたデータは、一般にマルチメディアデータと呼ばれる。マルチメディアデータの特徴は、そのデータの種類ごとに、データの取得方法や表示方法、データ構造が異なるため、専用のハードウェアやデータフォーマットを要することである。たとえばビデオ映像は 1 秒間あたり 30 フレームの画像データ列からなり、D1 などの専用信号規格と DVDA などの録画・再生装

表 1.1: マルチメディアデータベースの変遷

世代	第一世代 (80 年代)	第二世代 (90 年代)	第三世代 (2001 年以降)
対象データ	ビットマップ画像	映像、音声	3D グラフィック 3D(多視点) 映像
総データ量	100MByte 程度	100GByte 程度	100TByte 程度
時間軸依存性	なし	あり	あり
アプリケーション	地図情報システム CAD システム 写真・絵画検索システム	物体認識 ビデオマニュアル シミュレーションライブラリ	動作解析 仮想工場設計 情報人間工学
データモデル	関係モデル、オブジェクト指向モデル → 新たなデータモデルへ		

置を用いることではじめて計算機上で取り扱うことができるデータである。

マルチメディアの応用目的のひとつとして、実世界の情報をできる限りそのままの形で計算機上に表現し、計算機上で仮想的に実世界を再現しようとする試みが見られる。マルチメディアとネットワークを融合することで実現されるテレビ会議システムや、CG によって実世界の情報を仮想的に再現して現実感豊かに再生し体感することのできる仮想現実 (Virtual Reality: VR) システムなどがその例である。これらのシステムは、実時間性の保証やあたかもその場にいるように体験できる—すなわち、いかにリアリティに視覚化するかにかに主眼を置いている。このため、作成された仮想世界を永続的に蓄積し検索して再利用するといった、知識ベースとしての情報システムの実現には至っていない。

別の応用として例えばエンジニアリングの分野においては、熟練作業員の行なった作業を撮影し、映像ライブラリとして保存することで、より視覚的・直観的な作業マニュアルとして用いたり、映像に映った工具の動きを解析し作業の改善を図るなどの、視覚情報を基にした知識ベースシステムへの要求が高まっている。このように、視覚情報とともに現実世界の時間的・空間的情報をどのように計算機上にモデル化し、蓄積・管理・検索するかという視点からの研究が必要となり、視覚情報から「被写体」の「動き」や「意図」などの「知識」を抽出して、元の視覚データとともにこの「知識」を蓄積・管理し、必要に応じて自由に検索できるというマルチメディアデータベースの必要性が生じる。本研究の目的は、現実世界に生じた事象を、視覚情報を基としたマルチメディアデータベースによって一元管理するための、モデル化手法とデータ操作手法を確立することにある。

しかしながら、データの種類の依存した形態やデータ取り扱い方法のあるマルチメディアデータをデータベース化することは、従来のデータベース設計論では論じ得なかった問題に直面することになる。この結果、データベースシステムはデータに依存して構築され、あらゆるアプリケーションから独立してデータを一元管理するいわゆる「真のマルチメディアデータベース」は存在していなかった。例として、表 1.1 に、マルチメディアデータベースの変遷を示す。

表 1.1 に示したように、視覚情報を中心としたマルチメディアデータベースは、80 年代の画像データベース (第一世代) によって始まるといって良い。画像データは、時間的に変化することのない、すなわち時間に依存しないデータであり、その点で映像データと比較

するとデータ量も少ない。ただし、画像処理によって劣化画像を復元したり、画像から被写体を抽出するなどの要素技術が必要となる。応用として、地図情報システム (Geographic Information System: GIS) や写真や絵画のデータベース化、CAD などのベクトルデータによるデータベースがあげられる。従来の数値 (スカラー量) 情報データベースとの違いは、画像のラスターデータやベクトルデータを取り扱うために 2 次元の新たなオペレータを要したことである。このため、データベースの基盤となるデータモデルも、従来の関係モデルにベクトルオペレータを組み込むなどの方法で解決することができた。また、ラスターデータの取り扱いと、画像に映った被写体情報の取り込みのために、オブジェクト指向パラダイムを導入し、オブジェクト指向データベースをエンジンとしてシステムを構築した例も見られた。

しかし、90 年代に入り、映像や音声データを対象とするようになると、時間軸の取り扱いが必要不可欠となった。また、時間に依存したデータであるため、時間に比例してデータ量が増加した。その結果、データを取得する手段 (専用ハードウェア) や蓄積するための大容量 2 次記憶装置、さらに、データを取り扱うための高速プロセッサなど、より多くの計算機資源を要し、第一世代と異なりシステムアーキテクチャにも大きな変更を要求された。さらに、データ操作のためのオペレータも第一世代のように単純なベクトルオペレータだけでは済まず、映像や音声などのメディアに依存したデータ操作—メディア依存処理の実装が必要となった。この結果、メディア依存処理をカプセル化して用いるオブジェクト指向データベースの提案が盛んになされるようになった。しかし、時間軸の取り扱いが不十分だったことと、データベースの観点からは、ある特定の見方の保証しかできなかったこと、さらに、様々な見方を実装しようすると参照関係が複雑になり過ぎて高効率な実装が困難だったことなどから、さらに先の世代では純粋なオブジェクト指向モデルではさらなる工夫を要することになる。第二世代のマルチメディアデータベースは、映像を中心とした、すなわち、ある一つの視点から見た世界をデータベース化することが目的だったといえる。次世代のマルチメディアデータベースは、多視点化、3 次元化が進んでいる。これは、現実の世界をできる限りそのままの形でデータベース化しようという試みである。ここに至って、現在のところ、確立したシステム、確立したモデル化手法、確立したデータ操作手法は存在していない。

例えば議論を少し戻して、視覚化することを重視したシステムについて考えよう。映像蓄積/検索システムの一例として、ビデオオンデマンド (Video on Demand: VOD) があげられる。VOD は、映画やビデオ番組などのビデオコンテンツの配布に用いられるシステムである。このため、有限のネットワーク資源を用いて高効率にコンテンツの再配布を行うなどのデータ蓄積能力、再配布機能に優れる。また、3 次元空間を対象にした視覚情報システムの代表として、VR があげられる。VR は CG によって仮想的に世界を構築し、その仮想空間を現実の空間であるかのように利用者が体感することのできるシステムである。

いずれの視覚情報システムも優れた特徴を持ちながら、なお知識システムとして満足のいく機能を持ち合わせないのには、理由がある。それは、これらのシステムがどれもアプリケーションに依存したデータしか取り扱うことができないという点である。特に、特殊なグラフィックワークステーションやレンダリングエンジンなどの専用ハードウェアを要するマルチメディアコンピューティング環境においてこの問題は顕著に現れてくる。これ

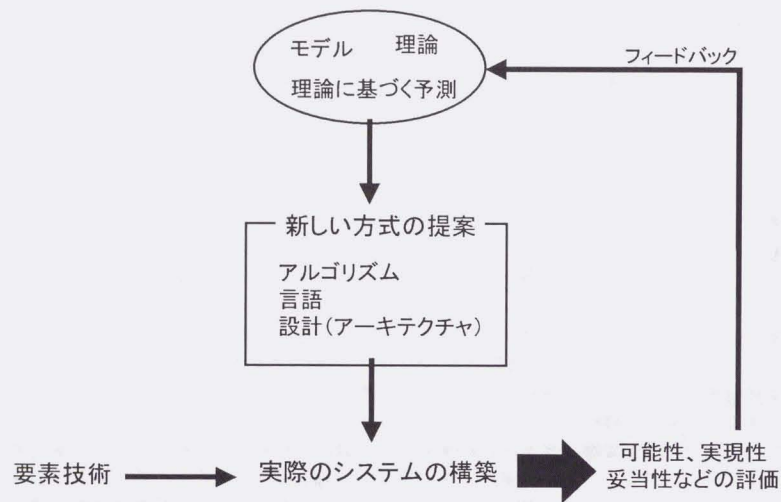


図 1.1: マルチメディアデータベースの研究のながれ

らのシステムおよびハードウェアは、CGによる仮想的な世界をいかに現実感豊かに利用者に見せるかということの実現に力が注がれており、データ取得や蓄積、検索などのトータルな情報システムに要求される機能を必ずしも保証するものではない。ただ単にハードウェア技術が向上してCGのレンダリング性能やテキストチャッピング機能が充実しただけでは、真に有効な視覚情報システムを構成することはできないと考えられるのである。

視覚情報に基づいて現実世界をデータベース化し、トータルな視覚情報システムを構築するするためには、特に本研究では以下の5点について考察されていなければならない。

1. 視覚情報（データ）をどのような手段によって取得するのか
2. 取得されたデータを、どのように解析してそこから知識を抽出するのか
3. 取得されたデータと抽出された知識は、どのような概念モデルによって計算機上にモデル化されるのか
4. モデル化された知識を基に、どのようなデータ操作を行うことが可能なのか
5. データ操作の結果得られるデータをさらにどのように視覚化するのか

このときの本研究における研究の流れを図 1.1に示す。

1.と5.は、システムを構成する要素として、どのようなハードウェアが利用可能なのかについて考察し、実際にシステムを構築してその可能性や実現性、妥当性などを検証することで解決する。また2.は、そのようなシステムを構築するための要素技術を開発することであり実際のシステム構築には必要不可欠である。これらのステップによって、新たに構築したシステムの可能性/実現性/妥当性を得られれば、それは研究成果として普遍的なものとなり、次なる研究の根拠・背景・理論となり得る。その上で、本研究の中心となるのは、これらの背景や理論に基づく予測の基に3. 新たなモデルやアルゴリズムを提案すること、と、4. 新たな言語体系によるデータ操作記述を提案することである。そして、提案したモデル化手法と操作手法を基に実際のシステムを構築し、その検証を行なっていく。

そこで本論文では、まず本研究の基盤となるデータモデル論について述べた上で、その操作言語について述べる。この二つを上記の3. 4. の柱として、映像を基にしたデータベースを実際に構築し、その検証を行なった。また、映像データベースシステムによって得られた成果を多視点映像へと拡張し、より単一視点による映像データベースよりもより現実の世界に近い、現実世界データベースの構築を試みた。

1.2 本論文の構成

本論文では、視覚データ、特に映像を基にして、実世界をデータベース化するために、データモデルと操作系、そしてシステムアーキテクチャの観点から議論し、新たな視覚情報システムを構築することを試みる。

本論文は6章で構成されている。

1章では、本研究の背景と目的について述べる。

2章では、実世界を計算機上に表現するために、データモデルに関する考察を行なう。本研究では、実世界の情報をできるだけ細かく分解し、複雑な構造を持たさずにモデル化できるAISモデルを用いた。AISモデルは関数型データモデルの一つであり、実世界のあらゆる情報を、主体とそれらの間の2項関係に分解して表現するデータモデルである。AISモデルによって対象世界の情報の概念的な関係だけでなく、映像や、映像を構成するフレーム、フレームに映り込んだ被写体などのマルチメディアデータを基にして対象世界を表現し、蓄積することができる。

3章では、AISモデルによって表現された情報に対するデータ操作言語であるMMQLを提案する。MMQLを用いることで、AISモデルによるデータベースの検索操作とは、データベースの一部を切り出してある特定の視点から構造化して新たなデータを創り出すことであると考えられる。これによって、データベースの構成はシンプルに、データ操作はad hocかつ複雑に表現することが可能となる。またMMQLは宣言的な関数型言語に基づくため、並列処理抽出の容易さや、事前に評価時間を概算して検索結果を得られるまでの時間を評価できるなどの特徴を持ち、データベース操作言語として有効である。

4章では、映像データベースの構築と操作について述べる。映像に映り込んだ被写体などの主題を対象として検索したりデータ操作を行なう場合には、VODのように単にフレーム列を蓄積するだけでは不十分である。本研究では、映像データの取得からモデル表現、映

像データベースのデータ操作手法など、システムアーキテクチャの観点も含めて議論する。マルチメディアデータは、データ構造や操作方法がメディアごとに異なるが、MMQLを用いることでメディア依存処理を関数として埋め込んで用いることができるため、例えば検索結果として特定の被写体にスポットライトが当てられるなどの強調加工がなされた映像を得ることができる。これによって、検索操作でデータベースには存在しない新しい情報を得ることが可能となる映像データベースシステムを構築して、その有効性を示した。

5章では、視覚情報を基にした実世界データベースについて述べる。映像を単一視点からの一本のムービーだけではなく、多視点画像 + 物体の3次元形状へと拡張することで、より現実の世界に近い実世界データベースの構築を試みた。この実世界データベースでは、物体の形状データもまた同時に蓄積されるため、任意の視点からの再構成が可能となる。ここで実世界をモデル化するために、「意味情報」と映像やポリゴンデータといった低レベルのデータとの関連づけを行なうために、簡略化されたCGによる mediator を用いる方法を提案する。本手法で用いた mediator は、人間の頭部の形状をサンプルとして用い、目や鼻、口といった意味づけがなされている。この mediator により、個別の形状データ間に意味の統一を図ることができるため、例えば鼻の体積の平均値を求めるなどの従来の形状データだけでは不可能であった集約演算が可能となる。さらに、この mediator は、簡略化されたポリゴン形状であるため、時空間質問を高速に評価することが可能である。

最後に6章で結論を述べる。

Chapter 2

関数型データモデル AIS とマルチメディアデータモデリング

既存のデータモデルでは、マルチメディアデータを取り扱うためにはいくつかの問題点があり、モデルの改良や拡張では容易に解消できない。我々は、後述するいくつかの特徴から、関数型データモデルがマルチメディアデータベースの基礎として有望であると考えた。この章では、関数型データモデルの背景について説明し、全く新しい関数型データモデルである AIS モデルを提案する。AIS モデルは、オリジナルの関数型データモデルが持たなかったいくつかの特徴を備え、マルチメディアデータベースのデータモデルとしての適性を備えているものと考えられる。

また、例として、実際に AIS モデルを用いて現実世界の情報のモデリングを行ない、AIS モデルの有効性を示す。

2.1 マルチメディアとデータモデル論

1960年代に、計算機を用いてデータを蓄積し管理するシステムが発表されてより、データベースシステムは、その基となるデータモデルとパラダイムとともに発展してきた。それ以来、データベースシステムの目的を抽象的に一言でいえば、「現実世界の情報を計算機上に表現して一元管理する」ことであるといえる。そのためには、技術的問題としては、大量データを蓄積できるハードウェアの導入と、高効率なアクセス法の開発が不可欠であった。しかし、単に高速なハードウェアが存在すれば、高効率に情報を管理できるツールが作成できるわけではない。データベースシステムの概念的・論理的側面の設計として、アプリケーションからデータを独立させ、データを統合的に管理できることと、一貫性制約保持が求められてきている。

一般にデータベースを構築し、利用する際概念を図 2.1 に示す。データベースで取り扱う対象となるのは、抽象的にいえば、「現実世界の情報」である。例えば、映像データベースを構築したい場合には、単なるビデオフレームデータだけではない。その映像に何が映っているのか、その映像が表現しようとする主題は何か、その映像が映し込んだ世界は

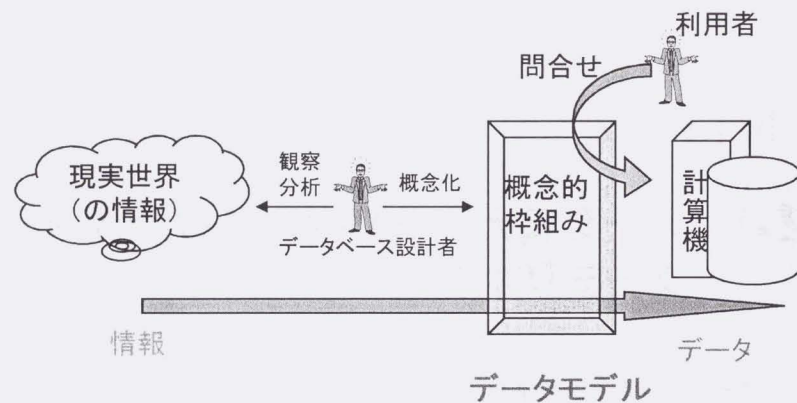


図 2.1: 実世界の情報の計算機上へのマップ

どのような要素によって構成されているかなどの情報をすべて対象としなければならない。これらの情報は、後にデータベース利用者が検索の対象とする情報であり、データベースを構築する際にデータベース設計者はあらかじめすべてのデータを抽出しておく必要がある。また、効率の良い検索を実現するために、現実世界の情報を観察・分析し、利用者が有効に利用できる形のデータとして計算機に保持しなければならない。

このためには現実世界の情報を、計算機上のデータとして効率良く表現するための枠組と、そうして保持されたデータを利用者が効率良く取り出すデータ操作方法が必須である。すなわち、「現実世界の情報を計算機上に表現して一元管理する」ためには、図 2.1 に示したように、有限の計算機資源の基に、現実世界の情報をどのような(理論的)枠組によってとらえて、その枠組をどのように計算機上に実装するのか、という、実世界情報と計算機システムとの間を担うモデルの導入が必要となる。

マルチメディア技術や 3D グラフィック技術、ヴァーチャルリアリティ技術など、計算機技術にさまざまな発展が見られる今日においても、データベースシステムに対する抽象的な目的は変わっていない。しかし、データの取り扱いが従来の文字・数値データとは根本的にことなるマルチメディアをデータベース化するためには、ただ単純に旧型のデータモデルとそれに基づくデータベースシステムでは、予想されなかった困難に直面することとなった。

マルチメディアデータベースは、前章の表 1.1 に示した通り、変遷を重ねてきた。80 年代の画像データベースでは、データベース作成者がデータを観察して決めた感性指標や被写体の名前などを手動で与えインデックスとして利用することが多かった。例えば Egenhofer[EGEN94] は、SQL の枠組の中で地図情報データベースを実現している。これ

は、従来型の関係演算の発展型であるといえる。

しかし、90 年代前半の映像データベースでは、動画像に映り込んだ被写体の動きや、その時区間情報などを、映像に対するインデックスとして利用することで、映像の内容に基づく検索(Content Based Retrieval: CBR)の実現を試みている。このため、データベースに蓄積されるデータは、元映像のフレーム列だけではなく、その中の特定の時区間や、その時区間の示す被写体形状など、様々な形態のデータが入り子状になって現れることになる。Oomoto ら [OOMO93] は、フレーム列や、特定の時区間をオブジェクトとして定義し、オブジェクト間の情報継承や相互参照によってデータベースを構築する方法で CBR を一部実現している。このように、オブジェクト指向プログラミングの概念をデータモデルとして拡張し用いたモデルは、オブジェクト指向データモデルと呼ばれる。また、Masunaga [MASU95] はオブジェクト指向データモデルの概念の基にして、データベーススキーマを設計した上で、実装したシステム上では映像と音声の同期問題を解決したシステムを開発した。このように、マルチメディアをデータベース化するためには、新たなモデルに基づいて実際にシステムを構築し、システムとしての総合的な評価を行なうことでモデルの妥当性の検証を行なうことが一般的である。

データベース化する対象を、単なる映像から多視点映像や 3 次元空間に拡張した第三世代のマルチメディアデータベースでは、より現実世界に近い物体の 3 次元的な形状や、動きをインデックス化する必要が生じる。また、多視点映像の同期や、空間中での動きの表現方法、ふるまいをイベントとしてとらえた時間のモデルなど、さらに複雑な形態のインデックスが要求される。したがってこれらの情報すべてをモデル化しデータベース化するためには、関係モデルと関係演算などによる伝統的手法では不可能出ある。たとえば関係モデルは、順序の概念を持たないため、時間を表現するために工夫を要する。また、メディア依存処理の組み込みや、順序に基づいた操作を規定する必要があるなど、モデルの根本となる基本概念からすべて考え直す必要が生じる。

そこで、真のマルチメディアデータベースを実現するためには、実際に理論的モデルを考え、それを実装し評価する必要がある。マルチメディアデータベースを実現するための計算パラダイムとして、大きく分けて 2 つの手法があると考えられる。ひとつは、データを特定の見方からカプセル化して、情報を隠蔽していく方法であるオブジェクト指向に基づく手法である。もうひとつは、逆に、データを最小単位まで分解し、できる限りフラットに表現して様々な見方を保証するおくエンティティベースの方法である。

オブジェクト指向モデルでは、特定のオブジェクトに依存するオペレータを、オブジェクトのメソッドとして埋め込んで実装することができる。このため、データベース利用者はデータベース操作をより上位概念から利用する際には、メディア依存処理などのデータ操作を意識せずに検索する見方が可能である。しかし、データのある特定の見方についてカプセル化した結果、別の見方からの参照に工夫を要し、直感的にデータを取り扱うことが難しい。この点は、あらゆるアプリケーションに対し、対等に情報サービスを行う必要のあるデータベースシステムの概念からは不適切であることが多い。

一方、情報を最小の単位まで分解し、それらの関連によって情報を記述するエンティティベースのモデルは、現実世界の「意味」を表現する意味モデルに多く見られる。このモデルでは、データベースのある型に対するインスタンスのアクセス法を、関数によって表現

できる。このため、意味の表現に適することと、集合操作による宣言的な操作記述を実現できることから、データベース設計によく用いられてきた。また、この方法で設計されたデータベースは、特定の見方を規定しない、フラットな参照関係を基に構築できるため、様々なアプリケーションから対等な形でデータベースを利用できるという利点を持つ。本研究では、このエンティティベースのモデルである AIS モデルを導入することで、マルチメディアデータベースの構築を試みる。表現手法としては、主体と呼ばれる情報の最小単位と、それらの間に規定される関数関係によってフラットな表現手法を提供する。AIS モデルではデータベースに蓄積された情報がすべて関数によって表現されるため、関数型操作系を基に宣言的なデータベース操作記述が可能である。

また、操作手法としては、関数によって表現されるため、これを利用して関数型操作系を基に宣言的なデータベース操作記述が可能である。また、操作のプリミティブ（基本要素）が関数であるため、メディア依存処理も関数として実装すれば、さまざまな操作操作系の中に新たに埋め込むことができる。

2.2 AIS モデル

前節で述べたように、マルチメディアデータベースを構築するためには、新たなモデルに基づいてシステムを設計し、評価を行なう必要が生じる。繰り返し述べているように、いろいろな見方やつながりを自由に表現できるという観点から、単位となる情報の小さく、かつ平坦な構造を持つモデルが望ましく、本研究ではデータモデルとして AIS (Associative Information Structure) モデルを採用する。

本節では、AIS モデルの基本要素について述べる。

AIS モデルの本質的な構成要素は、主体 (entity) と、2 項関係で表される主体間のつながり (association) である。

主体は、現実世界における事物・事象の代理物である。たとえば、一つの物理カットや、映像の一フレーム、映像の表す主題などはすべて主体である。一つの主体は内部構造を持たない「点」として表され、他の主体とは内部的な識別子 (entityID) によって明確に区別される。また、文字、整数、実数、ビットマップなどの値データもまた主体であり、値主体 (Value Entity) と呼ばれる。すべての主体は、ER モデル [CHEN76] のように主体型 (Entity Type) によってまとめられ、主体型名によって主体の表す意味を区別する。

主体間のつながりには、大きく分けて次の二つがある。一つは主体同士の $m:n$ や $1:n$ などの関連を表すつながりである。このとき、つながりは関数によって表現される。例えば、主体 b_1, b_2 を考え、主体 a は関数 f によって b_1 とつながれるとき、 $a: f = b_1$ という関数インスタンスがあるとみなすことができる。ここで、 $a: f$ は関数 f を a に適用する、すなわち $f(a)$ を意味する。ところで関数の返り値は多値となっても良い。例えば $a: f = \{b_1, b_2\}$ は、 a とつながる主体が b_1 と b_2 であることを示し、返り値は特定の順序を持った集合であると考えられる。また、関数は両方向に参照することができる。このような関数によるつながりもまた、一つの主体とみなし、これを関連主体 (Relationship Entity) と呼ぶ。ER モデルと同様、一般には n 項の関連を考えることができる。この場合、関連主体は、関連を構

成する主体集合の直積の要素を表していることになる。このように、関連をまた主体とみなすことによって、すべてのつながりを $1:n$ か $n:1$ 、もしくは $1:1$ の 2 項関係に分解することができ、関数のインスタンスは 2 項関係によって表されるといい換えることができる。

主体間のつながりのもう一つは、親となる主体が存在してはじめて存在する子主体を考えることができるの親子のつながりである。このような主体を成分主体 (Component Entity) と呼ぶ。例えば図 2.3 において、従業員 (EMP) の名前 (ENAME) や職歴 (JOB.HISTORY) は、親 (ある従業員) が定義されてはじめて導出される主体である。このため、親の主体がデータベースから削除されると、その成分である主体も同時に削除される。成分主体もまた、関数によって参照することができる。ある従業員を表す主体 $e1$ に対し、名前を求める関数 $EmpName$ を適用することにより、 $e1$ の成分である名前が得られる。また同様に、主体 $e1$ に対し、職歴を求める関数 $EmpJobHistory$ を適用することにより、 $e1$ の成分である主体列 $\{j_1, j_2\}$ が得られる。このように成分主体もまた、多値関数の定義が許される。多値関数によって結ばれる成分を多値成分 (multi valued component) と呼ぶ。また単に一つの成分を定義する際には、単値成分 (single valued component) と呼ぶ。成分主体もまた、関連主体と同様の 2 項関係によって表される。

値主体は特別な主体であり、通常の主体型とは異なる主体型に分類される。値主体を定義する主体型は、値そのものの (文字列、数値などの) データ型を示す。ピクセルマップなどのデータ型も、値主体型として分類され、ピクセルマップデータはピクセルマップの値主体型のインスタンスとなる。同値である値は、それを参照する、すなわち、それと結ばれる主体によって区別されるため、ある値主体型に同じ値が複数存在することはない。つまり、値を参照する主体が存在し、値主体とそれを参照する主体とは、 $1:n$ の関係で結ばれる。ここでこれらの主体を他から直接参照できると、スキーマが変更・成長した際に、単純に参照関係を増やすだけでは済まず、新たに主体型を設けないとならないという問題が生じる。したがって、値主体型と結ばれる主体型は必ず、ある主体型の単値成分でなければならない。

値主体について結論すれば値主体は、ある主体型の単値成分である主体型と、 $1:n$ の 2 項関係によって結ばれ、双方向からの参照が可能である。

上記に述べたように、データベース中の主体間のつながりは 2 項関係のみによって表すことができるので、任意の主体型から特定の 2 項関係のパスを通してある主体型への関数を定義することが可能となる。この関数をデータベース関数と呼ぶ。このとき、データベース関数はパス名を並べることによって記述される。例えば図 2.3 においてデータベース関数 $DeptEngageEmpNameStr$ は、主体型 DEPT からパス ENGAGE、EMP、ENAME を通って値主体型 STR への関数であり、ある課に所属する従業員の名前を参照することが可能となる。またこのようにデータベース関数は、関数の定義域のある 1 つのインスタンスに対し、単値または多値を返す関数である。

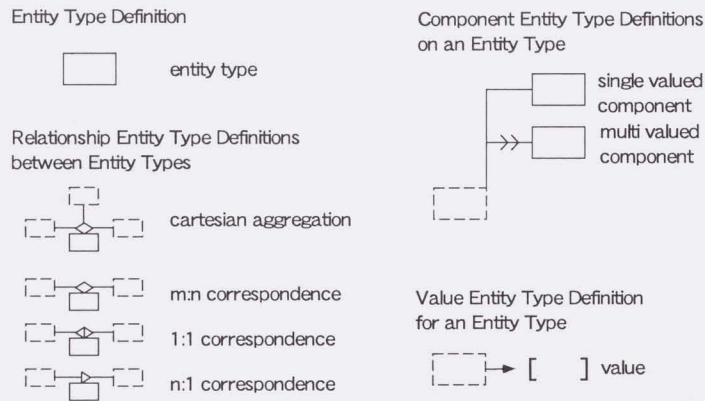


図 2.2: AIS ダイアグラムの基本要素

2.3 AIS ダイアグラム

前節で述べたように、AIS モデルは主体とそれらのあいだのつながりを基本要素とする。このモデルは、Chen による ER モデル [CHEN76] の拡張モデルであるということが出来る。このため、データベースのスキーマ設計はダイアグラムによる図示表現によって行なうことが可能である。AIS モデルは、インスタンスベースのモデルである。このため、インスタンスは特定の型 (主体型) に属するが、ダイアグラムによる手法では主体型とそれらの間の関係の規定を図示することで、集合とその間の関数的制約を表現することが可能である。本節では AIS ダイアグラムの基本要素について述べる。

AIS ダイアグラムの基本要素を、図 2.2 に示す。また、AIS ダイアグラムを用いて作成された映像データベーススキーマの例を図 2.3 に、インスタンスの例を図 2.4 に示す。

このスキーマは、次の情報を表している。「ある会社には、いくつかの課 (DEPT) が存在しその課名 (DNAME) もまたデータベースによって表現される。また従業員 (EMP) には、その名前 (ENAME) や給与額 (SALARY)、職歴 (JOB_HISTORY) などの情報が付与される。一人の従業員の職歴は複数存在して良く、個々の職歴 (JOB_RECORD) にはその職名 (JOB_NAME) や開始年月 (START_YEAR, START_MONTH)、終了年月 (END_YEAR, END_MONTH) が付与される。ある従業員が特定の課に所属 (ENGAGE) した場合、所属日 (EDATE) を記録しておくことにする。所属日は年 (YEAR)、月 (MONTH) および日 (DATE) からなる。ところで、一つの課には当然複数の従業員が所属するが、ある従業員が複数の課に同時に所属することも許される。この時、それぞれの所属について所属日を管理しなければならない。」

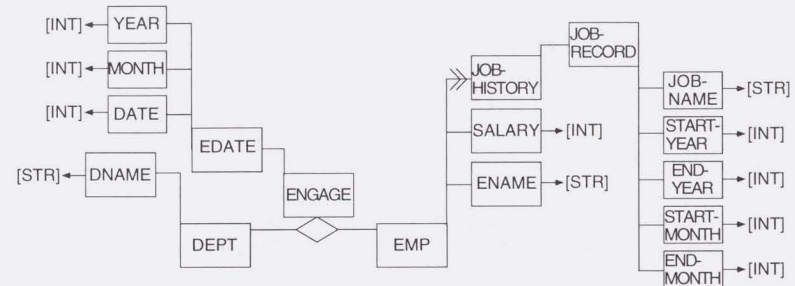


図 2.3: 従業員データベースのスキーマ

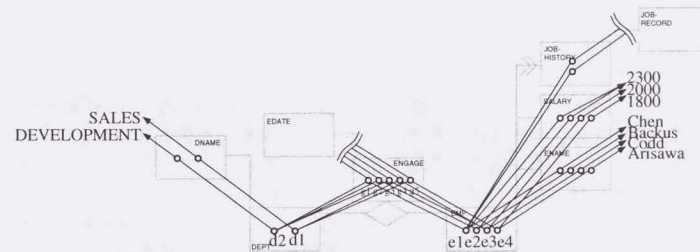


図 2.4: 従業員データベースのインスタンスの例

このスキーマのもっとも主要な構成要素は、従業員 (EMP) を表す主体と、課 (DEPT) を表す主体である。

従業員の名前 (ENAME) や給与額 (SALARY) は、主体型従業員の単値成分として定義され、それぞれ文字列 (string:STR) と整数 (integer:INT) のデータ型の値主体が付与される。また職歴 (JOB_HISTORY) は従業員の多値成分であり、その個々の職歴 (JOB_RECORD) の単値成分として開始・終了年月や職名がそれぞれ数値、文字データとして付与される。同様に課 (DEPT) についても、その課名 (DNAME) が単値成分として付与され、課名のデータ型は文字列である。

一方、一つの課に複数の従業員が所属することと、一人の従業員が複数の課に所属して良いことから、所属 (ENGAGE) は従業員と課の直積によって表現される。この結果、所属の主体型は直積集約を表す関連主体型によって表現される。

従業員データベースのインスタンスの例を図 2.4 に示す。

例えば図 2.4 において、主体型 EMP の主体 e1 は、主体型 ENAME、SALARY に成分主体を持ち、それぞれ“Chen”、2300 と結ばれている。これは、ある従業員 e1 が存在し、その名

前が“Chen”、給与が2300であることを示している。また、多値成分主体型JOB_HISTORYに2つの主体を持つことから、2件の職歴を持つことを表している。

このようにデータベース関数は、定義域の1つのインスタンスに適用すると、そのインスタンスと結ばれている単値または列(多値)を返し、パスを特定した双方向の参照が可能である。したがって、インスタンスを単純な2項関係としてとらえることができ、様々な角度から情報をとらえることができる。

この図に示した通り、すべての主体型はデータベーススキーマ中で一意である主体型名が英字大文字によって付けられ、データベース関数はパスとなる主体型名を、先頭だけ大文字としてつなげ合わせた文字列によって表わすこととする。

図4.3に示す通り、主体は内部構造を持たず、データベースは単純な2項関係のみによって構成される。また、自由に2項関係のパスを設定できるデータベース関数を用いることにより、すべてのつながりを2項関係としてとらえることができる。したがって、AISモデルに基づくデータベースはマルチメディアデータも含めて、情報構造が平坦な形で記述され、さまざまな見方を提供することが可能である。

2.4 まとめ

本章では、マルチメディアデータベースを構築するためのデータモデルについて述べた。データベースシステムの目的を抽象的に一言でいえば、「現実世界の情報を計算機上に表現して一元管理する」ことであるといえる。この目的を満たすマルチメディアデータベースシステムを構築するためには、新たなデータモデルが要求される。

本章では、データモデルとしてAISモデルを採用した。AISモデルは、情報を最小の単位にまで分解した主体と、それらの間に定義される2項関係によって、現実世界の情報を表現するデータモデルである。AISモデルによって、特定の見方からではなくフラットにデータベースを構成できるため、様々なアプリケーションから利用することが可能であることを示した。

Chapter 3

関数型データ操作言語 MMQL

本研究では、前章での考察から、視覚情報のモデル化のために、AISモデルを用いることにした。本章では、AISモデルによって設計されたマルチメディアデータベースから検索結果であるオブジェクト式を生成するための検索言語である、マルチメディア検索言語(MMQL)を提案する。

MMQLはJ.BackusのFP[BACK78]を基本概念とした関数型操作言語であり、大容量で長大なデータを対象とするデータベース検索操作を手順的な集合操作の適用という形で記述できる。また検索の中間結果をできる限り小さくなるような検索式の記述や埋め込み関数の定義、並列性抽出の容易さなどの点でマルチメディアデータベースの検索言語として有効な検索言語である。

3.1 データ操作言語

データベースは、実世界の情報を静的なデータで表現したものである。これに対し、データベースにアクセスしてデータベースの一部を切り出したり、データベースのデータを基に新たな計算を試みることを、一般にデータベースからのデータ操作と呼ぶ。また、データ操作を記述するためのプログラム言語の枠組をデータ操作言語(Data Manipulation Language: DML)と呼ぶ。

DMLは、データモデルを決定づける重要な要素である。例えば関係モデルの場合についていえば、関係代数(Relational Algebra)をベースに設計された関係演算(Relational Calculus)がデータ操作のための枠組として提供されている。また、関係演算を言語によって表現したDMLがSQLであり、関係モデルにおいては、データ表現手法もデータ操作手法も、関係と関係演算をもとに行なわれる。この結果、データベースの表現方法と検索によって得られる結果はどちらも関係という表の形式をしたデータである。このように従来型のデータベースでは、単純な数値や文字列の演算を集合演算としてとらえ数学的な枠組の上で実現されてきた。

しかしながら、マルチメディアデータベースに関してデータ操作を考えると、画像や映像、音声データなどの時間軸情報を含むような様々な形態をしたデータを対象にしなければ

ばならないばかりか、これらを合成した複合オブジェクトを取り扱えなければならない。また、メディアごとに演算方法がことなるため、これらのメディア依存処理をデータ操作の枠組の中に取り込まなければならない。したがってマルチメディアデータベースを実現するためには、従来の DML の実現方法とはことなり、

1. 様々なデータをカプセル化して、複合オブジェクトを生成できる
2. 大容量・長大なデータに対して、集合演算的にスマートに操作を記述できる
3. メディアに依存したデータ操作(メディア依存処理)を埋め込んで取り扱うことができる
4. 新たなメディア依存処理を、システムを停止せずに追加することができる
5. ネットワーク上に分散したデータベースに対しても、同一の枠組で透過的にデータ操作ができる
6. 超並列計算環境の基で、効率的に並列分散処理ができる

などの特性が要求される。

特に 4, 5, 6 については、ネットワーク技術の発展した現在では、データベースが単一の大容量で高速なホスト上で運営されているとは限らないという背景がある。複数のユーザーやアプリケーションから利用されることのあるデータベースでは、データを単一のホストに集約してしまうとネットワークの負荷や、記憶装置へのアクセスなどが集中し、結局全体としてのスループットを下げってしまう。これを解消するために、データをネットワーク上に分散して複数のデータベースサイトを集約すると一つの大きなデータベースを構成するというマルチデータベースアーキテクチャが提案されている。また、マルチメディアデータベースにおいては、メディア依存処理の分散も容易に考えることができる。例えば映像データについては、画像処理のための専用プロセッサを持つ計算機を用いることで処理の高速化を図ったり、並列分散型の計算機によって負荷分散を図ることが可能である。また、CG を用いる場合には、レンダリングエンジンを搭載した計算機を用いて行ったり、オブジェクト毎にことなるレンダリングエンジンを割り当てることなどが考えられる。

分散計算環境においては、計算機がいつでも同一のパフォーマンスを発揮できるとは限らない。また、同一のデータを持つ複数のサイトが存在し、サイトの選択法によっては処理を高速化できることが考えられる。この時、データ操作言語に求められる機能として、実際にデータ操作を行なう前に、事前評価として検索処理を開始した時点から終了するまでにどの程度の時間を要するかの概算することができ、どのサイトを利用するかの最適化が図れることは、データベースシステムのスループットを向上させるための重要な要素となる。検索操作の事前評価ができることは、宣言的に操作の記述ができ、データベースのインスタンスに依存せずに検索操作を実行できるような検索言語が望ましい。また、新たなハードウェアを導入してメディア依存処理をシステムを停止せずに組み込みたいなどの、動的なシステム変更にも対応できるようにするためには、検索言語に新たなオペレータを追加できるような操作系でなければならない。

そこで本研究ではデータ操作系として関数型操作系に着目し、マルチメディアデータベースの操作系のために、関数型操作系を基に設計されたデータ操作言語である MMQL (Multimedia Query Language) の提案を行なう。関数型操作系は、すべての操作を関数としてとらえ、操作対象のデータが関数によって処理された後、次の関数に渡され処理が進められていくデータ操作系である。関数型操作系の形式化としてはいくつかの方法が知られている [FIEL88] が、中でも FP は関数評価の定式化やユーザー定義関数の埋め込みなどの点でデータベース検索言語のモデルとして用いられている。FP を拡張したデータベース検索言語としては FQL [BUNE82] がよく知られている。FQL は遅延評価などの関数型言語の実現手法を用いて、検索評価をストリーム処理として実現している。しかし、マルチメディア情報を対象とする際に大量のデータが幾重にも処理される場合が起こり得、また、処理の並列化などはそれほど重要視されていない。マルチメディアデータベースを操作する際には、長大なデータに対して複雑な処理を効率よく行う必要があるため、本研究で提案する操作言語である MMQL では並列性の抽出により重点をおいている。

MMQL による検索の概念を図 3.1 に示す。以下の節で詳述するが、MMQL によって、

- 複合オブジェクトとしてデータを再構築できる
- 簡単な関数としてメディア依存オペレータを組み込むことができる
- あらゆる関数の特性まで見て最適化できる
- 分散並列アーキテクチャによって高効率に実装できる

ということが実現可能である。特に特徴的なのは、MMQL による検索操作とは、データベースから新たな情報を再構築できる点である。AIS モデルによってできる限り情報を細かく分解し、フラットにデータベースを構築する。これによってデータベースは様々な視点からの参照が可能となる。データベース利用者はこのデータベースに対し、MMQL によってデータベースの一部を切りだし、利用者の検索の意図に基づいて自由に複合オブジェクトを生成することができる。この結果、データベースには存在しないような新たなデータを検索によって再構築することが可能となる。

3.2 オブジェクト式

前節で述べたように、マルチメディアデータベースのための DML に要求される特徴の一つとして、複合オブジェクトを生成できることがあげられる。本研究では、図 3.1 に示すように、AIS モデルによってフラットに構築されたデータベースに対して、DML を用いてデータベースを参照し、新たな情報を複合オブジェクトとして再構築することを提案している。そこでまず本節では、この複合オブジェクトのデータ構造について述べる。

前節で述べた、階層構造を持つ複合オブジェクトを、一般的に形式化し、式の形で表したものが本研究で対案するオブジェクト式 (Object Expression: OE) である。本節ではオブジェクト式について述べる。

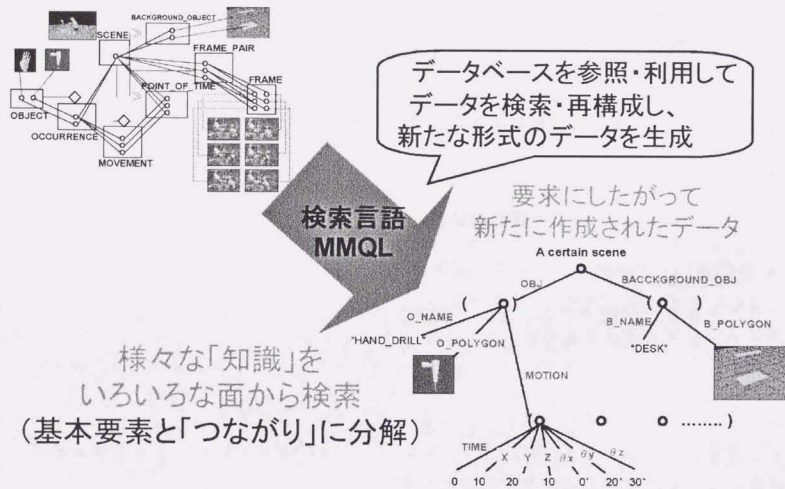


図 3.1: マルチメディアデータベースにおける検索操作

<オブジェクト式> ::= <原子> | <組> | <属性> (1)

<組> ::= <L bracket><属性>{<comma><属性>}+ <R bracket> (2)

<属性> ::= <属性名><colon><原子> | (3)

<属性名><colon><組> | (4)

<属性名><colon><属性> | (5)

<属性名><colon><列> (6)

<列> ::= <L brace><オブジェクト式> {<comma><オブジェクト式>}* <R brace> (7)

<原子> ::= 文字列 | 数値 | entity ID | 画像データ |
その他のマルチメディアデータ

<属性名> ::= 文字列

<L bracket> ::= “[”

<R bracket> ::= “]”

<L brace> ::= “{”

<R brace> ::= “}”

<colon> ::= “:”

<comma> ::= “,”

図 3.2: BNF によるオブジェクト式の構造の定義

前節の階層構造を持った複合オブジェクトを、伝統的に使用されてきた記述法と似た形式で得るために、図 3.1 の検索結果の複合オブジェクトを一つの式 (expression) で表現したものが、オブジェクト式 (Object Expression) である。AIS データベースの検索結果は、オブジェクト式で表現することができる。

オブジェクト式の構造を BNF 形式を用いて表したものを図 3.2 に示す。図 3.2 の定義において、属性は 4 種の構造の定義からなり、それぞれ属性名と属性の要素 (原子、組、属性、列) をコロン (“:”) で区切ることによって記述される。要素の構造をその属性の基本形 (cardinality) とよぶ。

さらに BNF で生成された木構造が階層構造を持つために列について、以下のような規則が加えられる。

- 列 $\{p_1, p_2, \dots, p_n\}$ について、すべての p_i が原子であれば、その列は階層的である。
- 列 $\{p_1, p_2, \dots, p_n\}$ について、 p_i が属性であるか組であるとき、すべての p_i について同一の属性名をもつ属性はすべて基本形が同じであり、かつその属性の要素すべてを含む列もまた階層的であれば、その列は階層的である。
- オブジェクト式において、すべての列は階層的でなければならない。

基本形の記述方式を、図 3.3 に示し、この規則によって示されたオブジェクト式の基本形をタイプ式 (Type Expression) と呼ぶ。すなわちタイプ式は、オブジェクト式より列の要

<タイプ式> ::= <原子型> | <組型> | <属性型>

<組型> ::= <L bracket> <属性型> { <comma> <属性型> }+ <R bracket>

<属性型> ::= <属性名> <colon> <原子型> |
 <属性名> <colon> <組型> |
 <属性名> <colon> <属性型> |
 <属性名> <colon> <列型>

<列型> ::= <L brace> <タイプ式> <R brace>

<原子型> ::= str | int | eID | pxm | その他のマルチメディアデータ型

図 3.3: BNF によるタイプ式の構造の定義

素を除外して、原子のデータ型と属性名によって、オブジェクト式の構造のみを表したものである。

あるオブジェクト式 o の基本形 (タイプ式) を求める関数を $type$ とし、 o に $type$ を適用することを

$type :: o \rightarrow type_expression$

と記述する。適用結果 $type_expression$ はタイプ式で、 o の基本形を表す。

ここではオブジェクト式とその基本形を、例を基に説明する。オブジェクト式の構造を図示した例を図 3.4 に示す。今、図 3.2(1) より、原子もまたオブジェクト式であり、整数型の原子オブジェクト式 o_{int_1}, o_{int_2} および o_{int_3} をそれぞれ、 $o_{int_1} = 3, o_{int_2} = 5, o_{int_3} = 7$ とすると、

$type :: o_{int_i} \rightarrow int$ (ただし $i = 1, 2, 3$)

である。また同様に、文字列型の原子オブジェクト式 o_{str_1}, o_{str_2} を $o_{str_1} = "abc", o_{str_2} = "def"$ とすると、

$type :: o_{str_i} \rightarrow str$ (ただし $i = 1, 2$)

である。

図 3.2(1) および (3) より、属性もまたオブジェクト式で、

$o_{attr_int_1} = A : o_{int_1} = A : 3$

また、 $o_{attr_int_1}$ の構造を図で示すと、図 3.4(a) のようになる。

次に、図 3.2(7) の列の定義より、

$\{o_{str_1}, o_{str_2}\} = \{ "abc", "def" \}$

は列であり、これ単体ではオブジェクト式ではないが、(6)、(1) より、属性名を付加することでオブジェクト式となる。

$o_{attr_seq_str} = B : \{o_{str_1}, o_{str_2}\} = B : \{ "abc", "def" \}$

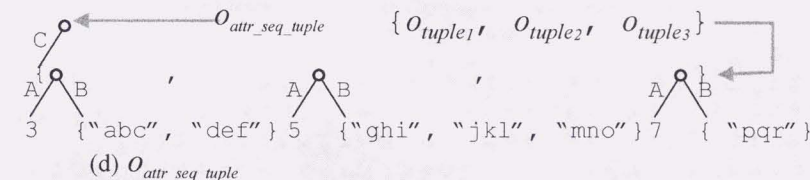
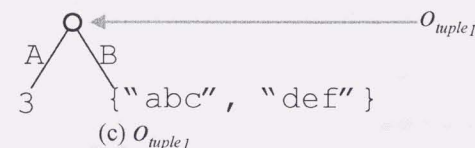
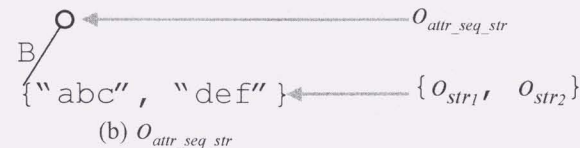
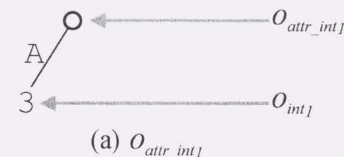


図 3.4: オブジェクト式

はオブジェクト式であり (図 3.4(b))、また、

$type :: o_{attr_seq_str} \rightarrow B : \{str\}$

である。

組は図 3.2(2) に示される構造を持つが、(1) より組もまたオブジェクト式である。すなわち、

$o_{tuple_1} = [o_{attr_int_1}, o_{attr_seq_str}] = [A : 3, B : \{ "abc", "def" \}]$

はオブジェクト式で (図 3.4(c))、

$type :: o_{tuple_1} = [A : int, B : \{str\}]$

である。

先に示した列の規則は、基本形を得る関数 type を用いて示せば、列 $\{p_1, p_2, \dots, p_n\}$ について、

$$\forall i, \forall j, \text{type} :: p_i = \text{type} :: p_j$$

であることを示す。これより、 $o_{\text{attr_seq_str}}$ のような原子列だけでなく、オブジェクト式の列を含むオブジェクト式を導出することができる。例えば、

$$\begin{aligned} o_{\text{attr_seq_tuple}} &= C : \{o_{\text{tuple}_1}, o_{\text{tuple}_2}, o_{\text{tuple}_3}\} \\ &= C : \{[A : 3, B : \{\text{"abc"}, \text{"def"}\}], \\ &\quad [A : 5, B : \{\text{"ghi"}, \text{"jkl"}, \text{"mno"}\}], \\ &\quad [A : 7, B : \{\text{"pql"}\}]\} \end{aligned}$$

ただし、 $o_{\text{tuple}_2} = [A:7, B:\{\text{"ghi"}, \text{"jkl"}, \text{"mno"}\}]$, $o_{\text{tuple}_3} = [A:5, B:\{\text{"pql"}\}]$ である (図 3.4(d))。ここで、

$$\text{type} :: o_{\text{tuple}_1} = \text{type} :: o_{\text{tuple}_2} = \text{type} :: o_{\text{tuple}_3} \rightarrow [A : \text{int}, B : \{\text{str}\}]$$

であるため、 $o_{\text{tuple}_1}, o_{\text{tuple}_2}$ および o_{tuple_3} は列の構成要素となり、

$$\text{type} :: o_{\text{attr_seq_tuple}} = C : \{[A : \text{int}, B : \{\text{str}\}]\}$$

である。

図 3.2 に示す定義、よりオブジェクト式は葉 (terminal node) にのみデータを持つような木構造であり、部分木もまたオブジェクト式である。これを特に部分オブジェクト式とよぶ。また、葉のデータを原子と呼び、これも部分オブジェクト式である。

原子には、文字・数値などのデータのほかに、ビットマップやピクセルマップによって構成される画像データ、さらに、データベースに格納されている主体識別子などがある。

オブジェクト式中で属性が単独でなく、2 個以上が組として扱われるとき、タプル記号 ("[]") を用いて表す。逆に、組から属性を削除した結果、そのオブジェクト式がひとつの属性しか持たなくなった場合、タプル記号を用いる必要はなくなり、自動的に消去される。

オブジェクト式が、従来の木構造と大きく異なる点は、枝 (node) に属性名が付される点と、列の規則である。上述した列の規則は、列の要素となる部分オブジェクト式がすべて同一の構造を持たなくてはならないことを意味している。ところで、オブジェクト式は、値を持たない属性を空 ("ϕ") として定義するため、列の要素となる部分オブジェクト式が、すべて同一の属性名を持っていなくても、対応する属性は空とみなされるため、同一の属性名の要素に関してのみ構造が等しいことが列の規則を満たす条件となる。

ところで、オブジェクト式では属性に名前が付されているため、ある特定の階層に属する部分オブジェクト式の集合を一意に指し示すことができる。オブジェクト式の階層構造において、特定の階層のことをランクとよぶ。特に、全体としてひとつのオブジェクト式に対する、先頭属性からのランクを絶対ランク、現在注目しているランクからのランクを相対ランクとよぶ。

ランクを指し示すためには、ランク記述子が用いられる。ランク記述子は、階層をたどる属性名をデリミタ "/" によって区切ることによって表される。先頭に "/" が記されるものを、絶対ランクを表す絶対ランク記述子、そうでないものを相対ランク記述子とよぶ。一般に、オブジェクト式の集合がランク記述子によって示される。

ランク記述子 RD から、オブジェクト式 o のランク RD の要素 (列または単値) を得る関数 rank は、

$$\text{rank}(\text{RD}) :: o \rightarrow \text{sequence of object expressions (or a single object expression)}$$

と記述できる。例えば前述した $o_{\text{attr_seq_tuple}}$ について $\text{rank}/C/A$ の要素を求めると、

$$\text{rank}(/C/A) :: o_{\text{attr_seq_tuple}} \rightarrow \{3, 5, 7\}$$

と、絶対ランクから見たランク C の属性 A の要素すべてを結合した列が得られる。

オブジェクト式を木構造としてみたとき、部分木の根 (internal node) には主体識別子は存在しない。この点が、文献 [NAGAE95] に示される主体木との違いであるが、ランク記述子などのその他の点については、大きな差はない。

オブジェクト式は、オブジェクト指向モデルのオブジェクトの概念とはことなり、単にデータを階層化しただけのものである。しかしデータベースをむしろ平坦に構築し、検索操作の段階で階層構造を構成することによって、データベースのアプリケーションからの独立性を保つと同時にユーザーにさまざまな見方を提供することを可能としている。例えば、ある主題の映像を再生するための検索だけでなく、フレームのある位置に映っている主題の情報やフレーム内の主題数を求めるなど、アプローチの逆転した検索を行うことも可能である。

また、オブジェクト式は列の定義により、同一の構造を持った大量のデータをランク記述子によって一意に指し示すことができるため、次章で述べる MMQL によって操作する際に、集合演算的な操作や並列性の実現に対して有効な実現方法を提供するが、その具体的方法については本稿では述べないこととする。

3.3 MMQL の基本構成要素

本節では MMQL の基本構成要素について説明する。MMQL の文法を図 3.5 に示す。MMQL による検索式は以下に示す通り、操作記述、単位操作、関数などを基として構成されている。

定義 1 関数 f(A)

関数 f とは、オブジェクト式を対象として、値や構造を返すような MMQL の要素で、定数や、あらかじめシステムによって与えられる原始関数、ユーザーによって定義されるユーザー定義関数などに分類される。ここで、A は関数の機能を決定づけるオプション引数として用いる。

< 検索式 > ::= < 単位操作 >
 | < 検索式 > < 結合子 > < 操作記述 >
 < 単位操作 > ::= < 属性名 > < colon > < 関数 >
 | < L-bracket > < 属性名 > < colon > < 関数 >
 { < comma > < 属性名 > < colon > < 関数 > }⁺ < R-bracket >
 | < OE 操作 >
 | < 定義検索式名 >
 < 操作記述 > ::= [< Alpha > < L-angle > < ランク記述子 > < R-angle >] < 単位操作 >
 | < Beta > < L-angle > < ランク記述子 > < R-angle >
 < L-parenthesis > < 論理式 > < R-parenthesis >
 | < Gamma > < L-angle > < ランク記述子 > < R-angle >
 | < Delta > < L-angle > < ランク記述子 > < R-angle >
 < 関数 > ::= < 原始関数 > | < 埋め込み関数 > | < 定数 > | < 条件関数 >
 < 条件関数 > ::= { < L-parenthesis > < 論理式 > < R-parenthesis > < arrow >
 < 関数 > < semicolon > }⁺ < 関数 >
 < OE 操作 > ::= < id > | < NULL >
 < ランク記述子 > ::= [< slash >] < ランク名 > { < slash > ランク名 }^{*}
 < ランク名 > ::= < 属性名 > | < 親属性 >
 < 原始関数 > ::= < generate > | < apply > | < id > | etc.
 < 論理式 > ::= < 真 > | < 偽 > | booleanexpression
 < 埋め込み関数 > ::= 文字列
 < 属性名 > ::= 文字列
 < 定数 > ::= 文字列
 < 結合子 > ::= “o”
 < L-bracket > ::= “[”
 < L-parenthesis > ::= “(”
 < L-angle > ::= “<”
 < semicolon > ::= “;”
 < Alpha > ::= “α”
 < Gamma > ::= “γ”
 < comma > ::= “,”
 < 真 > ::= “TRUE”
 < 親属性 > ::= “..”
 < NULL > ::= “φ”
 < arrow > ::= “→”
 < R-bracket > ::= “]”
 < R-parenthesis > ::= “)”
 < R-angle > ::= “>”
 < colon > ::= “:”
 < Beta > ::= “β”
 < Delta > ::= “δ”
 < slash > ::= “/”
 < 偽 > ::= “FALSE”

図 3.5: BNF 表記による MMQL の文法

一般に関数は、オブジェクト式または列を対象として、オブジェクト式または列を返す。したがって基本形として、あるオブジェクト式 o と、列 $\{p_1, p_2, \dots, p_n\}$ を考えたとき、それぞれに対する関数 $f(A)$ の適用を $f(A)::o$ および $f(A)::\{p_1, p_2, \dots, p_n\}$ と記述し、

case 1 $f_1(A)::o \rightarrow o'$ 単値から単値への関数
 case 2 $f_2(A)::o \rightarrow \{p'_1, p'_2, \dots, p'_m\}$ 単値から列への関数
 case 3 $f_3(A)::\{p_1, p_2, \dots, p_n\} \rightarrow o''$ 列から単値への関数
 case 4 $f_4(A)::\{p_1, p_2, \dots, p_n\} \rightarrow \{p''_1, p''_2, \dots, p''_m\}$ 列から列への関数

の 4 種類のいずれかである。

データベースアクセス関数として代表的なものに、次にあげる generate と apply がある。

• generate

関数 generate(TYPE) は、AIS ダイアグラムのスキーマにおけるある主体型名 “TYPE” を引数とし、適用対象は単値のオブジェクト式である。適用対象の内容に関わらず、返り値として TYPE の主体を列として返す。例えば、図 2.4 において、任意のオブジェクト式 o を対象として、

generate(DEPT) :: $o \rightarrow \{d1, d2\}$

を得る。基本形は $OE \rightarrow \{eID\}$ である。

• apply

関数 apply(DBF) は、単値の entity ID (eID 型の原子オブジェクト式) を対象とし、AIS ダイアグラムのスキーマにおけるデータベース関数 “DBF” を適用した結果を返り値として返す。例えば、図 2.4 において、eID 型の原子オブジェクト式 $d1$ を対象として、

apply(DeptEngageEmp) :: $d1 \rightarrow \{e3, e4\}$

を得る。基本形は、 $eID \rightarrow atom$ ないし $eID \rightarrow atom$ である。atom は原子型で、データベース関数に依存する。また、返り値が単値か列かは、データベース関数のパスに依存する。

MMQL のユーザーは、型と基本形を設計しさえすれば、自由に MMQL に関数を定義して良い。例えば典型的な関数の例として、整数平均値を求める関数 AVE は、基本形が {int} → int であり、列の平均値を整数化して返す関数として定義すれば良い。例として、

AVE :: {1, 2, 3} → 2

である。また、同様に列の要素数を求める関数 COUNT は、

COUNT :: {1, 2, 3} → 3

である。関数 AVE は整数メディアに依存したメディア依存処理の一つであると考えられることができる。画像データやポリゴンデータ特有の操作なども、メディア依存処理関数として定義すれば、MMQL に埋め込んで自由に用いることが可能である。

メディア依存処理を実現したユーザー定義関数として、例えば後述するように、概念を図 4.8 に示すメディア依存処理モジュールである FFO などがあげられる。また、最大値の導出などの、一般に集約関数と呼ばれる演算も関数として定義される。

定義 2 単位操作 N:f

単位操作 U とは、オブジェクト式に対する操作の最小単位であり、属性名 N と関数 f からなる。単位操作は、一つの (部分) オブジェクト式への操作を表し、与えられたオブジェクト式に新たに属性の追加・挿入を行なう。

原子オブジェクト式 $atom$ に対して、f の適用結果がオブジェクト式 o_r のとき、すなわち

$$f :: atom \rightarrow o_r$$

のとき、

$$N : f :: atom \rightarrow N : o_r$$

である。同様に、

$$f :: atom \rightarrow \{p_1, p_2, \dots, p_n\}$$

(p_i はオブジェクト式) のとき、

$$N : f :: atom \rightarrow N : \{p_1, p_2, \dots, p_n\}$$

である。つまり、関数 f の返り値に対し、属性名 N を付加した属性オブジェクト式を生成する。また、U の適用対象 o が属性オブジェクト式ないし組オブジェクト式である場合には、 $f:o$ の返り値に対し属性名 N を付加し、その属性を o に追加する。

単位操作は一つのオブジェクト式への操作であるため、あるランクに属するすべてのオブジェクト式に単位操作を適用するためには、後に述べるオペレータ “ α ” によって単位操作をそれぞれのオブジェクト式に分配すればよい。

単位操作によって始まる検索式は、一つのオブジェクト式への操作であるとみなすことができる。このような検索式もまた単位操作であり、これに名前をつけてマクロ化したものが定義検索式である。定義検索式は単位操作である。

例えば、 $EMP:apply(DeptEngageEmp)$ は単位操作であり、

$$EMP : apply(DeptEngageEmp) :: d1 \rightarrow EMP : \{e3, e4\}$$

である。同様に、

$$EMP : apply(DeptEngageEmp) :: d2 \rightarrow EMP : \{e1, e2, e3\}$$

である。

定義 2 より、与えられたオブジェクト式に対して関数を適用し、その結果に属性名を付してオブジェクト式に挿入する操作は、“属性” オブジェクト式を生成する単位操作である。この関数の適用は複数の関数を組としても良く、この場合 “組” オブジェクト式を生成する単位操作となる。

定義 3 操作記述

操作記述とは、操作対象となるオブジェクト式のあるランクに対する操作を記述したもので、検索式の逐次実行における 1 ステップを表す。したがって、操作記述はランク記述子を伴い、操作対象となるランクを明示する。

また、操作記述によって示される操作は大きく分けて 4 つに分類され、それぞれオペレータによって明示される。1 目はランクの生成・成長操作を表すオペレータ “ α ”、2 目はランクの要素の選択操作を表すオペレータ “ β ”、3 目はランクをオブジェクト式から削除するオペレータ “ γ ”、最後に、オブジェクト式の冗長な中間ランクを削除して再構成するオペレータ “ δ ” である。以下に、それぞれのオペレータについて説明する。

- 操作記述 $\alpha <RD>U$

α オペレータは、FP における “*apply to all*” に該当する。ランク記述子 RD によって示される列のそれぞれの要素に対し、単位操作 U を分配し、適用する。単位操作 U は、オブジェクト式に対して新たに属性を加える操作であるため、 α オペレータによってオブジェクト式に新たな属性が追加され成長する。

- 操作記述 $\beta <RD>E$

β オペレータは、FQL における “*restriction*” に該当する。ランク記述子 RD によって示される列のそれぞれの要素に対し、ブール代数式 E を分配し、適用する。適用結果が真となる部分オブジェクト式のみを残し、それ以外はオブジェクト式から削除する。 β オペレータによって、ランク RD の要素を条件 E によって選択することが可能であり、オブジェクト式の枝が剪定される。

- 操作記述 $\gamma <RD>$

γ オペレータは、関係演算における “*projection*” に該当する。ランク記述子 RD によって示される部分オブジェクト式すべてを削除し、同時に属性 RD もまた削除する。 γ オペレータによって、検索の中間結果として生じた不要な属性を、オブジェクト式から削除することが可能である。

- 操作記述 $\delta <RD>$

γ オペレータによって属性を削除していくと、属性オブジェクト式の直下の部分オブジェクト式として、属性オブジェクト式がつくという状態が生じる。この状態では、属性名が 2 重に付加され、冗長な属性名を持つランクが生じる。このとき、ランク記述子 RD によって示された冗長なランク (ランク名) を削除し、それに代わり RD の示す部分オブジェクト式と置き換えてオブジェクト式を再構成するのが δ オペレータである。

例えば、オブジェクト式 $A:[B:b, C:c]$ に対し、操作記述 $\gamma </A/C>$ を適用すると、

$$\gamma </A/C>:: A:[B:b, C:c] \rightarrow A:B:b$$

である。このとき、属性名 A と B は、どちらかだけが存在すれば良く、冗長に属性名が付加されている状態となる。このオブジェクト式 $A:B:b$ に対し、 δ オペレータを適用すると、

$$\delta </A/B>:: A:B:b \rightarrow A:b$$

である。このように、冗長な属性名 B を削除し、ランク記述子 A/B によって示された部分オブジェクト式 b をランク A の直下に置き換えたオブジェクト式として、再構成することができる。

関係演算に例えると、“ β ” オペレータは選択演算に、“ γ ” オペレータは射影演算に相当する。また、“ α ” オペレータでデータベース関数を分配適用する場合は自然結合演算に相当する。

定義 4 検索式

検索式は、操作記述を結合子 “ \circ ” によって結合したものである。MMQL における検索操作の評価とは、オブジェクト式に操作記述を逐次適用することによって、そのオブジェクト式の生成/成長/剪定を行なうことである。

検索式は一つの部分オブジェクト式に対して評価される。したがって、検索操作をはじめめるためにはオブジェクト式の木構造の「種」を必要とする。種となる対象を特に要さない検索は、根 (root) という特別なオブジェクト式を種として評価される。

また MMQL による検索式は、一つのオブジェクト式に対して操作を加え、結果としてオブジェクト式を得ることができる。したがって検索式もまた単位操作となり得、検索式を定義検索式としてマクロ定義することによって、ことなる検索式中において単位操作として用いることができる。定義検索式 U が “ α ” オペレータによってランク記述子 RD の各要素に分配適用される時、すなわち、

$$\alpha < RD > U$$

のとき、定義検索式 U に含まれるすべての絶対ランク記述子/URD は、ランク記述子 RD を基準としたランク記述子 RD/URD に置き換えられる。ここで、URD は U が対象とするオブジェクト式の根を基準とする相対ランク記述子である。すなわち、 U において絶対ランク記述子によって指定されたランクの要素とは、定義検索式を分配されるオブジェクト式集合であるため、このオブジェクト式を根としたランク指定に変換される。

以上に定義された要素が MMQL の基本構成要素である。

以下に「すべての社員 (EMP) の名前 (NAME) と給与 (SAL) を示せ」という検索を例に、MMQL の基本構成要素についてさらなる説明を加える。ここで、対象となるデータベースのインスタンスは、図 2.4 に示した例を用いることとする。

上記の検索は、次の検索式によって記述することができる。

```
SUBJ : generate(EMP)
      ◦α </EMP>[ NAME : apply(EmpEnameStr),
                SAL : apply(EmpSalaryInt) ]
```

この例では、2つの操作記述が結合されて一つの検索式を構成している。一方の操作記述は “SUBJ:generate(EMP)” であり、もう一方は “ $\alpha </EMP>[NAME:...,SAL:...]$ ” である。このように、MMQL における検索式は FP と同様に、結合子 “ \circ ” を用いて操作記述を結合したものによって表される。但し、評価順序は左から右の順に評価される。これは FQL と同様、どの順序でデータベースのパスをたどり、オブジェクト式が成長していくかを直観的に把握するためである。

この例の検索式において、前者の操作記述がまず評価される。この操作記述は、あるオブジェクト式に対して原始関数 **generate** を適用し、その結果として属性名 EMP としたオブジェクト式を得る単位操作を表す。単位操作とは、一つの (部分) オブジェクト式への操作であり、MMQL による操作の最小単位である。また属性名は、ランク記述子としてオブジェクト式に埋め込まれる。ここで関数 **generate(EMP)** は、適用対象に関わらず主体型 EMP に属するすべての主体を列として返す関数である。この単位操作は、「すべての従業員 (の存在) をランク EMP に示せ」という検索の意味を表す。この単位操作の結果、オブジェクト式

```
EMP : {e1, e2, e3, e4}
```

を得ることができる。e1, e2, e3, e4 は主体型 EMP に属する主体のインスタンスである。これらは、データベースではそれぞれ従業員を表す主体である。したがってこのオブジェクト式は、「ランク EMP は従業員の存在を示し、e1, e2, e3, e4 の 4 名の従業員が検索された」という意味を表す。

次に、このオブジェクト式を対象として、後者の操作記述が評価される。この操作記述は、オペレータ “ $\alpha </EMP>$ ” によって、ランク EMP の部分オブジェクト式 e1, e2, e3, e4 それぞれに単位操作 “[NAME:..., SAL:...]” が分配され評価される。すなわちオペレータ “ α ” は、FP の Apply to All と同様、列の個々の部分オブジェクト式に対し、単位操作を評価することを意味する。このように操作記述とは、あるランクの要素それぞれに対して単位操作を評価することを表している。

ここでまず、部分オブジェクト式 e1 を対象としてこの単位操作を評価することを考える。この単位操作に書かれている原始関数 **apply** は、データベース関数によって適用対象と結ばれている主体を列として返す関数である。この例では図 4.3 より、データベース関数 EmpNameStr によって主体 e1 と結ばれる主体が “Chen” であることがわかる。つまり、主体 e1 に対して関数 **apply(EmpEnameStr)** を適用すると、“Chen” が返される。同様に、データベース関数 EmpSalaryInt を適用するが、複数の操作を一つのオブジェクトに対して評価し、結果を組として得るために、タプル記号 (“[]”) が用いられている。

この操作記述は、「すべての従業員(ランク EMP)について、名前(属性 NAME) および給与額(属性 SAL)をデータベースから得て組にせよ」という検索の意味を表す。従って、後者の操作記述を評価した結果、ランク EMP の部分オブジェクト式はそれぞれ組オブジェクト式と入れ換えられ、次のオブジェクト式をこの検索式の検索結果として得ることができる。

```
EMP : { [NAME : "Chen", SAL : 2300],
        [NAME : "Backus", SAL : 2300],
        [NAME : "Codd", SAL : 2000],
        [NAME : "Arisawa", SAL : 1800] }
```

先述したオブジェクト式の意味は、さらに拡張されて、「ランク EMP は従業員の存在を示し、4名の従業員が検索された。これらの従業員には属性 NAME と SAL がランクとして付加され、それぞれ、従業員の名前と給与額を表す。例えば従業員の一人の名前は"Chen"であり、その給与額は 2300 である。」という検索結果が得られる。

このように MMQL では、ランク記述子によって操作を評価するオブジェクト式(部分オブジェクト式)が直観的に明記され、評価もそのランクのオブジェクト式のみに対して行われる。その結果、全体のオブジェクト式の中の一部分が置き換えられることによって、オブジェクト式の枝を成長させていくイメージで、評価が行われていく。この点が FP や FQL と大きく異なる点である。MMQL では長大なオブジェクト式に対して一部分を置き換えていくだけで、全体に操作を加えることができるため、莫大な量のストリーム処理を回避することが可能である。また、オペレータ "α" によって示される操作演算は、同じランクの他のオブジェクトとは独立であるため、並列評価を行うことが容易である。

この検索式に次の 1 行を加える場合を考えよう。

```
οβ </EMP>(SAL > 2200)
```

オペレータ "α" と同様の、ランクを構成する部分オブジェクト式すべてに対するオペレータとして、"β" が存在する。オペレータ "β" は、FQL によって提案された選択演算(restriction)で、集合の個々の要素に対し論理式として書かれた条件が真となる要素だけを残り、そうでない場合には空(φ)と置き換える。この例は、ランク/EMP の各要素に対し、属性 SAL の値が 2200 を越えるものだけを残り選択演算である。この操作記述は、「すべての従業員(ランク EMP)について、その給与額(ランク SAL)が 2200 を越えるものだけを選択せよ」という検索の意味を表す。この結果、先のオブジェクト式から組 "[NAME: "Codd", ...]" および "[NAME: "Arisawa", ...]" が削除され、

```
EMP : { [NAME : "Chen", SAL : 2300],
        [NAME : "Backus", SAL : 2300] }
```

が得られる。このオブジェクト式の意味は、「ランク EMP はその給与額が 2200 を越える従業員の存在を示し、2名の従業員が検索された。これらの従業員には属性 NAME と SAL が

ランクとして付加され、それぞれ、従業員の名前と給与額を表す。例えば従業員の一人の名前は"Chen"であり、その給与額は 2300 である。」という検索結果が得られる。

さらに、検索結果のオブジェクト式に属性 SAL が不要であった場合、次の 1 行を加えることによってこの属性を削除することができる。

```
ογ </SUBJ/SAL>
```

"γ" も、"α" や "β" と同様、ランクを構成するすべてのオブジェクト式に対するオペレータで、指定された属性がオブジェクト式から削除される。この操作記述は、「それぞれの従業員の給与額(ランク SAL)を削除せよ」という検索の意味を表す。

最終結果として、次のオブジェクト式を得ることとなる。

```
EMP : { NAME : "Chen",
        NAME : "Backus" }
```

結論するとこの検索式は、「すべての従業員のうち、2200 を越える給与を得ている社員の名前を示せ」という検索を示していたことになる。

ところで上述した原始関数とは、MMQL のインタプリタシステムによってあらかじめ定義され、ユーザーに与えられる関数であり、ここでは generate や apply について述べた。これらは次章で述べるシステムコンポーネントの中の、MMQL インタプリタによってあらかじめ定義された関数である。

原始関数とは対照的に、システムによって定義されていなくても、ユーザーが定義することによって集約関数やメディア依存処理を埋め込むことが可能であれば、ユーザーはマルチメディアデータ特有の操作関数をデータベース検索に用いることが可能となる。MMQL ではこのようなユーザー定義関数を埋め込み関数として扱うことが可能である。例えば後述するように、レシビを解釈して加工画像を生成するメディア依存処理モジュールを作成したとき、それを埋め込み関数として、先の原始関数と同様に MMQL のインタプリタで呼び出すことができる。

本節で述べた通り、MMQL の基本構成要素は、操作記述と単位操作、ランク記述子を伴うオペレータ "α"、"β"、"γ"、"δ"、それに原始関数や埋め込み関数などの関数である。

3.4 まとめ

本章では、AIS モデルに基づくデータベースの操作言語である MMQL について述べた。MMQL によって、平坦に構築されたデータベースから自由に階層構造を生成することが可能である。MMQL は、操作対象となるオブジェクト式を直観的に把握しやすく、また、並列性抽出の容易さや埋め込み関数の定義など、マルチメディアデータベース操作言語として有効な言語である。

我々は UNIX ワークステーションを用いて、映像データベースシステムのプロトタイプの開発を進めている。映像データベースプロトタイプについては次章で述べる。しかし、

本文中に述べたように MMQL は中間結果を小さくするための最適化が可能であるが、その手法についての検討は今後の課題である。さらに検索処理の並列化の実装をはじめする、真のマルチメディアデータベースシステムの開発については、今後の研究が待たれている。

Chapter 4

映像データベースの構成と実現

本章ではマルチメディアデータモデリングに固有の問題やその解決手段を考察するために、マルチメディアの典型的な例である映像をデータベース化することを考える。本研究でデータベース化する対象となる映像データは、フレームの列で構成されるカットなどの単位からなり、映像中に様々な知識を映し込んだものであるとする。すなわち、映画や娯楽のための長時間の映像ではなく、何らかの主題を持つ短時間のフレーム列の集合である。アプリケーションとして例えば、エンジニアリング分野を応用対象と考えた場合、一つの映像に含まれる情報は多様で、そこから得られる情報は単なるフレーム列だけではなく、被写体やそれらの関係などを表現しなくてはならず、マルチメディアデータベースシステムを実現する上で、解決されていない問題が数多く存在するためである。マルチメディアデータベースを構築するにあたり、本研究では特にビデオ映像データをどのようにモデル化・蓄積するかに着目する。そこで本章では、被写体や論理カット、マーカといった単位に映像情報を分解し、それらを AIS モデルを基としてデータベース化した、映像データベースの構築例について述べる。

4.1 視覚情報としての映像データと映像データベース

人間は視覚情報からより多くの情報を得ることができる。映像データを計算機上で取り扱うことが容易になってきた現在、映像データを中心とした視覚情報システムの要求が高まってきている。特に、マルチメディアの中で一番一般的でかつ多くの課題を抱えているものは映像データをどのようにデータベース化するかという問題である。本節では、映像データの特徴について述べる。

映像データの特性を、物理的な側面と論理的な側面から考察すると、次のような問題をあげることができる。

1. 大規模かつ多様性

映像データの物理的な側面を考えると、まず時間軸を持った大容量で長大なデータであることがいえる。一般に映像データは、静止画像の列としてとらえられ、連続

する静止画像をある時間軸に合わせて表示することで動画として視覚化することができる。この静止画像の精度は、一般的な NTSC 映像では横 640 × 縦 480 ピクセルの画像データであり、1 ピクセルあたりについて赤、青、緑成分に関し 256 階調を持たせることが多い。この時、1 ピクセルは 3 バイトで表現できる。また、時間軸方向については、1 秒間に 30 フレームの精度を持つ。ここでフレームとは、動画を構成する 1 枚の静止画像を意味する。したがって、映像データの 1 秒間当たりのデータ量は $640[\text{pixels}/\text{row}] \times 480[\text{pixels}/\text{column}] \times 3[\text{byte}/\text{pixel}] \times 30[\text{frame}/\text{sec}] = 27,648,000[\text{byte}/\text{sec}]$ にものぼる。

現在では、映像に関しては、Motion JPEG や MPEG などの、視覚的には高精度でかつ高効率に圧縮する技術がある。しかし、これらの高効率圧縮アルゴリズムはすべて非可逆圧縮である。映像データベースは、アプリケーションによってはデータベースに蓄積された映像を素材として再利用したり、後で解析を行なってそこから新たな情報の抽出を試みるが見込まれる。例えば、映像の一部を補完しながら拡大して見たいなどの要求がある。この観点からは、元の情報を欠落させてしまう非可逆圧縮アルゴリズムは、映像データベースには不向きであり、結局、上にあげたレートのデータをそのまま取り扱うことのできる映像データベースシステムが望ましい。

さらに、多視点、高精度な映像になると、データ量の問題は顕著に現れ、単に計算機にデジタルデータとして取り込むことも困難になってくる。また、映像データに対して計算機上で解析や加工などの処理を行なうためには膨大な計算機資源を消費する。

2. 多義性

映像検索をする場合、映像中に映り込んでいる情報を扱うことを考えると、それらの情報は、被写体や背景、動作は変化など、そのフレーム列の表しているなんらかの「ことがら」である。一つの映像には、これらの情報が多種、多様に含まれており、論理的にも大容量の情報であるといえる。このような、映像の内容を基にした検索を Content Based Retrieval (CBR) と呼ぶ。ここでの Content とは、多分にデータベース利用者 (検索者) の意図を含んでいる。このため、CBR を実現する映像データベースシステムを実現するためには、次の二つの面からの考察を欠かすことができない。

一つは、映像に映り込んだ Content をどのような手法・要素技術によって抽出するかといった技術的側面である。画像処理技術や画像認識技術の発展によって、CBR は現実味を帯びてきた。しかし実際の認識に際し、膨大な計算機資源と時間を消費しなければ検索の実行ができないようなシステムではデータベースシステムとして有効であるとはいえない。

そしてもう一方は、映像の持つ多種多様な情報をどのようにとらえ、モデル化するのかといった議論である。要素技術によって、CBR のために抽出された論理的な情報を、検索の際に有効に利用できるようなモデル化手法が必須である。

3. 一次情報性

映像データは、計算機に取り込まれた時点では単なる画像列データであり、そこに映り込んだ被写体の情報は何ら付加されていない。また、それぞれのフレームは二次元のピクセルマップであり、個々のピクセルは単にある一点の色を表したバイナリデータであるに過ぎず、ピクセルそのものは何の意味情報も含まない。このようなデータを対象として CBR を実現するためには、データに対して解析を行ない明示的なインデックス情報を二次情報として付与しておく必要がある。これは、二次情報を抽出するための解析手法の開発が必須であるなどの、要素技術的な問題を含んでいることを意味する。

以上の問題を解決した上で、CBR を実現することが可能な映像データベースシステムを構築するためには、映像をモデル化し、データベースに蓄積するための手法の確立と要素技術の開発、そして、データ操作手法の確立が必須である。

4.2 映像モデリング

ビデオ映像を計算機で扱うデータ構造としてとらえると、時間軸に沿って順次フレームが再生されていくデータであると定義することができる。たとえば、NTSC 信号によって表されるビデオ映像は先に述べた通り、1 秒間あたり 30 フレームの静止画像 (瞬間画像) によって構成され、それぞれのフレームは、640 × 480 ピクセルの画素 (ピクセルマップ) によって構成されていると考えるのが標準的である。本稿ではビデオ映像を時間軸をもつフレーム列として定義する。

ところで、ユーザーが検索の対象とするのは、単なるフレーム列の集合ではなく、ビデオ映像に映っている被写体や作業など、そのフレーム列の表している「ことがら」である。したがって、このような、ビデオ映像に写し込まれた「ことがら」をどのようにモデル化し、表現するかということは、ビデオ映像データベースを構築する際の重要なテーマとなる。このことは本研究の対象となるエンジニアリング分野では特に重要である。例えば、作業要素の映像データベースを構築する場合、各フレームには、作業の対象物、工具、治具、作業者の手、搬送系など、いろいろな「もの」が映り込んでおり、検索はこれらの「もの」の位置や相互関係をキーとして行なわれることが多い。

このような被写体を中心としたデータベース化を行なうために、我々は物理カットと論理カットと呼ぶ 2 つの要素を提案した。物理カットおよび論理カットの概念を図 4.1 に示す。

- 物理カット…物理カットは、ある一連の作業やストーリーなどの、一本のビデオ映像を構成する大きな意味的な区切りに対応するフレーム列を表す単位である。たとえば、「ロボット R が、工具 T を用いて、機械 M の部品 P を組み立てる作業」を撮影した一連の映像は、物理カットであり映像データベースに蓄積される単位である。

また、ひとつの物理カットを構成するフレーム列は、他の物理カットのフレームとは排他である。

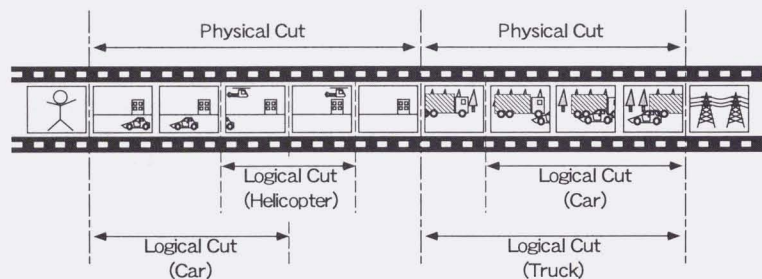


図 4.1: 物理カットと論理カット

- 論理カット…論理カットは、物理カット内における、個々の被写体などの主題を表すフレーム列である。たとえば、上記物理カットの例において、ロボット R や部品 P が映っているフレーム列はそれぞれ、ロボット R や部品 P を表す論理カットである。また、ロボット R と部品 P は同じフレームに同時に映っていることがある。このため、物理カットとはことなり、論理カットは互いに重なりあってもよい。

論理カットは物理カットごとに何が映っているかを分析して定義されるものであり、従って物理カットを跨ぐ論理カットは存在しない。但し、ある被写体がいろいろな物理カットの中に現れることはあり得るので、「ある被写体についての論理カット」は複数存在し得る。

論理カットを定義することによって、ユーザーはビデオ映像中の個々の主題または被写体を対象とする検索を行うことが可能となる。さらに、それがフレームのどの位置にどのような形状で存在しているかを示す情報として、それぞれのフレームに対してマーカという概念を導入する。

- マーカ…マーカは、論理カットの個々のフレームについて、主題がフレームのどの位置にどのような形状で存在しているかをビットマップや外郭線情報などを用いて示すものである。

マーカとして用いられるのは、ビットマップによって被写体の位置・形状を示すマスク、ベクトル列を用いて外郭線形状を示すアウトライン、被写体を囲む円を中心座標と半径で示すスポットがある。マーカは、検索の際に被写体を強調する必要があるとき、画像処理のパラメータとして用いられる。

一連のビデオ映像から物理カットや被写体（マーカ）に関する情報を抽出することは、映像データベースシステムにとって重要な要素技術であるが、画像処理の分野ですでに多くの支援技術が提案されている [UEDA93, NAGAS92]。したがって本稿では、これらの技術そのものについては議論しないこととする。

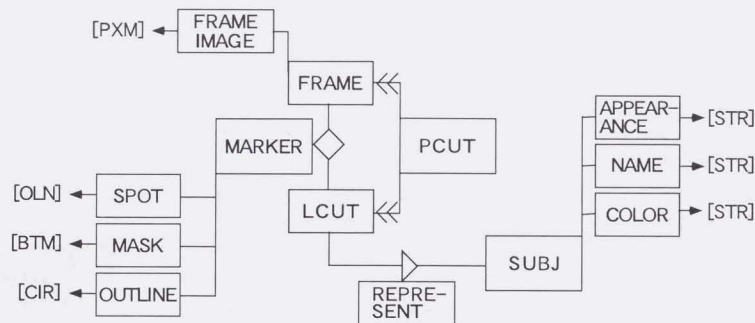


図 4.2: 映像データベースのスキーマ

主題、論理カット、マーカの情報を後に述べる関数型操作言語で任意に組み合わせることにより、非常に多岐にわたる検索が可能になり、また多様な検索結果の映像を得ることが可能となる。以降ではこれらの構成要素を基に、映像データベースの構築について論ずる。

4.3 AIS モデルによる映像データベースの構成

AIS ダイアグラムを用いて映像データベースのスキーマを定義すると、図 4.2 に示されるようなダイアグラムを得ることができる。

このデータベーススキーマでは、映像を構成するフレーム列の画像データやマーカのビットマップデータなどと、主題の名前などを表す文字データ、さらにはそれらの「意味的な」関係が統一的な手法で記述されている。

このスキーマのもっとも主要な構成要素は、映像に映り込んだ主題 (SUBJ) を表す主体と、物理カット (PCUT) を表す主体である。

フレーム列 (FRAME) が物理カットの多値成分として定義され、さらにそれぞれのフレームには、フレーム画像 (FRAMEIMAGE) がピクセルマップ (pixelmap: PXM) のデータ型で単値成分として蓄積される。また一つの物理カットにはさまざまな主題が論理カットとして映り込んでおり、また物理カットを跨ぐ論理カットが存在しないことから、論理カット (LCUT) も一つの物理カットの多値成分として定義される。

一方、映像を検索の対象とする場合には、物理カットとフレーム列のような物理的な構成要素を直接検索するよりも、ある「ことがら」を対象としてデータベースの検索を行なう必要がある。そこで、ユーザーが映像を見ることによりはじめて一意に認識される、映像に映り込んだ「ことがら」を主題として定義する。主題を映像に映り込んだ被写体に限

定すると、被写体の名前 (NAME) や色 (COLOR)、外観 (APPEARANCE) などのさまざまな属性を主題の成分主体として定義することができ、それぞれ文字列 (string: STR) の値を持つ。

被写体と論理カットの間には前節で述べた通り、「ある被写体についての論理カット」が複数存在し得るという関係がある。この関係は、一つの主題がさまざまな物理カットに論理カットとして現れる (REPRESENT) ことから、主題と論理カットは、1:n の関係を表す関連主体によって結ばれる。

また、論理カットを構成するフレーム列は、フレームと論理カットの関連主体であるマーカー (MARKER) によって示される。マーカーは、論理カットのそれぞれのフレームにおける主題の位置・形状情報を示す主体であり、成分主体としてフレーム内での被写体形状情報や検索時の画像強調データを保持することとする。マーカーが論理カットとフレームの関連主体によって定義されるのは、一つの論理カットは複数のフレームによって構成され、一つのフレームにはいくつかの被写体が映っていることによるためである。実際には、マーカーは論理カットをデータベースに蓄積する際に同時に用意され、被写体形状をビットマップ (bitmap: BTM) で表したマスク (MASK) や被写体の外郭線 (outline: OLN) を表すベクトル列、また、中心座標と半径による円情報 (circle: CIR) の「スポット」データが検索時の被写体強調に用いられる。

映像データベースのインスタンスの例を図 4.3 に示す。

例えば図 4.3 において、主体型 SUBJ の主体 $s1$ は、成分主体 $sa1, sm1, scl$ を持ち、それぞれ文字列 "STICK", "SCREWDRIVER", "RED" と結ばれている。これは、「赤い」色の「棒状」をした「ねじ回し」が主題として存在する事実を表し、その主体識別子は $s1$ であることを示す。

次に、2つの論理カット $l1, l2$ は、関連主体 $r1, r2$ によって主体 $s1$ と結ばれている。これは、関連主体 $r1, r2$ によって $l1, l2$ が主題 $s1$ を表す論理カットであることを示している。

また、論理カットとフレームの間に関連主体 MARKER が定義され、それぞれの論理カットのあるフレームにおけるマーカーが示されている。例えば上記の論理カット $l2$ については、マーカー $m4, m5, m6$ が定義されている。

ここで、データベース関数 *SubjRepresentLcutMarker* を考えると、インスタンス $s1$ については列 $\{m4, m5, m6\}$ が、パス $r1, r2$ 及び $l1, l2$ を経て導出されており、主題 $s1$ に関するすべてのマーカーを求めることができる。同様にインスタンス $s2$ については列 $\{m1, m2, m3\}$ が導出される。逆に、データベース関数 *MarkerLcutRepresentSubj* を考えると、インスタンス $m5$ については単値 $s1$ が、パス $l2, r2$ を経て求められ、マーカー $m5$ の示す主題を求めることができる。このようにデータベース関数は、定義域の1つのインスタンスに適用すると、そのインスタンスと結ばれている単値または列 (多値) を返し、パスを特定した双方向の参照が可能である。したがって、インスタンスを単純な2項関係としてとらえることができ、様々な角度から情報をとらえることができる。

この図に示した通り、すべての主体型はデータベーススキーマ中で一意である主体型名が英字大文字によって付けられ、データベース関数はパスとなる主体型名を、先頭だけ大文字としてつなげ合わせた文字列によって表わすこととする。

図 4.3 に示す通り、主体は内部構造を持たず、データベースは単純な2項関係のみによ

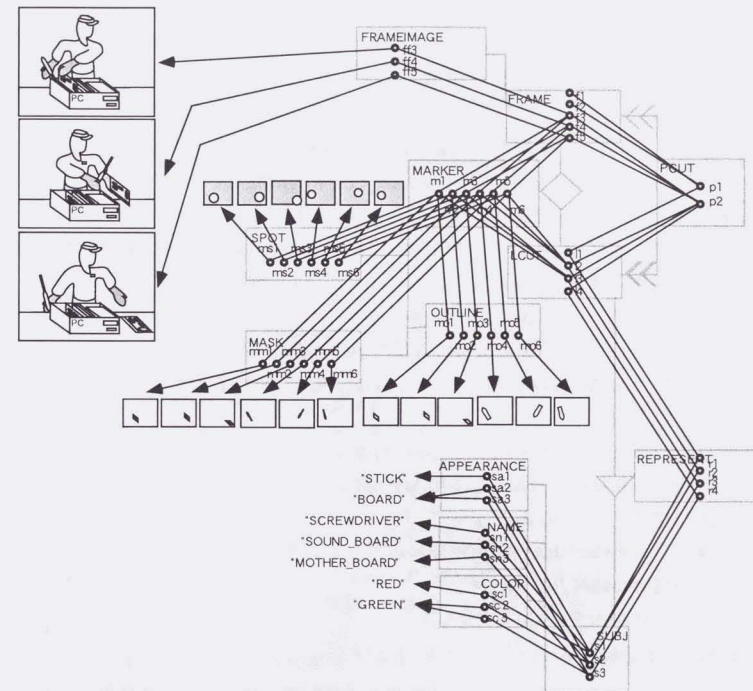


図 4.3: 映像データベースのインスタンスの例

て構成される。また、自由に2項関係のパスを設定できるデータベース関数を用いることにより、すべてのつながりを2項関係としてとらえることができる。したがって、AISモデルに基づくデータベースはマルチメディアデータも含めて、情報構造が平坦な形で記述され、さまざまな見方を提供することが可能である。

4.4 映像データベースのデータ操作

本節では、映像に映った主題について検索し、さらに主題にスポットやアウトラインなどの強調をした映像を得る検索を例に、MMQLによる映像データベースの検索について説明する。

ここでは、「ネジ回しを用いてパソコンからサウンドボードを取り外す作業について、サウンドボードにはスポットを当て、ネジ回しにはアウトライン強調をし、検索対象となった

```

1: PCUT : generate(PCUT)
2:  o $\alpha$  < /PCUT > [LCUT : apply(PcutLcut), FRAME : apply(PcutFrame)]
3:  o $\alpha$  < /PCUT/LCUT > [NAME : apply(LcutRepresentSubjNameStr),
                        MARKER' : apply(LcutMarker)]
4:  o $\beta$  < /PCUT > ("SOUND_BOARD" in rank(LCUT/NAME)
                  and "SCREWDRIVER" in rank(LCUT/NAME))
5:  o $\beta$  < /PCUT/LCUT > (NAME = "SOUND_BOARD" or NAME = "SCREWDRIVER")
6:  o $\alpha$  < /PCUT/FRAME > [IMAGE : apply(FrameFrameImagePxm),
                        BACKGROUND : 80,
                        MARKER : apply(FrameMarker)]
7:  o $\beta$  < /PCUT/FRAME/MARKER > (id in rank(/PCUT/LCUT/MARKER'))
8:  o $\gamma$  < /PCUT/LCUT >
9:  o $\alpha$  < /PCUT/FRAME/MARKER > MARKGEN
10: o $\alpha$  < /PCUT/FRAME > TARGETFRAME : FFO
11: o $\gamma$  < /PCUT/FRAME/IMAGE >
12: o $\gamma$  < /PCUT/FRAME/BACKGROUND >
13: o $\gamma$  < /PCUT/FRAME/MARKER >
14: o $\delta$  < /PCUT/FRAME/TARGETIMAGE >
15: oNAME : {"SCREWDRIVER", "SOUND_BOARD"}

```

図 4.4: MMQL による検索式

名前 ("SCREWDRIVER" および "SOUND_BOARD") と映像とを組にした複合オブジェクトを求めよ」という検索について考える。

この検索式を、図 4.4 に示す。この図において、行番号は便宜的につけられたものであり、実際の検索式には行番号をつける必要はない。

この検索式の 1 行目から 3 行目までが評価されると、前節での説明のようにオブジェクト式が成長していく。

次に選択演算である「ランク/PCUT のそれぞれの部分オブジェクト式に対して、相対ランク LCUT/NAME に "SOUND_BOARD" と "SCREWDRIVER" を含んでいるものだけを残せ」という操作を 4 行目で行なっている。4 行目の論理式に用いられている関数 rank(LCUT/NAME) は、ランク記述子によって指定されたランクの部分オブジェクト式を集約して返す関数である。論理式の適用対象となる要素は、オペレータ " β " によってランク/PCUT の部分オブジェクト式であることがわかる。従って、ランク/PCUT の部分オブジェクト式それぞれについて、その下の相対ランク LCUT/NAME の値の集合がこの関数によって返される。また、

```

Def MARKGEN  $\equiv$  [SPOT : (apply(MarkerLcutRepresentSubjNameStr) = "SOUND_BOARD")
                 $\rightarrow$  apply(MarkerSportCir);  $\phi$ ,
                OUTLINE : (apply(MarkerLcutRepresentSubjNameStr) = "SCREWDRIVER")
                 $\rightarrow$  apply(MarkerOutlineOln);  $\phi$ ,
                OP : (apply(MarkerLcutRepresentSubjNameStr) = "SOUND_BOARD")
                 $\rightarrow$  "KEEP";
                (apply(MarkerLcutRepresentSubjNameStr) = "SCREWDRIVER")
                 $\rightarrow$  "FILL";  $\phi$ ]

```

図 4.5: 定義検索式 MARKGEN

集合演算子 in は、ある値が集合に含まれているかどうかを判定する演算子である。

また、7 行目の関数 id は、対象となるオブジェクト式そのものを返す関数である。従って、7 行目の操作記述は、「ランク/PCUT/FRAME/MARKER のそれぞれの部分オブジェクト式に対して、ランク/PCUT/LCUT/MARKER' にも存在するものだけを残せ」という操作を意味する。

操作の段階で、それ以降不要となる属性は削除することができる。8 行目の操作記述には、オペレータ " γ " によってランク/PCUT/LCUT が示されている。これは、このランクがこれ以降必要ないため、ランク/PCUT の属性 LCUT は、オブジェクト式からすべて削除される。

9 行目では、定義検索式 MARKGEN が用いられている。定義検索式は、MMQL の検索式をより直観的にするため、部分検索式を別に定義して用いられる。

定義検索式 MARKGEN を図 4.5 に示す。検索式 MARKGEN は、条件関数によって表される単位操作ひとつで構成されている。条件関数は、条件によって異なる関数を用いるために使われ、括弧 "(" によって示される論理式と、記号 " \rightarrow "、そしてその論理式が満たされたときに用いられる関数の 3 つの組で一組となり、それぞれの組が、セミコロン ";" によって区切られて記述される。また、当てはまる条件が存在しないときには、最後に書かれた関数が実行される。この例では、2 通りの条件によって別々の関数が用いられ、条件が満たされない場合には関数 " ϕ " が用いられる。関数 " ϕ " は、空を返す、すなわちなんの操作も行わない定数関数を表す。条件関数もまた、関数である。この例では、「対象となるオブジェクト式（ここではマーカを表す主体）の、データベース関数によって結び付けられる主題名が、"SOUND_BOARD" ならスポットライト強調を、"SCREWDRIVER" なら外郭線強調を、強調効果として得られるためのデータを生成せよ」という単位操作を意味する。

図 4.4 の操作記述の 1 行目から 9 行目までが評価された結果、中間結果として図 4.6 に示すオブジェクト式を得ることができる。このオブジェクト式の構造を図示すると 4.7 を得る。この中間結果は、前章で述べたとおり、強調された映像を得るためのレシビであり、ランク/PCUT/FRAME の部分オブジェクト式はそれぞれ、強調されたフレームを生成するため

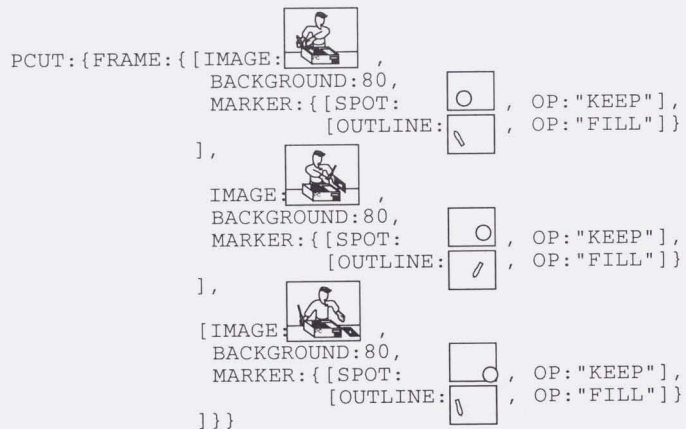


図 4.6: 中間結果 (オブジェクト式)

に必要なデータをすべて含んでいる。従って、図 4.8 に示した強調画像を生成するメディア依存処理モジュールである Frame-by-Frame Operator (FFO) にこのデータを渡すことにより、ターゲットとなる強調されたフレームを得ることができる。

検索式の 10 行目には、FFO が埋め込み関数として記述されている。特に FFO は計算機負荷の大きい処理であるが、“α” によって並列処理可能性が明記されているため、有効な実装を行うことが可能である。

最後に検索の最終結果には不要となる属性を削除していくが、その結果、属性名が冗長につけられることがある。この例では、13 行目までの操作記述が評価された結果、ランク/PCUT/FRAME の部分オブジェクト式は、属性 TARGETIMAGE だけしか持たないオブジェクト式となる。つまりランク/PCUT の部分オブジェクト式を見ると FRAME:TARGETIMAGE のように二つの属性名が重なっているが、これはひとつの属性名とみなすことができる。このような場合に、オブジェクト式操作として、冗長な属性名をひとつの属性名に改名することができる。14 行目では、“δ” オペレータによって、冗長な属性名 FRAME:TARGETFRAME を FRAME に改名している。

この検索式の操作記述がすべて評価された結果、図 4.9 に示すオブジェクト式を検索結果として得ることができる。

この検索によって得られる映像の例を図 4.10 に示す。この図は、後述する映像データベースの試作システムを用いて同様の検索を行ったときに得られた映像である。

このように検索によって、検索対象となったことがらスポット強調やアウトライン強調によって、検索の題意をより明示した映像を得ることができる。また、このような主題

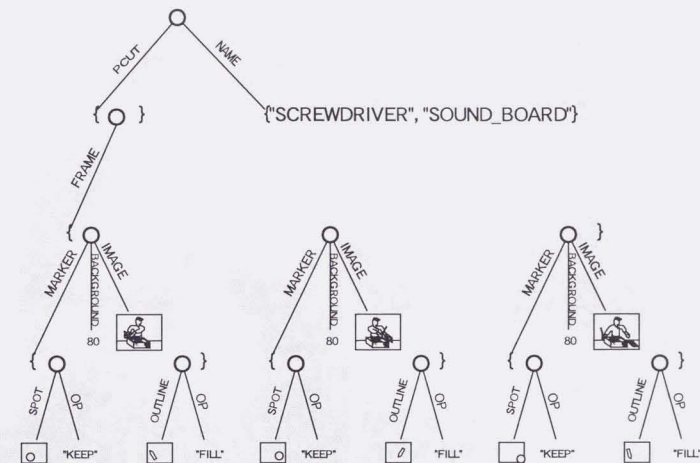


図 4.7: 中間結果の構造

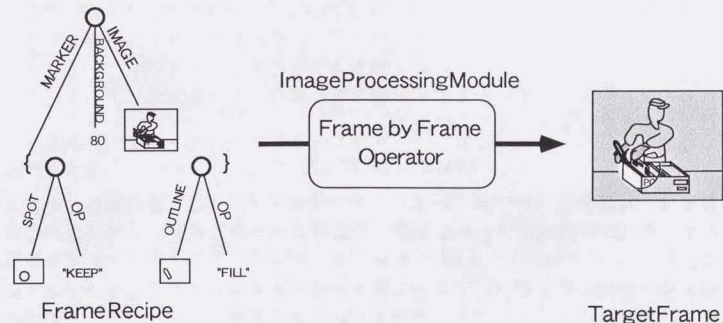


図 4.8: Frame by Frame Operator の概念

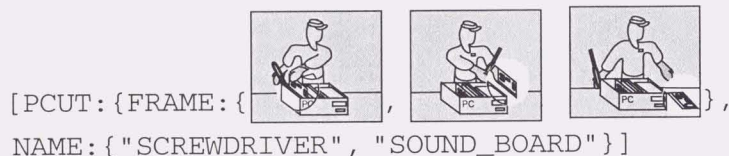


図 4.9: 検索結果

の強調処理は、検索の過程で任意に組み合わせることが可能であり、一つの映像に対して様々な検索結果を得ることが可能である。

4.4.1 様々な検索例

AIS モデルと、MMQL によって構成されるマルチメディアデータベースにおいては、平坦な構造をしたデータベースから自由に複合オブジェクトを生成することが可能である。本節では、関係データベースと SQL 言語や、オブジェクト指向データベースでは困難であると考えられている検索の例を示し、本研究における方法の表現力と柔軟性を明らかにする。

まず、特定のピクセルマップを与えられ、「このフレームに存在する主題の数を求めよ」という検索要求がなされたとする。ここでの主題とは、映像データベースにおいてインデックスづけされた被写体を指すこととする。この検索は、そのフレームを対象として、図 4.11 の検索式によって解を得ることができる。

図 4.3 の例ではピクセルマップの値主体であるのフレームが 3 件存在するが、このデータベースインスタンスはこれらのフレームについて、いずれもマークが 3 件存在することが表現されている。これは、フレーム中に主題が 3 件存在することを示している。したがって、これらのフレームのいずれか 1 件を対象とした場合、この検索式によって得られる検索結果は、次のオブジェクト式で示される。

FRAME: {COUNT: 3}

この結果は、対象となったフレーム中に、データベースに登録・蓄積された主題の数が 3 件であり、それ以外に同じフレームを用いた論理カットが存在していなかったことを示している。

オブジェクト指向モデルでは、フレーム列を一つのオブジェクトとしてカプセル化するため、そこに映り込んだ「ことがら」のモデル化に工夫を要し、元のフレームに何が映っているのかというような、逆転したアプローチの検索は困難であると考えられている。

次に、同じデータベースに対し、「物理カット毎に、存在する論理カットのフレーム数を求め、それらの平均よりも多いフレーム数によって構成される論理カットの主題名を求めよ」という検索を考える。この検索は図 4.12 に示す検索式によって表される。

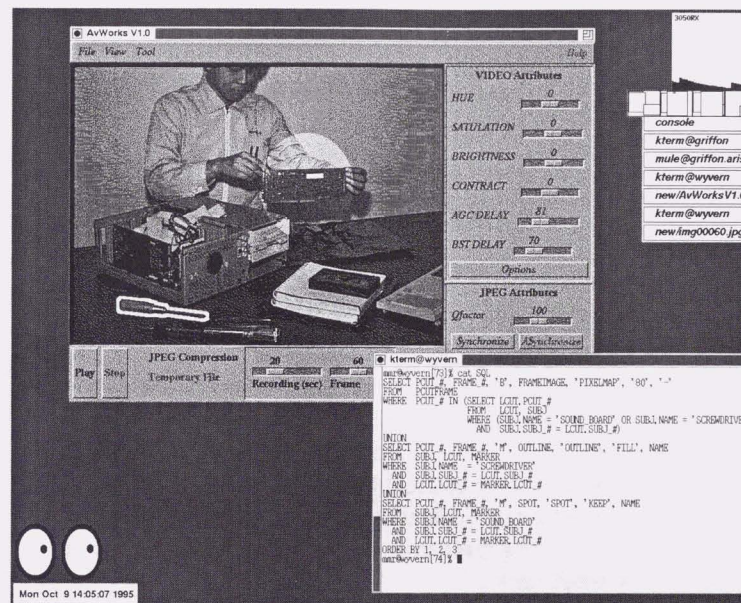


図 4.10: 検索結果の映像の例

この検索は、「物理カット毎」という条件が設けられ、物理カットというグループ毎に同じ演算を行なう操作である。このような検索操作は、集合演算による操作しか行なうことのできない SQL では実現し得なかった操作である。このように MMQL は、操作対象を階層化し、階層毎にグループ化してグループのすべての要素に対し、同じ演算を行なうことが可能である。

以上の通り、従来オブジェクト指向データベースや関係データベースと SQL では困難であった検索操作でも、AIS モデルによる平坦な構造を持ったデータベースと、そこから自由に複合オブジェクトを構築できる検索言語 MMQL によって実現できることが示された。

4.5 映像データベースの試作と評価

MMQL によって映像データベースから検索結果のオブジェクト式を得て、オブジェクト式を映像として再生しユーザーが見られるまでの実行環境は、図 4.13 によって示される。

```

FRAME : apply(PxmFrameimageFrame)
  oα < /FRAME > MARKER : apply(FrameMarker)
  oα < /FRAME > NUMBER : COUNT(MARKER)
  oγ < /FRAME >

```

図 4.11: フレーム中の主題数を求める検索式

```

PCUT : generate(PCUT)
  oα < /PCUT > LCUT : apply(PcutLcut)
  oα < /PCUT/LCUT > [MARKER : apply(LcutMarker),
    SUBJ : apply(LcutRepresentSubjNameStr)]
  oα < /PCUT/LCUT > NUMBER : COUNT(MARKER)
  oγ < /PCUT/LCUT/MARKER >
  oα < /PCUT > AVE : AVE(LCUT/NUMBER)
  oβ < /PCUT/LCUT > (NUMBER > rank(../AVE))
  oγ < /PCUT/LCUT/NUMBER >
  oγ < /PCUT/AVE >
  oδ < /PCUT/LCUT/SUBJ >

```

図 4.12: グループ毎の平均値問題

まず、AIS データベースについて述べる。

AIS モデルを基としたマルチメディアデータベース (MMDB) において、映像データはピクセルマップというバイナリデータで構成するフレームが時間的に並んだ長大な列データであると考えられることができる。このように、一般にマルチメディアデータはそのメディア特有のデータ構造を持っていると考えることができる。したがって検索操作の高効率化を考えると、このようなデータは単純な 2 項関係だけによってデータベースに格納されるよりも、構造を保持した形で格納されることが望ましい。そこで、このような長大なバイナリデータを高速に格納・アクセスすることのできるデータストレージ VARS (Video and Audio Recording Storage) を設け、マルチメディアデータは VARS に格納することとする。

一方、あらゆるデータにおける意味的・論理的情報は、繰り返し述べているように平坦な構造を持つことが望ましく、データベース中においてこのようなデータを格納する部分を特に意味構造部 (Semantic Part) と呼ぶ。

VARS は高速アクセスのための一つの物理構造を持っているに過ぎず、ユーザー側から

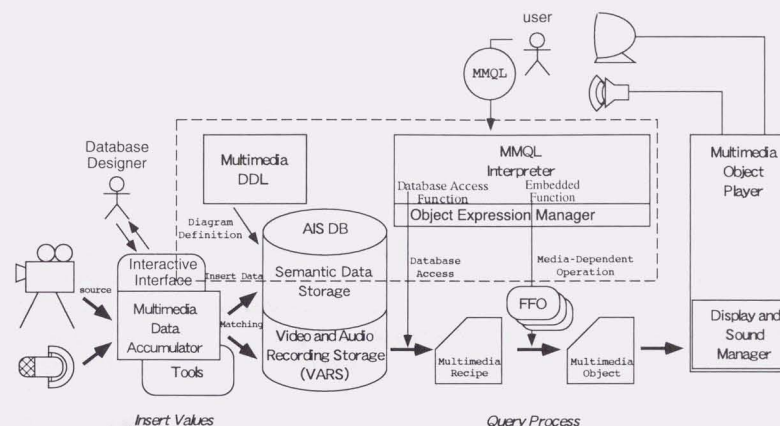


図 4.13: マルチメディアデータベースシステムアーキテクチャ

AIS データベース管理システム (AIS-DBMS) を通してデータベースを見ると、意味構造部も VARS のマルチメディアデータも、同様の 2 項関係による平坦な構造として見る事ができる。

次に、MMQL による検索について述べる。

MMDB の検索操作とは、MMQL によってオブジェクト式を生成し、マルチメディアオブジェクトとして結果を得ることである。MMQL は MMQL インタプリタ (MMQLI) によって実行される。MMQL では、大きく分けて 2 つのステップが考えられる。一つは、データベースにアクセスしてデータベースに存在するデータを構造化するステップであり、もう一つは、そのデータを用いてメディア依存処理を実行し、ターゲットとなるデータを生成するステップである。

前者のステップの結果として、図 4.6 のようなオブジェクトが得られる。この式には、強調のパラメータや検索対象の名前など、ターゲットとなる映像を生成するための情報がすべて含まれている。この意味で、この式をマルチメディアレシピ (Multimedia Recipe) と呼ぶ。

後者のステップでは、レシピからターゲットとなる映像を、埋め込み関数である FFO (Frame by Frame Operator) によって生成し、最終的にマルチメディアオブジェクトをユーザーは得ることができる。

以上の過程において、オブジェクト式はオブジェクト式フィールドと呼ばれるオブジェクト式の操作作業空間において、オブジェクト式マネージャ (Object Expression Manager: OEM) の ADT オペレータによって成長・剪定される。したがって、MMQLI によって MMQL

の評価、実行、最適化処理、並列処理管理及び評価結果のオブジェクト式 ADT オペレータへの変換を行なう。オブジェクト式はオブジェクト式フィールド上で局部的に演算されるため、従来の関数型言語による処理とはことなり、莫大なストリーム処理を回避することが可能である。また、FFO をはじめとする埋め込み関数として、ユーザーによって定義されたメディア依存処理 (Media Dependent Operator) は、並列実行可能な一つのツールとして存在し、OEM を通してオブジェクト式に対して演算処理を行なう。

MMQL によって検索された映像データは、オブジェクト式の形式でオブジェクト式フィールドに存在するが、OEM によってマルチメディアオブジェクトとして再構造化され、ユーザーに検索結果として返されることとなる。ここで、マルチメディアオブジェクトを再生するために、マルチメディアオブジェクト再生機 (Multimedia Object Player: MMOP) をシステムに導入する。MMOP は、映像や音声を表示・再生する機能を持つ端末などによって構成される。また、マルチメディアオブジェクトのヘッダ情報から、複数トラック間の時間軸あわせといった同期管理や圧縮情報の展開などの管理を行なうために、MMOP のコンポーネントとして Display and Sound Manager を持たせる。ユーザーは、マルチメディアデータベースに対して検索操作を行なう際には、MMQL による検索式を記述し得られたマルチメディアオブジェクトを MMOP によって生成することによって検索された情報を得ることができる。

4.5.1 データベース構成システムアーキテクチャ

マルチメディアデータベースに新たに情報を蓄積する際にも、映像に映り込んだ被写体を認識して論理カットやマーカ情報を生成するなど、メディア特有の操作や意味情報の抽出に代表される、従来のデータベースとはことなるデータ入力方法が必要となる。

本稿で述べた映像データベースにおいては、新たに蓄積・登録されるデータはフレーム列だけでは不十分で、フレーム情報から論理カットやマスキ情報の抽出、抽出された論理カットの、主題情報との関係など、様々な情報をデータベース設計者は蓄積しなければならない。

そこで、このようにマルチメディアデータ特有の原情報の登録、原情報からの主題情報の抽出・登録の支援をするシステムコンポーネントを導入し、このコンポーネントをマルチメディアデータアキュムレータ (Multimedia Data Accumulator: MMDA) と呼ぶ。

MMDA は、被写体追跡プログラムなどのツールを利用し、データベース設計者と対話型インターフェイスによって主題情報抽出の支援を行なう。データベース設計者は、データベースに対して MMQL による検索によって現在のデータベースの状態を参照・把握し、新たに作成されたデータを既にデータベースに存在するデータに追加する形で、登録・蓄積することができる。

また MMDA によって、構造を持つ長大なマルチメディアデータについては VARS に、それ以外の意味情報については意味構造部にデータを分配し、それらの間の整合性を保つようにデータベースに登録するよう、要求を発行する。

この結果、図 4.3 に示す通り、マルチメディアデータや意味的な主体、それにそれらの間のつながりを統合的に表現することができ、AIS モデルに基づくマルチメディアデータ

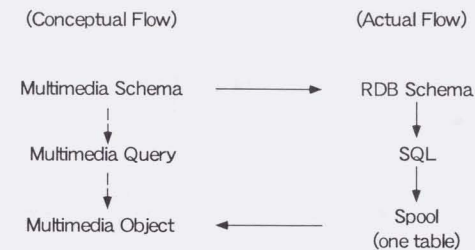


図 4.14: 試作システム実装の概念

ベースを構築することが可能となる。

4.5.2 映像データベース試作システム

本章で述べた映像データベースシステムの有効性を評価するために、試作システムを製作した。本論文では、試作システムの構成については概略だけを以下に述べることとし、詳細に渡る説明については述べないこととする。

この試作システムでは、AIS モデルによるデータベースと、MMQL による検索操作を実装するには至っていない。したがって、概念的には図 4.14 に示すように、中核となるデータベースエンジンに関係データベースを用い、検索は SQL によって行なわれる。したがって、図 4.13 の点線部を関係データベースを用いて実現し、それ以外の部分については、UNIX ワークステーション上において作成されている。これにより、映像データベースの検索は SQL を用いることによって可能である。ただし、SQL による検索で得られるのは、表形式によって表されたマルチメディアレシビである。すなわちここでのマルチメディアレシビは、関係データベースの検索結果のテーブルとして得られ、ファイルの形としてユーザーに渡されることとなる。したがって、レシビを解釈器によって解析し、FFO を用いることによってマルチメディアオブジェクトを得られる。

試作システムの構成を図 4.15 に示す。

試作システムでは、論理カットや物理カットなどの、主体とそのつながり情報を関係データベースで管理し、フレーム列やマーカなどの、いわゆるマルチメディアデータは VARS に保持されている。

ユーザーは SQL によって、検索した映像を生成するためのレシビを作成する。レシビにはフレームごとに、VARS のフレームアドレスやマーカのアドレスが書かれている。レシビを画像生成解釈器 (Image Sequence Generator: ISG) に与えることにより、ISG がレシビを解釈して図 4.7 に示すようなフレーム生成のための構造を生成し、それぞれのフレームについて FFO を用いて目的となる画像を生成している。この結果、マルチメディアオブジェクトが生成され、図 4.10 に示すようにユーザーは検索された映像を見ることが可能となる。

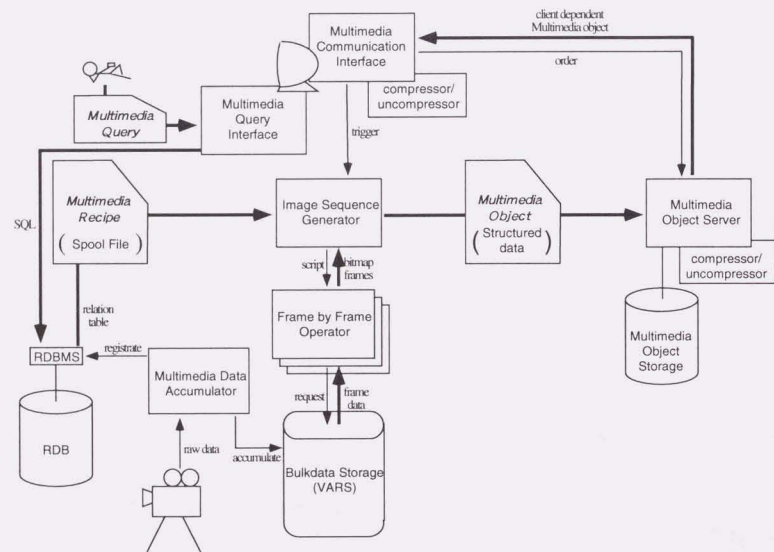


図 4.15: 試作システムの構成

以上が、試作システムの概略である。試作システムにおいては、AIS モデルによるデータベースや、MMQL による検索などは実現されていないため、順序の表現や階層構造をなした複合オブジェクトの生成など、本稿で述べたモデルとしての本質的な構成と統合された検索操作系によるマルチメディア検索には至っていない。

以上で述べた検索操作は、フレームごとのレシピをデータベースから生成し、FFO に代表されるメディア依存処理を埋め込み関数として定義することにより、レシピから目的となるフレームを得る操作である。MMQL の検索の一部をこのようにとらえれば、この試作システムによってその有効性が実証されている。すなわち、主題を指定し、任意の組合せでその主題を強調するという検索が実現され、マルチメディアデータベースの興味深い一面についてその有効性が示されたといえる。

4.6 まとめ

本章では、データモデル論的立場から、エンジニアリングアプリケーションを念頭においた、映像データベースの構築を行い、MMQL によって映像データベースを妥当に操作で

きることについて述べた。

映像データは、そこに映り込んだ被写体やその動き、背景などの様々な視覚情報を含んでいる。これらの情報をデータベース化し利用するためには、映像データをどのようにとらえるかという、映像モデリングの議論が不可欠である。特に、映像の内容に基づいた検索を実現するためには、被写体情報を抽出する技術と、抽出されたデータを基に映像にインデックスづけを行なうモデリングが必要となる。

本研究では、物理カットと論理カットという映像単位と、被写体形状を示すマーカの概念によって、内容検索を実現するデータベースの構築を試みた。これらの概念によって、映像データの持つ視覚情報をデータベース化することを実現できた。

また、MMQL によって、被写体の名称などを基に、平坦に構築されたデータベースから自由に階層構造を生成することが可能であることを示した。得られた検索結果は、データベース検索によって新たに生成された映像である。これは、MMQL に FFO などのメディア依存オペレータを組み込むことで実現できた。このことから MMQL が、マルチメディアデータベース操作言語として有効な言語であることを示した。

Chapter 5

実世界データベースの構成と実現

本章では、多視点映像と物体の3次元形状データを基にした実世界データベースを実現するために、実世界に登場する物体の形状に、意味を付加して簡略化する新しいモデル化手法を提案する。対象世界のシーンに出現する、物体の形状や動きをデータベース化するためには、形状とその意味を少ないデータ量で効率良く表現することと、質問処理の実現性の確立が必須である。そこで本章では、個別モデル形状と呼ばれる、個々の物体固有の形状と意味を反映した近似ポリゴンモデルを用いることによって、シーンのデータベース化を試みた。このデータベースは、個別モデル形状とその3次元動的な動きをシーンから抽出して、仮想的にCG空間にマッピングすることで構築される。また、実際に作成した個別モデル形状の誤差について評価し、空間オペレータに基づく質問処理の実現性の評価を行なった。

5.1 視覚情報を基にした実世界のデータベース化と時空間データベース

近年の計算機技術の発展により、グラフィック画像やビデオ映像などのマルチメディア情報を容易に取り扱うことが可能となってきた。特に、高速なグラフィックコンピュータの登場とマルチメディア技術を基に、実世界の情報を計算機上で仮想的に再現して、現実感豊かに再生し体感することのできる仮想現実 (VR:Virtual Reality) システムの研究が進められ、注目を集めている [HIRO97, KANA97]。VR がリアルな仮想空間を構築することに主眼をおいているのに対し、現実世界の時間的・空間的な情報をどのように計算機上にモデル化し、蓄積・管理・検索するかという提案が盛んになされている [KURO97][OREN88]。このような、現実が生じた、人や物体の表面形状やふるまいをそのまま蓄積し、それらの位置や動きなどを検索することのできるデータベースを時空間データベースと呼ぶ。

時空間データベースを構築する上での問題点として、3D情報の獲得方法や分析方法、時間やふるまいのモデル化手法、基盤となるデータモデルと操作言語に求められる機能などが従来のデータベースよりも高度化していることがあげられ、結局データベースのすべての要素を新たに見直す必要が生じる。3次元空間や時間のモデル化については、3D-CAD

システムや地図情報データベースシステム [EGEN94] など、分野ごとにいくつかの試みが見られる。しかし、実世界をデータベースに取得するための手段やモデリング手法、得られたデータの操作手法など全般に渡る確立した手法は存在しない。

また逆に、時空間データベースシステムが実現すれば、様々な分野に応用することが可能である。例えば我々が注目しているエンジニアリング分野においては、CAD/CAM データやグラフィックシミュレーションによる仮想空間、ソリッドモデルやサーフェスモデルなどによる物体形状データ、実際にカメラを用いて撮影されたサンプル映像など、多様なデータが生じる。そこでは、熟練作業者とロボットとの協調作業などの工程や作業に関する知識情報の蓄積や、規定動作と実際に行なった作業の比較・検討といった高度なデータ利用が求められている。時空間データベースによってこれらのデータを同一の枠組の中で取り扱うことにより、実際のデータの管理・検索だけでなく、そのデータから工程や作業を仮想的に構築して事前に評価を行なうことも可能となる。その他にもスポーツ分野においては、道具や選手の動きや戦術などの蓄積に、さらに、医療や芸術など、実空間の事象を対象とした多種多様な分野への応用を考慮することができる。

このように実際の応用を考えると、現実世界の情報をデータベースに蓄積するためにはまず、現実が生じたありのままの事物や事象をデータとして取得しなければならない。したがって時空間データベースの構築は、近似的な時空間のモデル化手法を提案するだけでは不十分である。例えば、実世界を計算機上に取り込むためには、多視点のビデオカメラやセンサなどを用いて時系列データとして取得したり、物体の3次元形状を測定するセンサによってスキャンしたデータを対象とするなどの手段が考えられる。これらの視覚情報は、例えば多視点映像についていえば、同期した複数の映像ととらえることができる。よって、前章までの映像に基づくモデルの延長上にある。したがって、これまでの議論と同様に、新たなモデル化手法と操作方法について考え直さなければならない。

本章では、時空間データベースを実現するために、多視点映像や3次元形状データを基にして、時空間をモデル化するためのシーンやカットなどの基本となる概念を導入し、時空間のモデル化手法について考察する。本研究では時空間のモデル化手法として、コンピュータグラフィック(CG)による仮想空間の構築を試みた。高速なグラフィックコンピュータの出現によって、CGによる視覚的な仮想空間の高精度な計算を現実的な時間で実現することが可能である。したがって、ここでは空間に存在する「物体」と時空間の領域との関係を基に「物体」の位置や動きを問い合わせる質問を、CGにおける位置関係計算に帰着させて実現する手法を提案する。

他の時空間データベースに関する提案として、物体を外接矩形(MBB:Minimum Bounding Box)で近似する手法が提案されている [ISHI97, MASU97, THEO96]。MBBはシンプルのため高速計算が可能だが、物体の細部までモデル化されておらず、正確な位置計算を実現するためには、近似物体の細分化や多重の座標変換などを要し、結局大きな計算コストを要求することになる。また、近似された空間そのものを視覚化したとしても、現実感を得ることは困難である。一方本手法では、データベースのモデルそのものにCGによる仮想空間を用いることにより、実データに対する単なる物体検索だけでなく、さらにVRの現実感をそのままデータベースでも利用できるという利点を持つ。

本研究では、マルチメディアデータベースの応用分野として、より高度で高精細な映像

をいろいろな形式で組み合わせるという点でエンジニアリングの分野に注目している。エンジニアリング分野においては、CAD/CAM データ、グラフィックシミュレータを用いた高度なシミュレーション映像、ソリッドモデルや、サーフェスモデルなどの立体モデル、実際にカメラなどを用いて撮影されたサンプル映像など、様々な形態のデータが生じる。このように現実世界(リアルワールド)に生じた多様なデータをできるだけ忠実に取り込み、さらに解析、モデル化したデータも同一の枠組の中に蓄積し、検索操作をすることのできるデータベースシステムとして、リアルワールドデータベースシステム [ARIS96] がすでに提案されている。

リアルワールドデータベースでは、カメラやセンサなどを用いてそっくりそのまま取り込んだデータと、時空間に存在する“もの”に関するデータの2種類のデータを主として蓄積することとなる。前者は、2台ないし複数台の高精度カメラを用いて対象世界を撮影して得られた映像データなどである。また、後者の“もの”に関するデータは、映像に対して被写体追跡プログラムを用いて解析した結果や、左右映像から算出される物体の空間における位置、物体の3次元形状を測定するセンサによって得られるグラフィックによる外形情報(サーフェスモデル)などを考えることができる。また、人体の構成などは、人間が古くから持っている知識であり、このような情報もまた、スキーマとしてリアルワールドデータベースに組み込まれ、そのインスタンスとして、人体の構成部品をグラフィックとして蓄積するという提案がなされている [KOBA96]。

このように、リアルワールドデータベースでは、“もの”の情報はグラフィックとして蓄積されるため、時空間質問の実現には次のような利点が考えられる。第一に、取り込まれた左右映像に対し、検索操作のたびに位置を算出するのではなく、グラフィックの位置を考慮することによって検索することができる。もちろんこの方が、単純な計算で実現することができる。第二に、グラフィックの粗さをデータベース設計者が決めることによって、高精細な位置情報に基づく検索をするのか、粗い検索をするのかのコントロールをすることができる。オブジェクトの外接矩形による位置計算の方法が提案されている [MASU97] が、本提案の粗いグラフィックのサーフェスモデルを用いることにより、外接矩形法なみに単純な方法で外接矩形法よりも細かい位置関係の導出が可能となる。

さらに、時空間ではものの位置は時間的に変化する(動く)が、その動きに対する質問として、「ある平面を通るかどうか」という位置関係は非常に興味深い。また、平面を考えると、対象空間を3次元空間(立体)から2次元空間(平面)へと制限することとなる。3次元空間中のたくさんのものの中から、ある指定された面を通るものだけに制限することにより、時空間質問の高効率化にもつながる。

5.2 mediator を用いた実世界モデリング

5.2.1 シーンの構成要素

現実世界の情報をデータベース化するためには、まず対象とする世界がどのような時空間的情報で構成され、どのようなデータによってその情報が表現されるのかについて考え

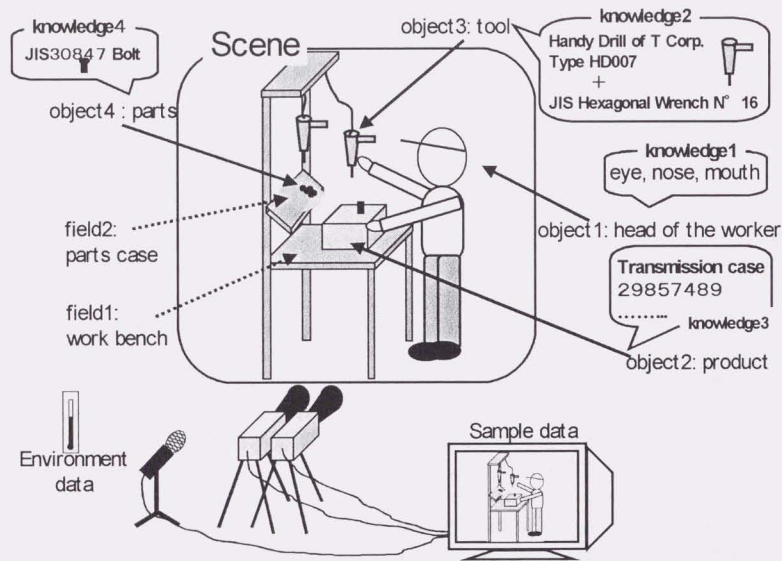


図 5.1: シーン概念

る必要がある。

ここでは、データベース化する対象範囲の世界（時間的情報を含む）を単位としてシーン (Scene) と呼ぶ。図 5.1に示す通り、シーンの情報を次のレベルで扱うことにする。

- シーン全体の情報 データベース化する対象世界の時間的・空間的な広がり（範囲）や座標系の基準位置などの、グローバル情報
- シーンを構成する場の情報 作業台やフィールドなどの、シーンに参加する物体が移動や運動を行なうための場の情報
- シーンに参加する物体の情報 作業員やロボット、道具やボールなどの、シーン内に存在する注目している物体の、形状や動きに関する情報
- ある種の物体に関する一般的な情報 人間であれば目鼻口がおおよそ頭部のどの位置についているかなどの、物体の個性差を除いた一般的な知識情報
- シーンを外界から取得したサンプルデータ

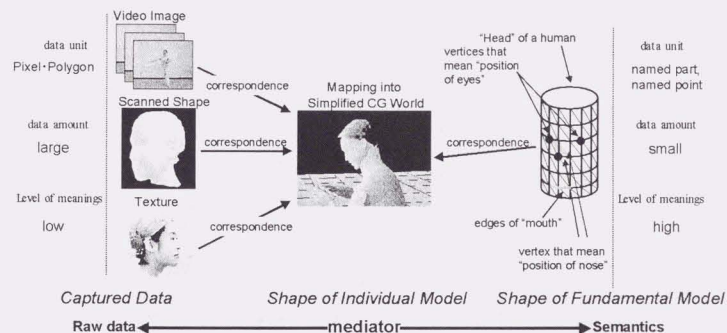


図 5.2: シーンデータの取得

ビデオカメラや3次元スキャナ、圧力計や温度計など、各種センサを用いて測定したサンプルデータ。もちろん、サンプルデータそのものには、シーン内に存在する物体の動きや意味などの情報は何も付加されていない。

現実生じた時空間情報をデータベース化するためには、以上のような各種、各レベルの情報をすべてモデル化し、蓄積する必要がある。

VRや3Dゲームでは、高精細なポリゴンデータを用いて視覚性を重視した表現がなされているが、複雑な形状を単純にポリゴン群だけで表現する方法は系統的なものではなく、大量のシーンの中から物体同士の位置関係などをキーとして検索操作を行なうことを考えると、計算量的にかなりの無理があり現実的ではない。さらに、個人間で「鼻」の高さを比較する」とか、「鼻」の体積の平均値を求める」など、形状の意味を踏まえた検索や集約演算まで実現するのは困難である。

また時空間データベース設計に関する提案として、物体を外接矩形 (MBB: Minimum Bounding Box) で近似して表現したり [ISHI97, MASU97]、MBBを基にしたデータ操作を行なう手法 [THEO96]、物体の空間的な位置関係をアイコンにマップさせて簡略化し [DELB93]、それらの操作を時空間論理を用いて実現する手法 [DELB95] などの研究がある。MBBはモデリング手法としてもデータ操作手法としても、Allenによる時区間論理 [ALLES3] を空間領域に拡張して適用することができるため、限定されたアプリケーションにおいてはデータ操作の高速化が可能である。しかし、物体の細部までモデル化されておらず、いずれにおいても実際の形状に基づいた正確な表現や位置計算を実現するためには、近似物体の細分化や多重の座標変換などを要し、結局大きな計算コストを要求することになる。また、近似された空間そのものを視覚化したとしても、現実感を得ることは困難である。

シーンを有効に蓄積することができ、同時にデータ操作の際に計算量的にも視覚性にも

優れた時空間データベースを構築するためには、新しいモデル化手法と検索のためのデータ操作手法の二つの観点から、シーンデータベースを構築していかなければならない。

5.2.2 シーンデータベースの構築

モデリング手法

本研究では新しい試みとして、図 5.2 に示すように、物体の意味に基づいた簡略モデルを導入し、サンプルデータへのフィッティングを半自動的に行なうことで、サンプルデータと意味との間に対応づける mediator (仲介者) を生成する。この mediator は、シーン内の個々の物体の形状を精度良く少ないポリゴン数で表現する。これらを仮想 CG 空間マップすることによってシーンデータベースの構築を試みた。

実世界から取り込まれる、または与えられるデータにはいろいろな種類、レベルのものが考えられる。本研究では、対象世界の物体の形状とその時間変化を蓄積することを目的とする。特に人間の動作を解析し、データベースへの蓄積対象と考えた場合 [KOBA96] には、その動作に影響を与えるような装置や環境では、正しい動きの情報が得られない可能性がある。作業や動作情報の広範に渡る取得という観点から、特殊な装置を必要としない撮影環境においてデータを取り込むことを前提としたい。そのためここでは、シーンを外景から取得するサンプルデータは複数台のビデオカメラによる立体映像と、非接触型の 3 次元スキャナによる物体の正確な外部形状とテクスチャデータのみとした。

これらのサンプルデータには当然、例えば映像のフレームであればどのピクセル領域が対象物体なのか、形状データならどのポリゴン群が人間の鼻を表しているのかなど、「意味情報」は含まれていない。物体についての情報は、人間がその物体に対して持っている知識や解釈にあたる「意味」を組み合わせることによってはじめて記述することができる。しかし、ピクセルやポリゴンで表されたデータはデータ量が多過ぎていちいち人間がそこに「意味」を手動で指定していくことはあまり現実的ではない。そこで、この「意味を付加する」過程を半自動的に実現でき、かつそのままデータベースに蓄積し、検索するために、ここでは物体の形状に対して次の 3 レベルの概念を導入することを提案する。

- **基本モデル形状 (Shape of Fundamental Model)** 目、鼻、口、耳を持つ「人間の頭部」や、金属部と柄の部分からなる工具「ハンマー」など、一般的に知識として、その大まかな形状と、部品内の特徴的な部分を少量のポリゴンデータによって表現できる形状情報を基本モデル形状と呼ぶ。例えば本研究では、人間の頭部の場合「9 層の断面を持つ 12 角柱」で基本モデル形状を構成し、下から 2 層目の断面は下顎の位置、上から 2 層目の断面は目の高さの位置といった情報を付加することとした。また、それぞれの層の円周方向の頂点についても、右目や左目の位置を表す頂点や、口の位置を表す辺といった情報を付加することも可能である (図 5.2)。このように、ある部品共通の情報を付加することによって、個々の部品固有のデータを基本モデル形状の「意味」によって統合化することができる。
- **スキャン形状 (Scanned Shape)** ポリゴンで表された対象物体の形状データとして、

3 次元スキャナによって対象物体固有の精密な形状データを得ることができる。これをスキャン形状と呼ぶ。スキャン形状のポリゴン削減処理やテクスチャの貼り付けを行えば、任意の精度のグラフィックモデルの作成ができるが、スキャン形状はサンプルデータの一つであり、そのデータは一切の意味情報を持っていない。また、個体データ間にも統一性がないため、「意味」を付加しても比較や集約演算の対象としにくく、結局有効ではない。したがって、意味に基づくデータ表現や操作を要する際には、次に述べる個別モデル形状を作成する必要がある。

- **個別モデル形状 (Shape of Individual Model)** 基本モデル形状の持つ「意味」を反映しつつ、スキャン形状から近似して得られるポリゴンデータを個別モデル形状と呼ぶ。すなわち、個別モデル形状とは、実世界から直接取得したサンプルデータと、データベース構築のためのモデルとの間の対応づけを行なう mediator (仲介者) である。個別モデル形状は、個々の対象物体の形状を持ちつつ、少量のポリゴンで形成されるため、仮想 CG 空間を個別モデル形状によって作成し、シーンデータベースを構築する。個別モデル形状は、個体データ間で「意味」の統一をはかることができるため、検索や比較評価を行なう上で極めて有効となる。

個別モデル形状の取得アルゴリズムを人間の頭部を例にとり説明する (図 5.3)。取得の際に用いられる基本モデル形状は、すでにデータベース作成者によって作成されているとする。例として、ここで用いる基本モデル形状は、前述した「9 層の断面を持つ 12 角柱」であるとし、層ごとに「目」や「鼻」などの意味が付加されていると仮定する。

まずスキャン形状に対し、中心軸を設定する。次に「目」や「鼻」といった、基本モデル形状で意味づけされた部位の位置で、軸に対して垂直な面でスキャン形状を切断し、この断面を求める (図 5.3(a))。この断面に対して、基本モデル形状の該当する層 (図 5.3(b)) を軸を合わせて当てはめる (図 5.3(c))。軸 (中心) と基本モデル形状の各頂点とを結ぶ直線と、スキャン形状切断面の外郭線との、交点を求める。得られた交点を順に結んでできる平面を、個別モデル形状の層とする (図 5.3(d))。各層についても同様の処理を行ない、最後にすべての層を結び付けて個別モデル形状とする。

上記の取得方法は、計算機によって自動的に処理することが可能である。ただし、基本モデル形状の「意味」に該当する部位をスキャン形状から自動的に算出することは容易ではない。例えば、「目の高さ」の層をスキャン形状から自動的に抽出することは、「目の高さ」に関する知識を自動算出プログラムに与えなければならない。しかし本研究の目的は、近似形状取得の完全自動化の実現ではなく、グラフィックに基づく mediator の設計と、mediator による質問処理の高効率化である。このため本稿の段階では、各層の対応づけについては自動化を行っていない。それに代わり、支援ツールを開発して、手動による層の指定を行なっている。

なお、少量のポリゴンによるモデル化のアルゴリズムとしては、スキャン形状の頂点からいくつかの点をランダムに選択し、選択された頂点群だけでポリゴンを形成し直すサンプリング手法など、別の方法も考えられる。本研究では、基本モデル形状の持つ「意味」を個別モデル形状に反映させる立場をとるため、基本モデル形状との関連づけを行なえるアルゴリズムを重視し、ランダムなサンプリング手法はあえてとらなかった。

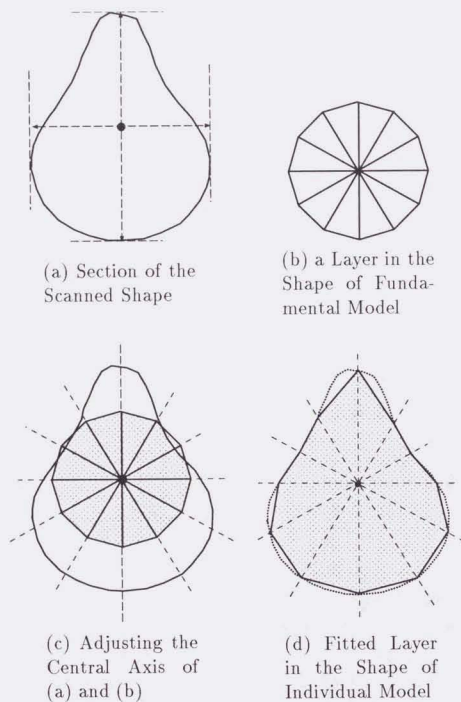


図 5.3: 個別モデル形状の取得

データ取得の流れ

ある人間の運動の様子をシーンデータベースに蓄積する場合を例にとり、シーン内の物体形状データを取得する流れを図 5.4 に示す。サンプルデータとしては、頭部のスキャン形状と複数台のビデオカメラで撮影した映像（フレーム列）とする。まず、人間の頭部の基本モデル形状をデータベース作成者が作成したとする。次にスキャン形状に対し、個別モデル形状取得支援ツールを用いて目の高さや顎の高さなどのデータを手で入力し、スキャン形状にフィットするように基本モデル形状を変形していく。この結果、基本モデル形状と同じポリゴン数で表現された個別モデル形状を得ることができる。

また、カメラ画像からの姿勢抽出や奥行き情報抽出を行なう際、個別モデル形状を用いることによって、高速化、高精度化の実現が可能であろう。個別モデル形状を仮想空間上で移動・回転させ、ある時点のカメラ画像（フレーム）と同一の面に投影した画像と比較

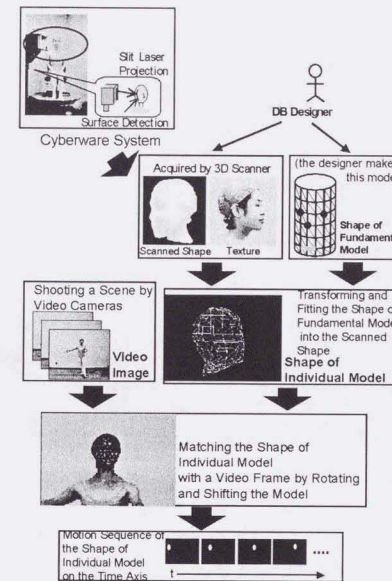


図 5.4: データ取得の流れ

し、マッチングをはかることによって、その時点での物体の姿勢を求めることができる。そこで個別モデル形状を実画像にマッチさせる技術は重要な役割を持つが、我々は目などの特徴的な色領域を連続するフレーム内で追跡する“Chaser”アルゴリズムを開発しており、システム構築の技術的な見通しが立っていると考えている。

これらの一連の流れの中で、精度面などから見て開発途上のものもあり、トータルシステムを構築して評価を行なうことは次なる課題である。しかし、システム実現のための技術的な根拠として、画像からの人間の姿勢情報の抽出に関する研究がいろいろなところでなわれている。例えば、人体の簡略化された CAD モデルを画像にフィットさせるアプローチ [YAMA96] や円筒モデルを用いた近似方式 [BROOS3] などがある。これらの方法は、直方体や円筒などに簡略化されたモデルを画像にマッチさせる手段をとる。本手法で作成された個別モデル形状を mediator として用いれば、目や口の位置（ポリゴン領域）に該当するテクスチャデータから、目や口の色データなどを取得することができる。このため、これらのデータを基に画像処理を行えば、簡略化されたモデルよりも高い精度で画像からの空間情報が可能になると考えられる。

また、複数台のカメラ映像から、物体の位置や形状を認識する技術としては、カメラ画

像の任意視点への変換 [SATO96] や、距離画像からの3次元物体モデルの導出 [HIGU96]、ステレオ画像からの物体認識 [SUMI98] などがあり、画像と個別モデル形状のマッチをとる際に、技術的に応用が可能である。

各時点ごとに個別モデル形状の仮想空間上での位置を算出することはすなわち、姿勢の時系列を算出することによって運動をモデル化することにつながる。このデータを、元の情報と合わせてデータベースに蓄積すれば、仮想CG空間を基にしたシーンデータベースを構築することができる。加えて、個別モデル形状による手法は、個別の物体同士の位置計算を仮想空間上で簡略化して行なえるため、データベースにおける質問処理の際に有効である。質問処理に関する評価は4章で行なう。

個別モデル形状の取得と評価

以上の議論を基に、個別モデル形状取得システムを試作し、評価を行なった。

1. スキャン形状の取得

スキャン形状として、サイバーウェア社製非接触型3次元スキャナ (COLOR 3D DIGITIZER 3030RGB/PS/LN, 図5.4) によって人間の頭部のデータを取得した。この3次元スキャナでは、高さ30cm、直径55cmの円筒状の空間をスキャン対象領域とする。対象物に赤色の垂直スリットレーザー光を照射し、別の角度から観察したレーザー光の曲線を全円周方向から組み合わせることで対象物の形状を作成する。得られる原始的データは約35万個の三角形ポリゴンから構成され、これを幾何的アルゴリズムによってポリゴンの統合化を行ない、表面形状を取得する。

この結果、図5.5に示したポリゴンデータを得られた。この形状は、頂点数888個、1814枚の三角形ポリゴンによって構成されている。

2. 基本モデル形状の構成

人間の頭部の基本モデル形状として、最も単純なケースとして図5.2に示すような正十二角柱を用いた。この角柱は9面の断面層を持ち、それぞれの層の意味情報として、再下層は首の下部、3層目が顎の先端、4層目が口、5層目が鼻の頂点、6層目が目、最上層が頭頂部とし、2、7、8層は、その上下の層を等分する高さであると仮定した。この形状は、頂点数108個、216枚の三角形ポリゴンによって構成されている。

3. 個別モデル形状の生成

3.1節で示したアルゴリズムによって、図5.5のスキャン形状に基本モデル形状のフィッティングを試みた。ここでは、まず基本モデル形状の各断面層をスキャン形状の該当する部位に支援ツールを用いて手動で指定して合わせた。次にその部位でのスキャン形状の断面が一致するように、基本モデル形状の断面形状を変形した。変形処理、断面形状からのポリゴン化処理は自動化されている。この結果として図5.6の個別モデル形状を得た。この形状の頂点数、面数は基本モデル形状と同様である。

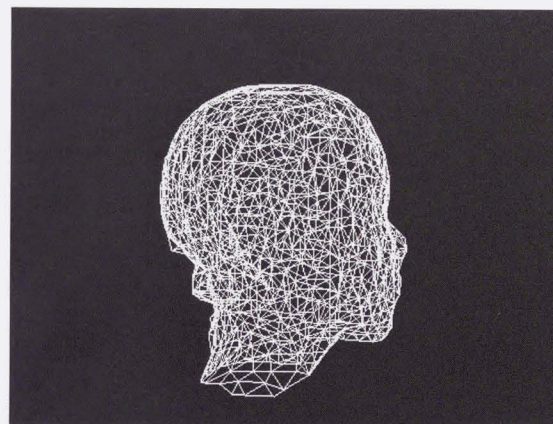


図5.5: スキャン形状

ここで得られた個別モデル形状が、どの程度スキャン形状を反映しているか評価するために、スキャン形状の各頂点が、個別モデル形状の最も距離の近いポリゴンからの程度離れているか、すなわち、形状の誤差を算出した。その結果を図5.7のヒストグラムに示す。また、上部視点からの比較を図5.8に示す。

図5.7の示す通り、ほとんどの個別モデル形状のポリゴンは、オリジナルのスキャン形状の頂点から3mm以内の誤差で生成されている。また、10mm以上の誤差を持つ頂点も見られるが、これらはほとんどがスキャン形状を取得する際の誤測定であり、実際の対象物体には存在しない頂点であった。これは、スキャン形状と個別モデル形状を重ね合わせて視覚化し、拡大して目視することで確認した。また、図5.5と図5.6の比較、および図5.8に見られるように、視覚的にも元の形状を反映している。この結果から、人間の頭部のように、形状が凸物体であるかまたはそれに近いものに関しては、3.1節で述べたアルゴリズムの性質上十分な精度で個別モデル形状を取得できることがわかった。しかし、表面形状に凹凸が激しく現れる物体に関するアルゴリズム作成手法の確立は今後の課題である。

以上の様に、個別モデル形状は mediator として有効であるが、シーン内のすべての物体に対して mediator を作成することは現実的ではない。しかし、本稿で提案した mediator の役割は、データベースにおいて後に検索の対象となる物体について、オリジナルデータ (形状データや映像) にインデックスづけを行なうことである。したがって、mediator によるインデックスづけがなされていない情報については、別の方法でインデックスづけを行なう必要がある。画像内に現れる物体 (オブジェクト) に対するインデックスとしては、その存在と特徴量をデータベースに蓄積するアプローチ [AKAM97] などがある。

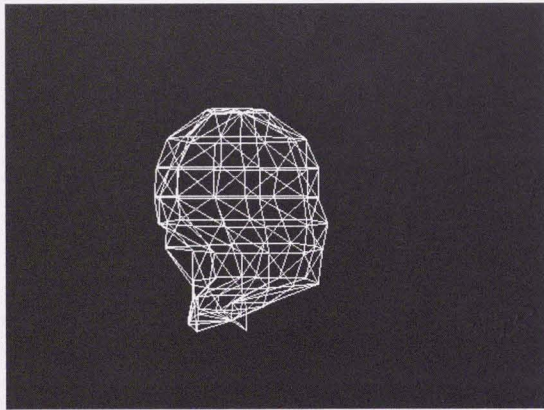


図 5.6: 生成された個別モデル形状

一方 VR などに見られるように、仮想世界と現実世界を融合しようという研究においても、空間内のオブジェクトを抽出して組合せて、新しい世界を合成しようという試みが見られる [TSUK97]。本研究では、オリジナルデータとして映像も蓄積されるため、mediator の作成されていないものについては、(検索時に計算コストを強いるが)そのまま映像を利用してサイバースペースを構築することもでき、この際には同程度の現実感を達成できると考えられる。

5.3 AIS モデルによる実世界データベースの構築

本章では、個別モデル形状とその仮想空間での運動情報を基にしたデータベースを設計し、その上での検索操作について述べる。

5.3.1 シーンデータベースのスキーマ

前章で述べた、個別モデル形状と、その運動の時系列データをデータベースに表現し、蓄積した例を図 5.9 に示す。

今まで述べたようにシーンデータベースにおいては、シーンに参加する物体の情報(基本モデル形状、個別モデル形状)や、場の情報、さらには元となるサンプルデータ(スキャン形状や立体映像)などの基本要素(primitives)があり、かつそれらが相互に対等に参照し合うという構造をとっている。このため、階層構造などの特定の見方のデータモデルよりもフラットな構造を持つデータモデルの方が好都合である。

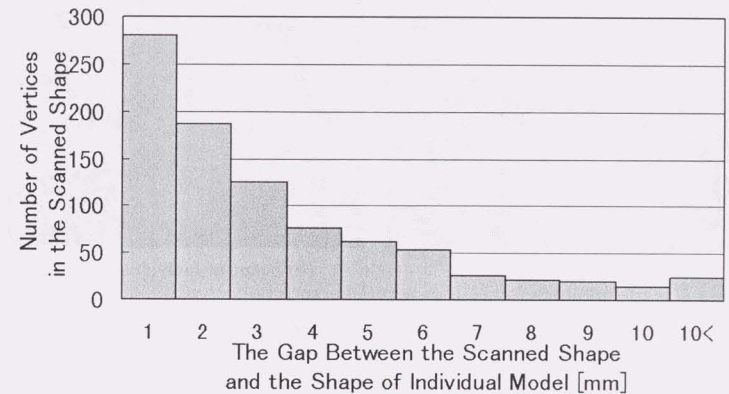


図 5.7: スキャン形状と個別モデル形状の誤差分布

関係モデルは、双方向の参照とフラットな構造を持つといえるが、順序を明示的に蓄積することができず、また、グラフィックオペレータや画像オペレータなどの、シーンデータベースで用いられる特殊なオペレータを操作系に埋め込んで用いるのは困難である。また、オブジェクト指向モデルはグラフィックや画像を取り扱うには向いているが、オブジェクトごとに上下構造を持たさずに双方向の参照やフラットな構造で情報を表現するためには工夫を要し、必ずしも適切であるとはいえない。本稿の目的は、mediator による意味と実データの対応づけをデータモデリングの中核として実現し、その評価を行なうことである。そこで本稿では、シーンデータベース設計のテストケースとして、できる限り汎用的な設計手法で、フラットな構造による表現を実現したいという観点から、エンティティベースのデータモデルによる情報表現を試みた。

図 5.9 のスキーマダイアグラムは、AIS (Associative Information Structure) モデルによって記述されている。AIS モデルは、ER モデル [CHEN76] を基にした、エンティティベースのデータモデルであり、実世界の事物・事象の代理物であるエンティティ(entity)と、それらの間の対応づけ(association)によって情報を表現する。また、3章で述べたように AIS モデルに対する一般的な操作系として MMQL をすでに提案しており、データ記述、データ操作の両面からデータベース設計を行なうことが可能である。図 5.9 において、矩形がエンティティタイプを表し、小円(O)はそのタイプに属するエンティティを表す。また、タイプ同士の対応づけとして、直積の集約のタイプ化(菱形)や、多値属性(二重矢印)などがあり、エンティティ間の実線としてインスタンスの対応が示されている。

図 5.9 のスキーマで表される情報は次のようになる。

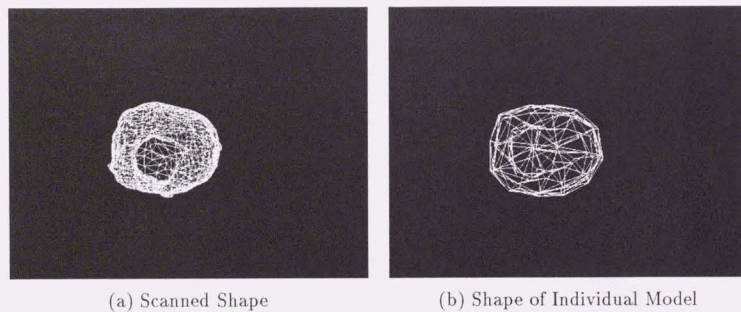


図 5.8: 上部視点からの両形状の比較

1. 取得されたシーン (SCENE) に関する情報として、場を構成する物体 (BACKGROUND.OBJECT) やシーン内の時間 (POINT_OF.TIME)、それにステレオカメラで撮影されたシーンの映像 (FRAME.PAIR, FRAME) らがそのサンプルデータと共に蓄積される。
2. 取得された個別モデル形状 (SHAPE_OF_INDIVIDUAL.MODEL) は、ポリゴン情報も含めてエンティティとして、蓄積される。
3. シーンに参加している物体の動き (MOVEMENT) は、その物体の個別モデル形状がシーンに出現する (OCCURRENCE) ことを表すエンティティとシーン内時間の各時点との直積によって表され、その時点での姿勢として座標値や回転角 (TRANSLATE, ROTATE) と共に蓄積される。

すなわち、ある物体のシーン内での運動に関する情報は、個別モデル形状の位置座標と向きが各シーン毎に時系列データとして示されているため、仮想 CG 空間上に現実のシーンをマップしたシーンデータベースを構築したとすることができる。

またこのデータベースには、シーンの場を構成する背景物体の情報や、ステレオカメラで撮影したビデオ映像などのサンプルデータも同一の枠組によって表現されるため、仮想 CG 空間での検索を基にして、ビデオ映像を検索結果として得るなどの多様な質問に対応することを実現することができる。

以上により、仮想 CG 空間を用いたシーンデータベースの構築を実現できた。

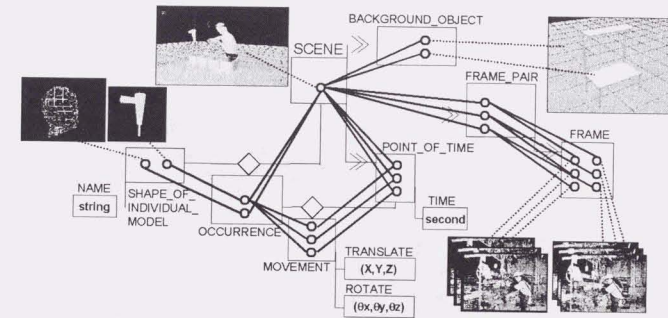


図 5.9: シーンデータベースのスキーマとインスタンス

5.4 時空間質問処理

5.4.1 空間オペレータによる質問処理

実世界データベースに対する質問処理としては、物体間の時空間的關係や、時空間条件に合う「シーン」の検索など多岐に渡る。エンジニアリングアプリケーションを例にとると、時空間的關係として「作業者から 1m 以内にある工具の種類を求めよ」とか、「搬路で運搬される部品にロボットのアームが干渉した事例を示せ」といった質問要求が生じる。これらの質問処理には、ある座標系に基づく特定範囲内の物体を探索したり、物体相互の移動や位置関係を算出するなどの空間オペレータを要する。

これらのタイプの質問処理に対して、基本モデル形状と個別モデル形状の導入の利点は 2 つある。一つは、オリジナルの形状情報を残したままポリゴン数を削減できることである。少ないポリゴンで表現できるということは、物体同士の空間的な位置関係などを求める空間オペレータをそのままのデータ量で実装しても計算時間的に無理なく実現できる。この利点は計算量と精度のトレードオフの問題であるともいえるが、それだけではない。もう一つの利点は、mediator を用いてポリゴンオブジェクトを統一されたモデルによって表現することにより、高効率な空間質問処理アルゴリズムの提供と、平均値や最大値などの集約演算を実現することである。

個別モデル形状の導入は、単に視覚化のためだけに CG を用いるのではなく、シーンのデータベース化と、CG に基づく選択演算や集約演算の実現可能性を示している。そこで本節では、質問処理をポリゴンオブジェクトに基づく空間オペレータによる計算に限定して計算量について考察し、個別モデル形状の有用性について述べる。

空間のある範囲に存在する物体を問うといったタイプの質問は range query と呼ばれ、空間に対する問合せとしては頻繁に行なわれる典型的な質問処理である。ここで本稿では、図 5.10 のように、質問対象の「範囲」を「複数の無限平面によってバウンドされる」範囲と

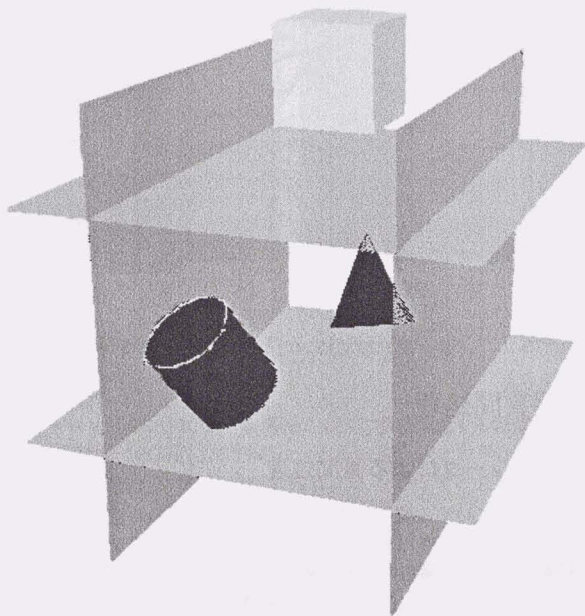


図 5.10: Range Query

とらえ、この無限平面を質問処理の基本要素と考えた。図 5.10は、「空間中のある矩形領域の内側に存在するオブジェクトを選択せよ」という質問を表す。

これには実現性の面から以下の理由がある。ポリゴンオブジェクトと無限平面との関係を算出することは、それぞれの頂点や辺、面が無限平面とどのような位相になっているのかを算出することであるといえ換えられる。無限平面は、法線ベクトルと平面上の一点により単純な方程式だけで定義することができる。また、任意の点と無限平面の位相関係（面上、上方、下方）を四則演算のみの関数で算出することができる。したがって無限平面を基本要素とすることは、複雑な演算を行わずに浮動小数点の四則演算だけで実現が可能である。

そこで本稿では、次の3種類の質問処理演算について考察し、実験・評価を行なった。

1. 物体と任意の無限平面とが交差するかどうかの確認
2. 物体を任意の無限平面で切断した時の断面形状の導出

3. 物体と任意の点との間の最短距離の算出

(1)を解くアルゴリズムは、ポリゴンオブジェクトの任意の2頂点が、無限平面によって分割される空間の両側（すなわち平面の両面側）に存在すれば真である。したがって、計算量は最悪時で頂点数に比例した浮動小数点の四則演算を行なえば良い。

また(2)は、(1)が真であるポリゴンオブジェクトに対し、無限平面と交わる辺の交点を算出すれば良い。このアルゴリズムの計算量も辺数に比例した浮動小数点の四則演算となる。

最後に(3)は、ポリゴンオブジェクトのすべての頂点と任意の点との距離を算出し最小値を求めれば良いので、頂点数に比例した平方根演算である。

これらの演算は総じて、頂点数に比例する計算量である。したがって、頂点数を削減した個別モデル形状を用いることは有効であると考えられる。

また、単に頂点の削減による効率化だけではない。個別モデル形状は軸を持つという仮定から、(3)の計算は次のようなアルゴリズムに改良することができる。まず、距離を求める対象の点と軸とによって生成される無限平面で個別モデル形状を切断し、断面上の頂点を(2)によって求める。個別モデル形状に極端な凹面が存在しない場合には、(2)から求められた頂点の中に、最短距離となる頂点が存在する。よって平方根の計算はこれらの頂点だけについて行なえば良い。

以上の考察から、時空間質問を評価することは、仮想オブジェクトと平面などの幾何図形との関係を計算することに帰着される。そこで、時空間質問を以下のように考える。

時空間質問

時空間質問とは、シーンのオブジェクト群に対し、時空間領域との位置関係によってオブジェクトを選択することである。

またここで、時空間領域を以下のように考える。

時空間領域

時空間領域とは、幾何図形によって示される空間領域プリミティブと、それらの演算結合によって表される幾何図形群である。空間領域プリミティブ R_p とは、幾何図形を構成する最小要素であり、表 5.1に示される種類が存在する。

また、空間と時間は本質的にことなるため、別のモデリングを要する。ここでは、シーンの内部時間 t に対し、空間領域プリミティブ R_p の時間関数 $R_p(t)$ を時空間領域プリミティブと呼ぶ。

時空間領域 $R(t)$ は、

$$R(t) = \begin{cases} R_p(t) \\ R(t)\rho R_p(t) \\ R_p(t)\rho R(t) \end{cases}$$

ただし $\rho \in \{and, or, \dots etc\}$ である。

表 5.1: 空間領域プリミティブ

一般図形	制限つき図形	集合
点		疎な任意空間
直線	線分 (区間)	任意曲線
平面	面分 (平面領域)	任意曲面
空間	閉空間	任意空間

すなわち時空間領域とは、直線や平面などの単純な式で示される無制限の領域や、それらに線分や面分などのように制限を加えたような領域が時間にしたがって変化する空間領域を意味する。仮想オブジェクトは各時点におけるポリゴン集合によって表現された閉空間であるため、仮想オブジェクトもまた時空間領域である。

したがって時空間質問とは、シーンの内部時間 INTERNALTIME における時区間 $I(\subseteq \text{INTERNALTIME})$ に対し、二つの時空間領域 $R_1(t), R_2(t)$ について、

$$\forall t \in I, \text{relation}(R_1(t), R_2(t)) = \text{phase} \rightarrow \text{true}$$

または

$$\exists t \in I, \text{relation}(R_1(t), R_2(t)) = \text{phase} \rightarrow \text{true}$$

となる評価を行なうことである。ここで、関数 $\text{relation}(R_1, R_2)$ は、二つの空間領域 R_1 と R_2 との間の空間的な位置関係を返す関数であり、それらの間の位置関係 phase は、共通領域を持たない (disjunction)、接する (meet)、交わる (intersect) などの関係によって示される。Allen による時区間論理など [DELB95, ALLE83] と似て、一般に二つの空間領域の関係は、表 5.2 に示したような関係によって表現することが可能である。

表 5.2 は、disjunction 以外の関係を示したものであり、種類として表 5.3 にあげた関係を考えれば良い。これらの関係は、いくつかの一般化した関係で表すこともできる。例えば、表 5.2 において、meet の関係以外はすべて intersect の関係にあると考えて良い。

二つの空間領域の交わりについて、線分と閉空間の交わり関係の例を図 5.11 に示す。

時空間質問を評価することは、各時点毎に二つの空間領域間の関係を計算することに帰着される。本節では、平面と仮想オブジェクトとの関係を例に、データベースからの値を用いて空間領域の位置関係を計算する手法について述べる。

まず一般に、 \mathbf{r}_0 を通り、法線ベクトル \mathbf{n} である平面上の位置ベクトル \mathbf{r} は、 $f(\mathbf{r}) = \mathbf{n} \cdot (\mathbf{r} - \mathbf{r}_0) = 0$ なる関数 f の方程式によって得られる。ここで \mathbf{n} や \mathbf{r}_0 は、例えばデータベースの値を参照して作成するなど、検索の意図にしたがって任意に設定して良い。

図 5.9 のデータベースにおいて、ポリゴン集合 P について、 $\forall p \in P$ が仮想オブジェクト $vo(\in \text{PARTICIPANT})$ を構成するポリゴンであるとする。このとき、 $P \rightarrow \text{PARTICIPANT}$ なる写像が存在し、インスタンス間の関係を $p \succ vo$ と書くことにする。また同様に、位置ベクトル \mathbf{v} がポリゴン p の頂点であることを $\mathbf{v} \succ p$ と書く。

表 5.2: 二つの空間領域の関係

	点	直線	線分	平面	面分	閉空間
点	equal	onto	onto	onto	onto	inside
直線	include	equal cross	include cross meet	onto cross	divide cross meet	meet through h
線分	include	onto cross meet	equal include onto overlap meet cross	onto cross meet	onto stick divide meet cross	meet through h stick inside
平面	include	include cross	include cross meet	equal cross	include cross meet	meet divide
面分	include	divided meet cross	include stuck divided meet cross	onto cross meet	equal include onto overlap meet cross	meet divide stick inside
閉区間	contain	meet penetrated	meet penetrated stuck contain	meet divided	meet divided stuck contain	equal inside contain overlap meet

表 5.3: 関係の種類

順関係	逆関係	意味
equal	equal	等しい
onto	include	平面上にある、平面中を含む
inside	contain	空間内にある、空間中を含む
divide	divided	分断する、分断される
through	penetrated	貫く、貫かれる
stick	stuck	刺さる、刺される
overlap	overlap	重なる
meet	meet	接する
corss	cross	交差する

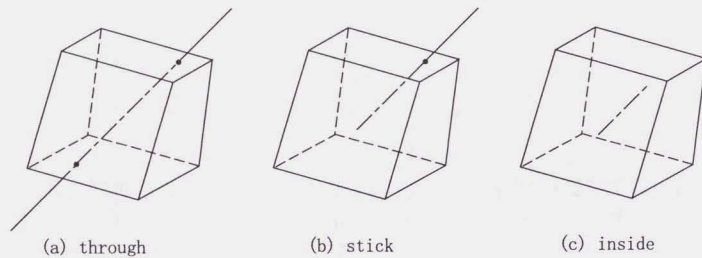


図 5.11: 制限つき直線と閉空間の位置関係

このとき、平面 f とポリゴン p との位置関係 $relation(f, p)$ は、

$$relation(f, p) = \begin{cases} disj(upper) & \text{if } \forall v \succ p, f(v) > 0 \\ disj(lower) & \text{if } \forall v \succ p, f(v) < 0 \\ onto & \text{if } \forall v \succ p, f(v) = 0 \\ cross & \text{otherwise} \end{cases}$$

である。ただし $disj$ は disjunction を表す。したがって、 f と vo との位置関係 $relation(f, vo)$

表 5.4: 質問処理の測定結果

	頂点数	実験 (1)		実験 (2)	実験 (3)	
		該当数 [件]	処理時間 [秒]	処理時間 [秒]	平均誤差 [mm]	処理時間 [秒]
スキャン形状	888	2,570	8.93	16.71	—	43.16
個別モデル形状	108	2,567	1.18	1.79	0.37	5.39
MBB	6	3,556	0.11	0.26	36.38	0.39

は、

$$relation(f, vo) = \begin{cases} disj(upper) & \text{if } \forall p \succ vo, \\ & relation(f, p) = disj(upper) \\ disj(lower) & \text{if } \forall p \succ vo, \\ & relation(f, p) = disj(lower) \\ meet & \text{if } \exists p \succ vo, \\ & relation(f, p) = onto \\ divide & \text{otherwise} \end{cases}$$

によって得られる。

上記の計算は、シーンのある一時点における平面と仮想オブジェクトとの関係を計算したものだが、これをいくつかの時点毎に計算することによって、時間にも依存した質問を記述することができる。つまりこの時空間質問の現実世界における意味は、時間的に変化する(動く)平面について、その面と仮想オブジェクトとの間の関係をすべて算出することを表す。このような、時間依存の平面に基づく時空間質問の方法を特に時空間断面による時空間質問と呼ぶ。

本節で述べた計算手法は、グラフィックワークステーションによって高速に計算できる他、計算の順序を制御することによってすべての計算をしなくても結果を得ることができるため、仮想オブジェクトに基づく時空間モデリングの一つの実現手法であるといえる。

5.4.2 質問処理に関する定量的評価

前節での考察を基に、質問処理の実験を行なった。3.3 節で得たスキャン形状と個別モデル形状を対象にして、任意の無限平面と点をランダムに 10,000 件生成し、前節の (1)、(2)、(3) の演算を行なった。また参考のために、スキャン形状から外接矩形 (MBB) を作成し、同様の演算を行なった。

物体の空間内で移動を問う質問処理の際には、各時点ごとに同様の質問処理を行なえば良い。よって、1 時点あたりでは数面の無限平面について空間オペレータを評価するだけで良いが、一度の質問処理で大量の時点を取り扱う時空間質問の場合には、計算すべき無限

平面のトータルの面数が増す。例えば、1時点あたり無限平面を10面、1分間のシーン当たり100時点の精度でこの質問処理を評価した場合には、10,000面の無限平面の計算は10分間のシーンに対する質問に該当する。したがって、10,000件の無限平面による実験は、エンジニアリングアプリケーションなどでは妥当な量であるといえる。

実験結果を表5.4に示す。実験環境には、Sun Sparc Station 5 (SunOS 4.1.4-JLE1.1.4)を用いた。スキャン形状が、物体の元の形状を最も反映しているという観点から、実験(1)と(3)についてはスキャン形状による解答を正解として評価を行なった。

実験(1)は、個別モデル形状とMBBの該当数の中に、実際には(スキャン形状では)正解でないものが含まれており、それぞれ23件と986件であった。また、スキャン形状では該当するが、個別モデル形状では該当しなかったものが26件であった。この結果から算出すると、適合率・再現率共に99%程度である。この精度は、実験(3)の距離の誤差も含めて、3.3節で述べた個別モデル形状の誤差をそのまま反映した結果となっている。この結果は、4.2節で述べたエンジニアリングアプリケーションなどの質問処理にも有効な精度である。

MBBは再現率は当然100%であるが、適合率が低く、そのまま質問処理に用いるのは適切ではない。MBBを用いた場合には、後で(スキャン形状を用いた)高精度の計算をやり直さなければならず、結局計算コストを要する。個別モデル形状による質問処理に要する時間は、頂点数の少ないMBB程ではないが、スキャン形状をそのまま用いた場合よりも高速に演算を終了することができる。

また、個別モデル形状は、基本モデル形状によって示された意味の統一モデルに基づいて作成される。このため例えば、スキャン形状やMBBでは不可能である「鼻の高さの平均値を算出せよ」といった、集約演算を行なうことができる。個別モデル形状をこのような質問処理に用いた場合にも、精度と処理時間の両面から十分に有用であることが示された。

5.5 実世界データベースのデータ操作

前節で述べたような時空間質問を、前章のデータベースに対して実行するためには、質問機構に以下のような機能が少なくとも必要である。

- あるシーンにおける仮想オブジェクトのCGを、時系列として導出するといったように、データベースを関数によってアクセスし、操作ができる
- 全称質問や存在質問 (universal quantifier) と同等の表現能力を持つ
- 各時点の要素について、*relation* 関数を質問機構の中で計算することができる
- *relation* 関数の並列的計算など、評価の最適化ができる

ここではMMQLによる簡潔な時空間質問の例として図5.12に検索式を示した。図5.12は、あるシーンに対して行なう検索で、まずシーンの内部時間を導出 (TIME) し、それぞれの時点に対してそこに存在する仮想オブジェクト (CG) と、検索対象領域 (REGION) を

```
TIME : apply(ScenelInternaltime)
 $\alpha$  < /TIME > [CG : apply(InternaltimeBehaviorCg), REGION]
 $\alpha$  < /TIME/CG > [OBJ : id, PHASE : relation(../REGION, id)]
 $\beta$  < /TIME/CG > (PHASE = intersect)
```

図 5.12: MMQL による時空間質問の例

導出している。ここでは REGION はマクロの形で用いられており、その内容は検索の意図によって任意に定義して良い。最後に、検索対象領域と仮想オブジェクトの位置関係を算出しておき、その関係が intersect になる仮想オブジェクトを残している。これによって、シーン毎に、時間的に変化する時空間領域 REGION と intersect の関係になる仮想オブジェクトを選択することができる。

この検索式の示す、特徴的な部分としてデータベース操作を関数によって統一的に取り扱うことができる点があげられる。すなわち、データベースの参照も、仮想オブジェクトの位置関係計算も、関数によって表現され、実現されている。また、関数の評価対象が集合として取り扱われるため、集合内の要素への関数評価を並列化するなどの最適化を実現することが可能である。また、得られる検索結果は実世界情報のみでなく、このように対象となる仮想オブジェクト (ここではユーザの指定する領域と交わるもの) のみを残したような新たな仮想空間として得ることができる。

5.6 まとめ

本章では、実データに基づく実世界データベースのためのモデリング手法とその設計について述べた。

実世界データベースの構築については、まずシーンという概念によるデータベースの単位化を行なった。データベースの構築は、ビデオ映像やポリゴン情報を実データとし、そこから得られるシーン情報を、個別モデル形状を基にした仮想空間にマップすることで実現した。個別モデル形状は、実データと意味に基づくモデルの対応をとることによって作成されるため、実世界から取得されたデータと意味との間の mediator としての役割を持つ。また、mediator として個別モデル形状を導入した場合、次の結果を得られた。

- 視覚的には十分な近似形状を得られた
- 十分な精度で質問処理に用いることができた
- 計算量が改善され、質問処理を高速に実現することができた
- 統一された意味に基づく集約演算を実現することができる

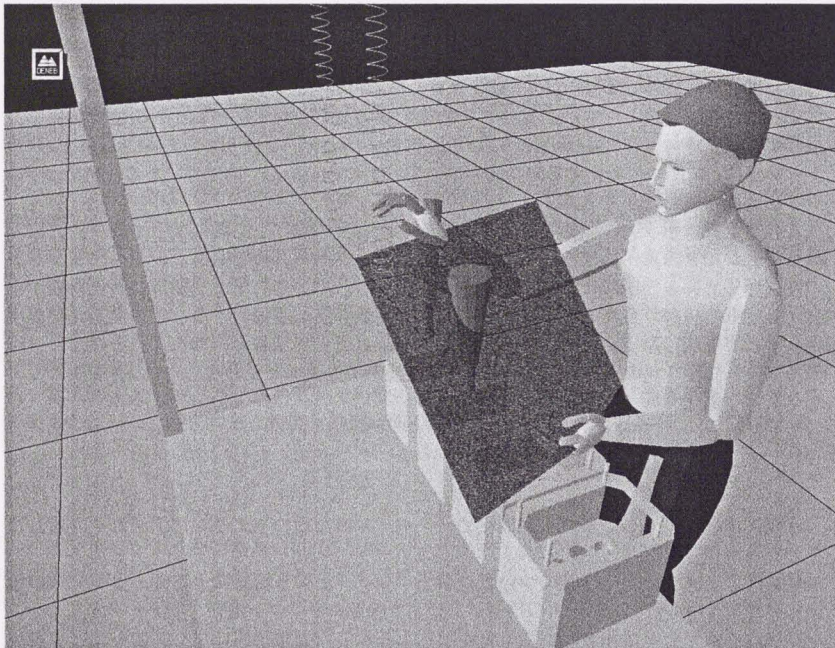


図 5.13: 時空断面による時空間質問

- 画像からのデータ取得を高精度かつ低計算量で実現できる

この結果、本手法による実世界データベースはCGを用いたサイバースペースによって構築され、質問処理もポリゴンオブジェクトに基づく空間オペレータによって行なわれる。このため物体同士の位置関係をできる限り正確に算出したり、検索結果をそのままサイバースペースとすることが可能である。

しかし、本稿で提案された「9層の断面を持つ人間の頭部」という基本モデル形状では、各頂点やポリゴンごとの意味づけがなされておらず、ここから作成した個別モデル形状は mediator として十分であるとはいえない。本稿の目的は個別モデル形状を導入することの定量的な評価を行なうためのものである。よって本稿の実験に用いた個別モデル形状は、「鼻」や「目」などのより細かな意味づけが完全にはなされていない。基本モデル形状の個々の頂点や、辺、ポリゴンに関する意味づけと、より精密な個別モデル形状取得アルゴリズムを導入すれば、「鼻の表面積の平均」や「目の幅の最大値」を求めるなどの集約演算を含んだ検索が可能になる。さらに、人間の頭部以外の基本モデル形状の設計と評価についても課題として残されている。したがって mediator 作成アルゴリズムの改良と精密化は、今後に残された重要な課題の一つである。

また、得られた個別モデル形状を用いて画像から高精度に奥行き抽出を行なったり、物体の姿勢抽出を行なうようなシステムの開発を現在行なっている。これにより、この mediator のさらなる評価を検証することができる。実際に mediator を用いて実世界データベースを構築することのできるトータルシステムの開発と、時間に依存した質問処理の設計、実世界データベースの実装および、時空間質問の評価が今後の課題として残されている。

Chapter 6

結言

本論文では、映像や物体の形状データなどの視覚情報を基にして、実世界をデータベース化する手法について述べた。

データベースシステムの目的は、「実世界の情報を計算機上に表現し、効率良く管理・蓄積・検索する」ことであるといえる。そのためには、データモデルという概念的・理論的枠組を仲介にして、実世界の情報を計算機上のデータにマップする必要がある。ここで実世界の情報として、「対象世界を目で見た」視覚情報をデータベース化する対象にすることは、エンジニアリング分野をはじめとする様々な応用分野において有効である。そこで本研究では、映像や物体の形状データといったマルチメディアデータを基に、視覚情報データベースの構築を試みた。

データベースの概念的枠組であるデータモデルとしては AIS モデルを用いた。AIS モデルは関数型データモデルの一つであり、実世界のあらゆる情報を、主体とそれらの間の 2 項関係に分解して表現するデータモデルである。AIS モデルによって対象世界の情報の概念的な関係だけでなく、映像や、映像を構成するフレーム、フレームに映り込んだ被写体などのマルチメディアデータを基にして対象世界を表現し、蓄積することができた。このデータモデルではフラットな表現方法をとるため、ある特定の視点からの見方だけでなく、2 項関係のパスをたどることだけで様々な関連を参照することが可能であることを示した。

また、本論文では、AIS モデルによって構築されたデータベースのための操作言語である MMQL を提案した。AIS モデルは関数をプリミティブとしたデータベースアクセスメソッドを持つ。MMQL は関数型言語である FP を基にして設計されているため、関数型データモデルである AIS モデルの操作系として良くマッチする。MMQL を用いることで、AIS モデルによるデータベースの検索操作とは、データベースの一部を切り出してある特定の視点から構造化して新たなデータを創り出すことであると考えることができる。マルチメディアデータは、データ構造や操作方法がメディアごとに異なるが、MMQL はこのメディア依存処理を関数として埋め込んで用いることができるため、例えば検索結果として特定の被写体にスポットライトが当てられるなどの強調加工がなされた映像を得ることができる。これによって、検索操作でデータベースには存在しない新しい情報を得ることが可能となることを示し、映像データベースシステムを構築してその有効性を示した。また MMQL は宣言的な関数型言語に基づくため、並列処理抽出の容易さや、事前に評価時間を

概算して検索結果を得られるまでの時間を評価できるなどの特徴を持ち、データベース操作言語として有効である。

次に、映像を単一視点からの一本のムービーだけではなく、多視点画像 + 物体の3次元形状へと拡張することで、より現実の世界に近い実世界データベースの構築を試みた。このために、「シーン概念を導入し、実世界のモデル化を行なった。ここで実世界をモデル化するために、「意味情報」と映像やポリゴンデータといった低レベルのデータとの関連づけを行なうために、簡略化されたCGによる mediator を用いる方法を提案した。本手法で用いた mediator は、人間の頭部の形状をサンプルとして用い、目や鼻、口といった意味づけがなされている。この mediator により、個別の形状データ間に意味の統一を図ることができるため、例えば鼻の体積の平均値を求めるなどの従来の形状データだけでは不可能であった集約演算が可能となる。また、この mediator は、簡略化されたポリゴン形状であるため、時空間質問を高速に評価することができることを実験に基づいて示した。

したがって本研究の成果として、

- 関数型モデルに基づくデータモデルによって、様々な見方のできるデータベース設計を実現した
- FP に基づく操作言語によって、マルチメディアデータベースの操作を実現した
- 内容に基づく検索が可能な映像データベースを実現した
- 映像データベースを拡張し、3次元的な空間をデータベース化するためのモデルを提案した
- 時空間質問処理について考察を行なった
- 物体形状の意味と個別の形状データとを対応づける mediator を提案し、時空間質問処理を高速に実現できることを示した

ということが得られた。

また、AIS モデルと MMQL に基づく汎用データベースマネジメントシステムについては、現在のところ開発中の段階にある。開発中のシステムでは、関数型データ操作言語の事前評価可能性や並列抽出性に着目している。そのため、エージェントシステムによる実装と、メディア依存処理の独立プログラミング環境を提供し、並列コンピュータを用いた高効率なシステム構成を実現しつつある。

本研究によって提案された実世界データベースの構築手法は、映像や3次元形状データなどの視覚情報に基づくものであった。実世界情報をデータベース化するという観点からは、視覚情報に基づくデータベースシステムの発展として、実世界から得られるありとあらゆるデータを基にして実世界データベースを構築するという拡張を考えることができる。このようなシステムはリアルワールドデータベース (Real World Database: RWDB) システムと呼ばれる [ARIS96]。視覚情報から空間の情報に意味づけを行ない、検索によって世界を再構築できるという点で、RWDBの一部を実現できたといえる。しかし本研究で提案した手法では、データベース化する対象情報として視覚情報しか取り扱わなかったこと、明

示的にインデックスつけられたコンテンツのみについてしか内容検索ができないこと、さらに、mediator を実際に設計して基本モデル (意味情報) と個別モデル (インスタンス) との対応づけが技術的に可能なコンテンツのみについてしか時空間質問の対象にできないことなどの限界がある。また、本研究によって提案された手法は、実世界をデータベース化する手法の一部であり、完全なるトータルシステムの実現までには至っていない。

以上の点より、リアルワールドデータベースを実現するという観点から見ると、多くの未知の課題が残されている。その例として、

- より複雑な構造を持った物体 (関節を持つロボットや、人間など) のモデリングが確定していない
- 表面形状モデルでは表現できない物体 (液体や気体などの剛体近似できない物体) のモデルがない
- 形状だけでなく表面のキメ (テクスチャ) の示す情報に対するインデックスづけが行なわれていない
- 検索言語によらない、より直観的なインターフェイスに基づく検索処理が実現されていない
- モデリングのための要素技術が未開発のものが多い
- アプリケーション依存となる部分が多く、システムに一般性がない
- 視覚情報以外のマルチメディアデータ (センサによって得られる熱や圧力、触覚インターフェイスなど) のモデルが存在しない

などの課題が残されており、今後の研究が急がれている。

謝辞

まずはじめに、1993年に筆者が学部生であった頃より本研究をはじめのきっかけをいただき、以来6年間の長期に渡り忙しいお時間の多くをさいて懇切な御指導御鞭撻を賜った横浜国立大学工学部 有澤博教授に最大の感謝の意を表します。また博士課程後期において、常に本研究を見守り、本論文をまとめるにあたり有意義な御助言と御討論を頂いた横浜国立大学工学部 関口隆教授、北田泰彦教授、中川裕志教授に心より感謝致します。さらに、本研究に興味を示して頂き貴重な御意見を頂いたとともに、1995年より2年間に渡り有意義な研究環境を御提供下さった横浜国立大学工学部 影井清一郎教授に深く感謝致します。

本研究を進める上で、日頃からディスカッションに加わって頂き、多くの有意義な意見を頂いたり御協力頂いた、横浜国立大学工学部電子情報工学科有澤研究室に在籍された諸先輩方、大学院生、学部生、研究生の皆様に謝意を表します。その中でも、本研究においてヒューマンモデリングとその応用について御意見・御協力いただいた有澤研究室 大学院生 今井さやか氏に感謝致します。続いて、事務手続き等について多大な御尽力をいただいた有澤研究室秘書 樽沼多恵子様、日野恵利子様に深謝致します。

最後に本研究をまとめるにあたり、計算機環境を整備して頂き、また様々な面から御支援頂いた横浜国立大学工学部電子情報工学科 長谷川紀幸技官に感謝の意を表します。

発表論文

● 査読論文

1. Hiroshi Arisawa, Takashi Tomii, Hitoshi Yui, and Hidehiko Ishikawa: "Data Model and Architecture of Multimedia Database for Engineering Applications," *IEICE Trans. Inf. & Syst.*, Vol.E-78-D, No.11, pp.1362-1368 (Nov.1995).
2. 富井尚志, 有澤博: "マルチメディアデータベースにおける映像モデリングと操作言語," 電子情報通信学会論文誌, Vol.J.79-D-II, No.4, pp.520-530 (Apr.1996).
3. Takashi Tomii, Kiril Salev, Sayaka Imai, and Hiroshi Arisawa: "Human Modelling and Design of Spatio-Temporal Queries on 3D Video Database," Y. Ioannidis and W. Klas eds., *Visual Database Systems 4 (VDB 4)*, IFIP, Chapman & Hall Pub. Co., pp.317-336 (May 1998).
4. 富井尚志, 小林みな子, 有澤博: "仮想CG空間へのマッピングによる現実シーンデータベースの設計," 電子情報通信学会論文誌, Vol.J.82-D-I, No.1, pp.211-222 (Jan.1999).

● 国際会議

1. Hiroshi Arisawa, Ichiro Fukuzaki, Takashi Tomii and Hitoshi Yui: "Implementation of Multimedia Database for Video and Audio Data," Proceedings of the Multimedia Japan, IPSJ, pp.303-309 (1996).
2. Hiroshi Arisawa, Takashi Tomii and Kiril Salev: "Design of Multimedia Database and a Query Language for Video Image Data," Proceedings of ICMCS96, IEEE Multimedia Systems, pp.462-467 (1996).
3. Takashi Tomii and Hiroshi Arisawa: "Functional Data Model and a Query Language for Multimedia Databases," Proceedings of the Workshops Challenges of Application and Challenges of Design, 15th International Conference on Conceptual Modeling ER'96, Brandenburg Technical University Cottbus, Germany, October 9, 1996.
4. Sayaka Imai, Kiril Salev, Takashi Tomii and Hiroshi Arisawa: "Modelling of Working Process and Working Simulation Based on Info-Ergonomics and Real

World Database Concept,” Proc. of Japan-USA Symposium on Flexible Automation, ISCIE & ASME, pp.147-156 (1998.6).

5. Kiril Salev, Takashi Tomii and Hiroshi Arisawa: “Extracting Event Semantics from Video Data Based on Real World Database,” Proc. of International Workshop on New Database Technologies for Collaborative Work Support and Spatio-Temporal Data Management (NewDB’98), Singapore, November 19-20, pp. 555-568 (1998).

● 研究会発表

1. 有澤博, 由井仁, 富井尚志: “映像データベースシステムの構成の一方式”, 1993年アドバンストデータベースシンポジウム論文集, 情報処理学会, pp.181-190 (1993).
2. 富井尚志, 有澤博: “関数型データベースにおける動画の表現,” 1994年データベースワークショップ論文集, 電子情報通信学会 (1994).
3. 有澤博, 由井仁, 富井尚志: “知的作業支援基盤としての映像データベース”, 1994年情報学シンポジウム講演論文集, 日本学術会議他, pp.247-256 (1994).
4. 富井尚志, 由井仁, 石川英彦, 有澤博: “関数型データモデルに基づく映像データベースの構成とその実現,” 第100回データベースシステム研究会記念合同研究会論文集, 情報処理学会, pp.63-70 (1994).
5. 由井仁, 富井尚志, 有澤博: “映像メディアデータベースシステムにおけるフレームデータの加工表示,” 電子情報通信学会技術研究報告, IE94-125, DE94-80(1995).
6. 小林みな子, 今井さやか, 富井尚志, 有澤博: “リアルワールドデータベースにおける3次元物体のモデル化,” 風林火山ワークショップ, 電子情報通信学会技術研究報告, DE96-44, pp.123-128 (1996).
7. 富井尚志, 有澤博: “マルチメディアデータベース向けデータ操作言語 MMQL,” 風林火山ワークショップ, 電子情報通信学会技術研究報告, DE96-52, pp.171-176 (1996).
8. 舩谷和幸, 富井尚志, 有澤博: “関数型データベース検索に適したデータ構造とそのオペレータの実装,” 情報処理学会第53回全国大会, pp.57-58 (1996.9).
9. 家富誠敏, 舩谷和幸, 富井尚志, 有澤博: “関数型データベース管理システムの実現手法,” 情報処理学会第53回全国大会, pp.59-60 (1996.9).
10. 富井尚志, 小林みな子, 有澤博: “リアルワールドデータベースにおける時空間質問の実現,” 電子情報通信学会技術研究報告, DE96-84, pp.61-66 (1997.1).
11. 林路彦, 富井尚志, 有澤博: “映像データベースのための被写体形状の取得,” 情報処理学会, データベースシステム研究会, pp.95-102 (1997.1).

12. 家富誠敏, 富井尚志, 有澤博: “関数型データベースシステムの試作と評価,” 電子情報通信学会 第8回データ工学ワークショップ (DEWS’97) 論文集, pp.161-166 (1997.3).
13. 家富誠敏, 河信司, 富井尚志, 有澤博: “マルチメディアデータベース検索言語 MMQL における機能拡張機構,” 北海道ワークショップ, 情報処理学会, データベースシステム研究会 pp.197-202 (1997.7).
14. 有澤博, 富井尚志, 今井さやか, 小林みな子: “リアルワールドデータベースにおける3次元物体のモデル化と実現,” 情報処理学会第55回全国大会 (1997.9).
15. Kiril Salev, Takashi Tomii and Hiroshi Arisawa: “Conceptual Modeling of Events in Real World Database,” 情報処理学会第55回全国大会 (1997.9).
16. 今井さやか, 富井尚志, キリルサレフ, 有澤博: “工場作業データベースにおける時空間オブジェクトの表現と設計,” 第18回バイオメカニズム学術講演会予稿集, バイオメカニズム学会, pp.181-184 (1997.11).
17. 富井尚志, 有澤博: “リアルワールドデータベースシステムの概念と試作,” 科学研究費重点領域研究「高度データベース」SCS会議 (1997.12).
18. 家富誠敏, 富井尚志, 有澤博: “関数型検索言語 MMQL における並列処理機構の設計,” 電子情報通信学会 第9回データ工学ワークショップ (DEWS’98), pp.29・1-29・4 (1998.3).
19. 小林みな子, 今井さやか, 富井尚志, 有澤博: “時空間データベースにおける物体情報のモデル化とその手法,” 電子情報通信学会 第9回データ工学ワークショップ (DEWS’98), pp.57・1-57・4 (1998.3).
20. 奥井宏昌, 岡本陽介, 今井さやか, 富井尚志, 有澤博: “工場作業データベース設計支援のためのビデオ映像に基づく動作解析,” 情報処理学会, データベースシステム研究会 116-58, pp.241-248 (1998.7).

参考文献

- [AKAM97] 赤間浩樹, 紺谷精一, 三井一能, 串間和彦, “画像内オブジェクトの自動抽出を使った画像検索システム ExSight -写真 (PhotoDisk) への適用-,” 情処研報, Vol.97, No.64, pp.155-160 (1997).
- [ALLES83] J. F. Allen, “Maintaining Knowledge about Temporal Intervals,” *Comm.ACM*, Vol.26, No.11, pp.832-843(1983).
- [ARIS91] Hiroshi Arisawa, Hisayoshi Nagae and Yasuko Mochizuki: “Representation of Complex Objects in Semantic Data Model “AIS” and Imprementation of Set Operators,” *IEICE Transactions*, Vol.E74, No.1, pp.191-203(1991).
- [ARIS96] 有澤博, “リアルワールドデータモデリングについての考察,” 信学技報, DE96-4(1996.5).
- [BACK78] John Backus: “Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs,” *Communications of the ACM*, Vol.21, No.8, pp.613-641(1978).
- [BROO83] R. A. Brooks, “Model-Based Three-Dimensional Interpretations of Two-Dimensional Images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-5, No.2, pp.140-150 (1983).
- [BUNES2] Peter Buneman, Robert E. Frankel and Rishiyur Nikhil: “An Implementation Technique for Database Query Languages,” *ACM Transactions on Database Systems*, Vol.7, No.2, pp.164-186(1982).
- [CHEN76] P.P.Chen: “The entity-relationship model toward a unified view of data,” *ACM Transactions on Database Systems*, Vol.1, No.1, pp.1-49(1976).
- [DELB93] A. Del Bimbo, M. Campanai and P. Nesi, “A Three Dimensional Iconic Environment for Image Database Querying,” *IEEE Trans. on Soft. Eng.*, Vol.19, No.10, pp.997-1011 (1993).

- [DELB95] A. Del Bimbo, E. Vicario and D. Zingoni, "Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic," *IEEE Trans. on Knowledge and Data Eng.*, Vol.7, No.4, pp.609-622 (1995).
- [EGEN94] M. J. Egenhofer, "Spatial SQL: A Query and Presentation Language," *IEEE Trans. on Data and Eng.*, Vol.6, No.1, pp.86-95 (1994).
- [FIEL88] Anthony J. Field and Peter G. Harrison: "Functional Programming," Addison-Wesley, Wokingham(1988).
- [FUJI92] 藤川和利, 下條信司, 松浦敏雄, 西尾章治郎, 宮原秀夫: "オブジェクト指向に基づくハイパーメディアシステム Harmony の構築," 電子情報通信学会論文誌 (D-I), Vol.J75-D-I, No.11, pp.1015-1024(1992).
- [GIBB93] Simon Gibbs, Christian Breiteneder and Dennis Tschritzis: "Audio/Video Databases: An Object-Oriented Approach," IEEE 9th International Conference on Data Engineering, pp.281-390(1993).
- [HIGU96] 樋口和則, マーシャル ヘバート, 池内克史, "複数視点レンジデータからの3次元物体モデル構築," 信学論 D-II, Vol.J79-D-II, No.8, pp.1354-1361 (1996).
- [HIRO97] M. Hirose, "Image-Based Virtual World Generation," *IEEE Multimedia*, Jan-Mar 1997, pp.27-33(1997).
- [ISHI97] 石川正敏, 高倉弘喜, 植村俊亮, "3次元仮想空間を用いたコミュニケーション支援," 信学技報, DE96-85, pp.67-72(1997.1).
- [ISOT86] 磯田浩, 鈴木賢次郎: 図学入門 コンピュータグラフィックスの基礎, 東京大学出版 (1986).
- [KANA97] T. Kanade, P. Rander and P. J. Narayanan, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes," *IEEE Multimedia*, Jan-Mar 1997, pp.34-47(1997).
- [KATA95] Ryoji Kataoka, Tetsuji Satoh and Ushio Inoue: "Architecture and Storage Structure of an Interactive Multimedia Information System," *IEICE Transactions on Information and Systems*, Vol.E78-D, No.11, pp.1354-1361(1995).
- [KOBA96] 小林みな子, 今井さやか, 有澤博, "リアルワールドデータベースにおける3次元情報の取得と表現," 情報処理学会アドバンストデータベースシンポジウム'96 論文集, pp.96-106(1996).

- [KURO97] 黒木進, 石塚健作, 牧之内顕文, "時空間データベースシステム Hawks における時空間質問の表現," 情処研報, Vol.97, No.64, データベースシステム 113-58, pp.347-352(1997).
- [MASU95] Yoshifumi Masunaga: "An Object-Oriented Approach to Temporal Multimedia Data Modeling," *IEICE Transactions on Information and Systems*, Vol.E78-D, No.11, pp.1477-1487(1995).
- [MASU97] Y. Masunaga, "An Interval-Based Approach to a Spatio-Temporal Data Model for Virtual Collaborative Environments," Y. Kambayashi and K. Yokota eds., *Cooperative Databases and Applications — Proc. of the International Symposium on Cooperative Database Systems for Advanced Applications (CODAS)*, World Scientific, pp.420-427(1997).
- [NAGAE95] 永江尚義, 有澤博: "関数型データベースのための視覚的検索言語," 情報処理学会論文誌, Vol.36, No.6, pp.1474-1485(1995).
- [NAGAS92] 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).
- [OOMO93] Eitetsu Oomoto and Katsumi Tanaka: "OVID: Design and Implementation of a Video-Object Database System," *IEEE Transaction on Knowledge and Data Engineering*, Vol.5, No.4, pp.629-643(1993).
- [OREN88] J. A. Orenstein and F. A. Manola, "PROBE Spatial Data Modeling and Query Processing in an Image" Database Application, *IEEE Trans. on Soft. Eng.*, Vol.14, No.5, pp.611-629 (1988).
- [SATO96] K. Satou, I. Kitahara and Y. Ohta, "3D Image Display with Motion Parallax by Camera Matrix Stereo," *IEEE, Proc. of the Intl. Conf. on Multimedia Comp. and Syst.(ICMCS96)*, pp.349-357 (1996).
- [SUMI98] 角保志, 河井良浩, 吉見隆, 富田文明, "セグメントベーストステレオによる自由曲面体の認識," 信学論 D-II, Vol.J81-D-II, No.2, pp.285-292 (1998).
- [TABU93] 田淵仁浩, 村岡洋一: "動画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," 電子情報通信学会論文誌 (D-I), Vol.J76-D-I, No.6, pp.288-299(1993).
- [THEO96] Y. Theodoridis, M. Vazirgiannis and T. Sellis, "Spatio-Temporal Indexing for Large Multimedia Applications," *IEEE, Proc. of the Intl. Conf. on Multimedia Comp. and Syst.(ICMCS96)*, pp.441-448 (1996).

- [TSUK97] 塚本昌彦, “透明人間: 実空間と仮想空間を統合するコンピューティング環境,” シミュレーション, Vol.16, No.1, pp.20-27 (1997).
- [UEDA93] 上田博唯, 宮武孝文, 炭野重雄, 長坂晃朗: “動画像解析に基づくビデオ構造の視覚化とその応用,” 電子情報通信学会論文誌 (D-II), Vol.J76-D-II, No.8, pp.1572-1780(1993).
- [YAMA96] 山本正信, 川田聡, 近藤拓也 a, 越川和忠 “ロボットモデルに基づく人間動作の3次元動画像解析,” 信学論 D-II, Vol.J79-D-II, No.1, pp.71-83 (1996).

