半自動ビデオアノテーションと
それに基づく意味的ビデオ検索に関する
研究

山本 大介

名古屋大学工学部電気電子・情報工学科

2003年2月

目 次

概要

近年、Web ページをはじめ、さまざまなオンライン情報に対する検索が頻繁に行われている。しかしながら、オンラインのビデオコンテンツに対する検索はいまだ実用化されているとは言い難い。ビデオコンテンツに対する検索にはさまざまな手法が存在するが、ビデオを全自動で解析した結果に基づいて検索する場合、精度の観点から見るときわめて不十分である。検索の精度を十分に実用的なレベルに引き上げるためには、ビデオコンテンツに検索や変換・編集等に有効な意味内容記述をなんらかの方法により関連付ける必要がある。そこで、まずビデオコンテンツの自動解析を行い、その後、人間がその解析結果を効率よく修正・補完できるツールを作成した。さらに、そのツールを使用して得られたアノテーションデータ（カラーヒストグラム情報・カット情報・オブジェクト意味属性情報など）に基づいて、高度な意味的ビデオ検索を Web ブラウザ上で自然言語を用いて行うシステムを試作した。

第1章　はじめに

1.1　研究の目的と背景

現在の情報科学が解決すべき本質的問題とは何か？そう考えたときに、重要なものの一つとして情報検索があげられるであろう。現在さまざまな情報が、新聞・放送・雑誌・インターネットなどを通じて溢れ返っている。我々は特にインターネットにある情報に関してはだれでもいつでもアクセスすることができる。しかしながら、あまりにも多種多様な情報は必ずしも、個人一人一人において重要であるとは限らない。ある人にとってその情報は重要であっても別の人にとっては重要でないということが日常的に起こりうる。インターネットから個人に特化した情報を取ってくるためにWeb検索などの技術が盛んに研究・実用化されている。Webページに対する検索ではGoogle [15]等が有名であり、十分実用的なレベルにあると考えられ、筆者も愛用している。しかし、それはテキスト検索やイメージ検索などに限られ、動画や音楽などのコンテンツに対しての検索は実用化されていない。それはなぜか。その理由は二つあると考えている。第一に、Webページのように無料でいつでも誰でも見られるビデオコンテンツはまだ少ないということ、第二に、ビデオコンテンツに対する検索はテキスト検索に比べて難しく、検索コストがかかることが考えられる。しかしながら、これからの時代、FTTH (Fiber To The Home) やADSL(Asymmetric Digital Subscriber Line) に代表される広帯域インターネットの発達に加え、ビデオカメラの低価格化・ハードディスクの容量増加により個人ページ・商用ページ問わずデジタル動画コンテンツが氾濫する時代がくることは容易に想像される。現在でも毎日膨大に放送されるテレビ放送に対する検索と蓄積は難題であり、また検索・要約等の要求は非常に大きいと考えられる。

動画コンテンツに対する検索には様々な手法 [7] があるが、コンピュータで全自動解析した結果を元に検索することは非常に難しい。せいぜい、音声認識と画像のパターンマッチングやオプチカルフロー等を組み合わせるのが限界であり、現状のノイズや環境音に弱い音声認識技術、特殊な条件下でしか機能しない画像処理技術では限界だと思われる。検索の精度を十分に実用的なレベルに引き上げるためにはビデオコンテンツに検索や変換・編集等に有効な意味内容記述（本研究ではアノテーションと呼ぶ）をなんらかの方法により付加する必要がある。そこで、本研究では、動画コンテンツに対し、コンピュータが自動解析した結果をユーザが効率よく修正・補完できるツールを作成した。つまり、コンピュータが得意な問題はコンピュータが自動解析し、人間が得意な問題は人間が担当するという、人とコンピュータの協調作業を支援するツールである。さらにそのツールを使用して得られたXMLアノテーションデータに基づいて、高度な意味的ビデオ検索を、Webブラウザを用いて自然言語で行うシステムを試作した。

動画コンテンツに対する意味内容を記述する規格として、XML [1] ベースのMPEG-7 [4]

が存在するが、現状ではそれほど普及していない・まだ規格化の段階である・記述方式が複雑であるという理由から採用を見送ったが、将来的には対応も考えている。また、アノテーション一般に言えることだが、コンピュータによる自動解析も含まれるが、ある程度人間が介入する必要がありそれには少なからず労力が必要である。そのために、アノテーションを作成することはナンセンスであるという人もいる。しかし、それは間違いである。アノテーションを必要とせずに動画コンテンツに対する検索ができればそれほど素晴らしいことはないが、動画に対する検索・要約は古くから研究されている分野であるにも関わらずまだ十分な精度で検索できる例はない。それよりも、アノテーションの手助けによって、動画コンテンツに対する検索を実現した方がよいだろう。まだ、アノテーションに対する労力は大きく面倒なものであるが、面倒であればあるほど便利にしようとする欲求が高まり、研究目的が分かりやすくなり、より発展すると考えられる。アノテーション作成が自動化されれば、それは動画コンテンツに対し全自動で解析したといえるわけで、アノテーションによる半自動解析であろうが、全自動解析を目指すものであろうが、最終的な目標は同じである。コンピュータによる全自動解析がトップダウンから攻める技術だとすれば、アノテーションツールの解析はボトムアップから攻める技術である。コンテンツを解析するという点において本質的に両者は同じであり、アノテーションベースの方が機械や人間が何をやっているかを明確に示すことができる分有用であると考えている。また、アノテーションは動画コンテンツ製作者側が使用すればそのコンテンツの内容を正確に相手に伝えるという使い方も可能である。動画を要約する場合は、このように要約して欲しいという製作者側の思いもあるはずなのでそれを動画コンテンツに重要度の形で記述できれば製作者と要約をする編集者（場合によってはユーザ自身）の間での意思疎通が図りやすくなる。

さらに、動画コンテンツにアノテーションを作成する場合、著作権上の問題も考えなければならない。製作者側がアノテーションを作成する分には問題ないが、ユーザが独自にアノテーションを作成して、インターネット等で公開する場合、アノテーションを作成したデータにはシナリオや発言情報の知的財産が含まれる可能性があるので、アノテーションデータを不特定多数の人と共有する場合、知的所有権の問題も考えなければならない。アノテーションはアノテーションを作成する人の主観と機械での自動解析による客観的データが含まれているが、批評や感想の一種であるととらえれば、著作権的に問題ないように思われるが、現行法での正確な捉え方は筆者の理解の範疇を超えるので詳しく追及することはしないが十分な注意が必要であることは留意しなければならない。

動画コンテンツにアノテーションを行ううえで重要なことは、大きく分けて３つあると考えている。１つはアノテーションする負担がなるべく小さいこと。２つ目は、アノテーションしたデータが特定のアプリケーションに依存することなく、検索・要約・編集等さまざまな場面で活用できること。３つ目はアノテーションしたデータの意味を人間・機械ともに一意に特定できることである。それらの点を踏まえたうえで今回の研究を行うことにした。

1.2 本研究で提案するシステムの概要

本研究では、MPEG ビデオなどの動画コンテンツに対し、いかにして簡単かつ十分なアノテーションを行うかについて議論し、さらにその支援ツール及び応用システムの実装を目的とする。具体的には、本研究で製作されたツールをもとにして得られたアノテーションデータを元に、自然言語での意味的ビデオ検索を実現した。

本ツールによって作成される XML アノテーションデータには、コンピュータによって自動解析されるデータと、ユーザがツールを使って付与されるデータが含まれている。コンピュータによって自動解析されるデータには、動画像に含まれるカットの時間位置や、カラーヒストグラム情報、オブジェクトトラッキング情報などがある。ユーザーがツールを使って付与するデータには、シーンやオブジェクトに対する意味情報をあらわす意味属性情報やコメント情報などが含まれる。さらに、それぞれの意味属性は検索や要約に便利な意味内容も定義されている。つまり、解析されたデータとともに、そのデータの意味する内容も記述されており、アノテーションデータ単独で、意味内容が理解できるよな仕組みになっている。そのために特定のアプリケーションに依存しにくく、永続性・独立性の高いアノテーションデータが得られるように工夫している。

本システムの概要は図 1.1 のように、インターネットで扱うことを想定したシステムとなっている。複数のユーザがインターネット上に存在する MPEG ムービーコンテンツに対し、ビデオアノテーションエディタを用いて、アノテーションデータを作成しそれをデータベースに登録する。現状ではアノテーション管理サーバは未実装であるが、複数ユーザが同一のムービーコンテンツに対してアノテーションする場合などに、バージョン管理や複数データの統合などを考慮する必要がありそのためのサーバとしてアノテーション管理サーバが必要になると考えられる。

現在はアノテーション管理サーバは未実装なので直接 XML データベースにデータを入れている。ユーザが検索サーバに検索要求を出すと、検索サーバがアノテーションデータをもとにして検索を行い、検索結果を Web ブラウザに表示する仕組みである。検索結果をユーザに適した形で表示するために、長尾らが開発した Semantic Transcoding [2] も利用することを検討している。

また、このアノテーションデータを元に XML データベースである Xindice [6] を用いてデータベースを構築し、Java Servlet を用いて、従来は困難だと考えられていた自然言語による動画像検索システムをも試作し、アノテーションとその利用方法についても言及することにより、より実用的なビデオアノテーションについての提案をすることができた。

また、本研究では動画像を映像の立場から研究することにした。音声部分の研究は今回は対象外とした。

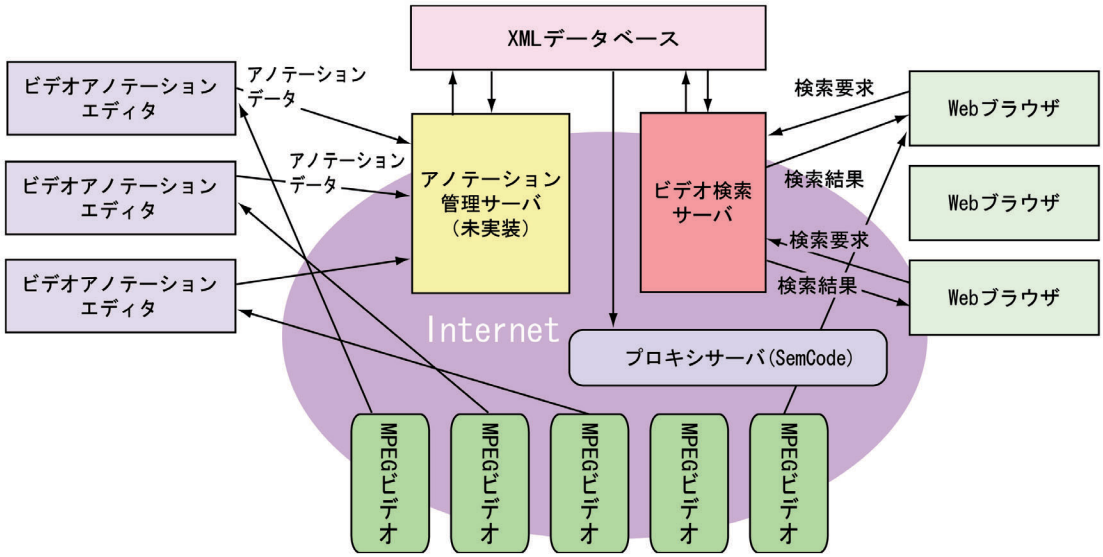
XMLデータベース
ビデオアノテーション
エディタ
アノテーション
データ
ビデオアノテーション
エディタ
アノテーション
データ
ビデオアノテーション
エディタ
アノテーション
管理サーバ
（未実装）
ビデオ検索
サーバ
検索要求
検索結果
検索要求
検索結果
Webブラウザ
Webブラウザ
Webブラウザ
Internet
プロキシサーバ(SemCode)
MPEGビデオ
MPEGビデオ
MPEGビデオ
MPEGビデオ
MPEGビデオ

図 1.1: システム概略図

第2章　ビデオアノテーション

我々はテレビやインターネットのストリーミング配信・ビデオ・DVD 等様々な形で動画コンテンツを見て楽しんでいる。しかしながら、膨大な動画コンテンツが存在しながらそれらを検索したり要約したりすることはできない。それは、動画コンテンツ自体は単なるバイナリデータであり、動画の意味する内容が記述されていないためである。その意味内容を記述する規格として MPEG-7 [4] が規格化されている。MPEG-7 は MPEG によって規格化された動画コンテンツの意味内容を記述するための規格であり、動画圧縮を扱った既存の MPEG(MPEG-1 MPEG-2 や MPEG-4) などとは違って、それ自身は動画のバイナリ情報を含んでいない。つまり、動画コンテンツに対するメタデータ（本研究ではアノテーションデータと呼んでいる）を記述する規格である。

そこで、本研究の要になるのは、いかに効率のよいアノテーション作成ツールを作るかということである。ビデオにアノテーションやインデックス情報等を付与するツールは様々なものが存在するが、Web に最適化されたアノテーションツールは少ない。MPEG-7 記述ツール [9][10][11] のように、XML ベースのツールは存在するが、高度に自動化されたツールはなく、大部分が手入力によるものであり、使い勝手が悪く試作段階のものでしかないという事実は否めない。そこで、本研究で、効率よくアノテーションをするツールとしてビデオアノテーションエディタ [図 2.1] を試作した。このツール自体は長尾らが開発したビデオアノテーションエディタ [3] を元にしているが、新たに作り直したツールである。

ビデオアノテーションエディタの機能としては以下のものを備える。

1. カットの自動検出と修正機能
2. カット・シーンの重要度の推定と修正機能
3. カットの階層構造の推定と修正機能
4. カットに対する選択式手動アノテーション
5. オブジェクトトラッキング
6. オブジェクトに対する選択式手動アノテーション
7. アノテーター情報の付与
8. コンテンツ情報の付与
9. 色ヒストグラムの自動抽出
10. XML 形式での保存・出力

図 2.1: ビデオアノテーションエディタ

オブジェクトトラッキングやカット検出、ヒストグラムの自動抽出等はほぼ完全に自動化されている。もちろん、ヒストグラム以外は誤検出も考えられるので、直感にあった修正機能もつけている。本来ならば、音声認識部分にも対応している必要があるが、現状ではサポートされていない。読み込める動画コンテンツの種類は、Microsoft の DirectX8 に含まれる DirectShow でサポートされる形式であり、MPEG-1, MPEG-2, MPEG-4 などに対応している。

2.1　開発環境と動作環境

2.1.1　開発環境

OS が Microsoft Windows XP Professional、CPU がモバイル Pentium3 1.2GHz、メモリ 640MB、HD30GB の B5 ファイルサイズのノート PC である ThinkpadX24 で主に開発を行った。コンパイラは Microsoft Visual C++ 6.0 Professional Edition(SP4)、にライブラリとして、XML ファイルを扱うために Xerces for C++ (xerces-c2_1_0-win32)、JPEG ファイルを扱うために LibJpeg.lib を、動画を扱うために、Microsoft DirectX 8.0a SDK を利用した。

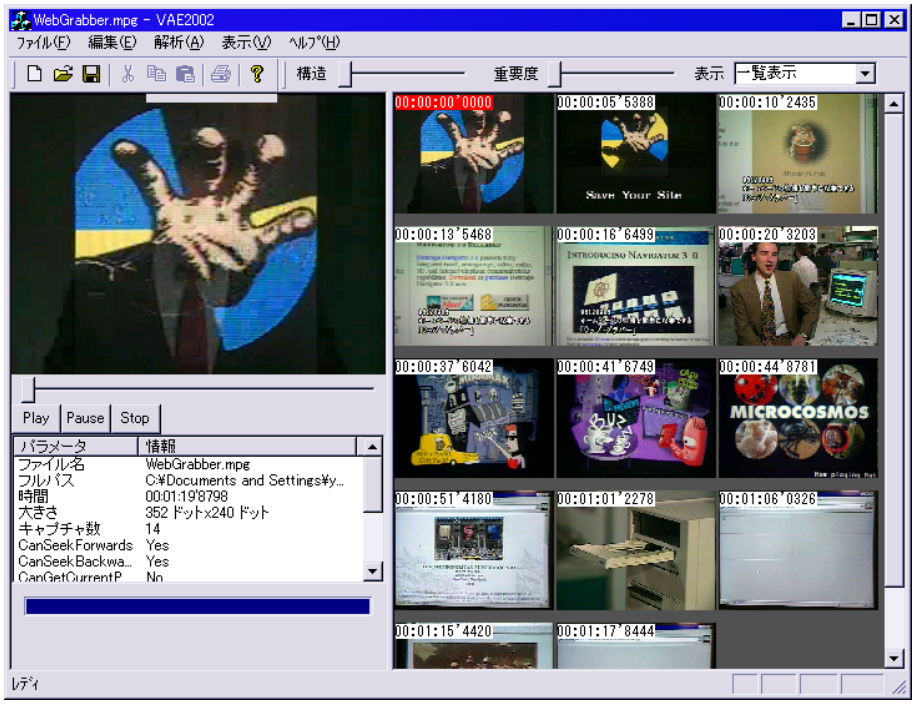

図 2.2: カット検出画面

2.1.2 必要動作環境

本ツールは、Microsoft Windows2000 もしくは WindowsXP での動作を確認している。Windows2000 の場合は、Microsoft Direct X8.0 以降がインストールされていることが条件である。

2.2 カットの自動検出と修正機能

カットの自動検出機能を備える。本研究におけるカットとは、映像の途切れから次の途切れまでの一連のストリーム映像である。カットの自動検出とは、そのカットをコンピュータによる全自動解析により検出する手法である。カットの検出画面は [図 2.2 に示す。

カット検出アルゴリズムとしては以下のようになる。

DirectShow の機能の一部としてある SampleGrabber の機能を用いて動画から静止画を１フレームごとにとりだして評価を行う。数フレーム置きに行っても良いが、MPEG などの圧縮形式は１フレームごとに前フレーム情報を利用してデコードする必要があるため、数フレームごとにやるのは効率がよくないためである。まず個々のフレームにおいてヒストグラムを生成する。本研究では RGB 要素それぞれ４ビット分合計１２ビット、４０９６分割して生成した。ヒストグラムでの前フレームとの比較により、ある閾値以上の差が生じた場合は新たなカットとして検出するプログラムである。さらに、動画の任意での位置で右クリックのコンテキストメニューを選択することにより、カット分割・カット併合をマウスで簡単に行えるようにもした。

カット検出の具体的なアルゴリズムは次のようになる。

1. DirectShow の SampleGrabber 機能を使い MPEG コンテンツから静止画をとりだす。
2. RGB 要素をそれぞれ 4 ビット合計１２ビット（４０９６）分割をしてヒストグラムを計算する
3. 前フレームとのヒストグラムの要素ごとの絶対値誤差の合計を求める。
4. 3. の絶対値誤差の合計がある閾値以上になったら、カットであると認識し終了。
5. メディア時間を 1 フレーム分進めたのち 1. に戻る。

このままでもカットの検出は可能であるが、動きが激しいシーンではカットが大量に誤検出してしまう。そこで、動きが激しいカットをひとつのカットであると認識するために以下の手法を用いた。

1. 上記アルゴリズムでカット検出を行う。
2. 1. を繰り返し、8 フレーム以内に再びカット検出を行ったら、それをカット検出とはせずに、２へ。
3. １へ飛ぶ。

8 フレームとしたのは、秒間３０フレームとすると、0.26 秒にあたり、人間の目では、これ以上短いカットはカットと認識しないだろうという限界であると考えたからである。

今回はカメラの物理的なカットごとに分割したので、サッカー中継や講義ビデオなど同じシーンが長く続くような映像に対してのカット検出の自動化には対応していない。それは、講義ビデオ等は物理的なカメラのカットは少なく、話の論理的な推移や PowerPoint のスライド等に応じて推移するべきであるからである。しかしながら、手動でカット分割も行えるので映像をみつつ、ここでカットであるということがわかればカット分割も行える仕様ではあり、自動化はされていないものの手動で簡単にカット分割・併合が行えるので、実用上はそれほど問題がないと思われる。

2.3　カット・シーンの重要度の推定と修正機能

カットやシーンに対して、重要度というものがアノテーションできると考えられる。ここでいう重要度とは、そのカットがストーリ上重要なカットであるかをあらわす指標であり、主にビデオコンテンツの要約をする場合などに使える。さらに検索時にも重要なシーンほど上位に検索結果がでるのがよいので検索などにも利用可能であると考えられる。

重要度というものは、ある時（ある検索要求時）には重要であるけれども、そうでない場合はそれほど重要でないという場合もあり、ある意味曖昧な指標である。それでも、ある程度は客観的にこのシーンは重要で、別のシーンは重要でないという事が考えられる。そこで、本ツールはカット一つ一つに対して重要度をアノテーションできる機能もある。さらに、重要度をある程度自動推定する機能もつけた。重要度は基本的に長いカットほど重要であるという過程のもとに以下のようなアルゴリズムを用いた。

図 2.3: 構造解析と修正画面

おのおののカットに対し重要度の推定と修正機能をつけた。重要度を I、カットの時間を x、全カット数を m のビデオコンテンツの全時間を L とすると、

$$I = \frac{xm}{L}$$

で表現した。つまり長いカットほど重要ということにしたのだが、もちろんこれでは直感にそぐわない場合もある。その場合カットを Shift キーで押しながらマウスでドラッグすることにより、簡単かつ直感的に重要度を修正できるように工夫している。その様子の操作画面を図 2.3 に示す。

映像の長さだけでそのシーンの重要度を推定するのは危険が多い。同じ静止画面が永遠と続く映像と、動きの激しい映像とでは重要度が違うであろうし、無音映像と雑音映像と人が話している映像とではまた重要度は変わってくると考えられる。これらの情報を使いつつシーンの重要度というものを推定することは可能であると思われるが、必ずしも一意で決まるとは思えない。重要度を推定するにはストーリ展開などが重要になってくると思われ、「しかし」、「つまり」などの言葉がストーリを理解するうえでキーとなる言葉であるとは思われるが、それだけでストーリを得る事はできない。そのあたりの重要度の推定は非常に難しくまたあいまいな部分も多く今後の課題となる。むしろその部分は人間が得意な分野であり人間が推定するのが適切であるとも思われる。

2.4 カットの階層構造の推定と修正機能

カットにはシーンごとの固まりなど階層構造が存在すると考えられる。ここでいう階層構造というのは、カットのストーリ上の階層的構造のことでありストーリを理解するため

に役に立つ。具体的には、ある連続するAというカットとB,C...というカットが同じ内容であるカットであるとするならばそれは、同じシーンであると考えられB,C...というカットはAというカットの子になると考えられる。さらに、もっと大域的なストーリの流れを考えると、起承転結などのストーリ上のまとまりがあるとかんがえられ、それをひとまとまりにすれば、階層構造が表現できると考えられる。このようにして、複数存在するカットを意味的な階層構造により表現できると考えられる。

その階層構造をある程度自動的に推定するためにシーンのまとまりを自動抽出することにした。連続するカットA1, A2, A3, ...とすると、A1とA2が同じシーンであるとの認識にはヒストグラムを用いて、A1のカットの全フレームのヒストグラムの合計をH1、A1のフレーム数をm1、A2の全フレームヒストグラムの合計をH2、A2のフレーム数をm2とするとA1とA2が別シーンである確率Pを

$$P = |\frac{H1}{m1} - \frac{H2}{m2}|$$

として、Pがある閾値以上ならば同じシーンであると認識した。ただし、これで検出できる階層構造は二段階までであるので修正機能としては、マウスでドラッグすることにより、階層構造を変更できるようにした。

今回は映像は階層構造であるという前提で研究を行ったが、必ずしも階層構造であるとは限らないと思われる。あるいはグラフ構造であらわせるのかもしれない。特に時系列情報を持ったコンテンツは単純な構造をもたない場合もある。またたとえ存在しても、分かりにくいものもある。しかしながら、カットの集合でシーンが成り立っているなど局所的な階層構造は必ず存在するわけで、階層構造を記述することは悪くないと思われる。

階層構造が表現できれば、検索結果を表示する場合ある特定のカットだけを表示するよりもそれよりも上位階層、つまり１シーンを表示したほうがよいからである。

2.5　内容に関する意味的属性アノテーション

カットやオブジェクトに対してアノテーションを行う場合は、自然言語で無秩序に記述するよりも、あらかじめ登録されている意味属性を入力したほうが、意味内容が一意に捕らえることができ都合がよい場合が多い。また、意味属性をあらかじめカテゴリ化されたメニューからマウスで選択したり数値を選択したりできたとすればそちらのほうがアノテーションしやすく使い勝手がよいと思われる。ただ、意味属性を入力するとすればそれで表現できる事項は限られてしまうので複数の意味属性を組み合わせてやることにより、正確な表現が記述できる。たとえば、若い男が話しているというものをアノテーションしようとする場合、「若い」「男」「話す」という三つの意味属性を同時選択してやればよい。それでも、意味属性の数は膨大になってしまうので、カテゴリわけすることによりある程度意味属性を探しやすくなるように工夫した。具体的には、項目をカテゴリと詳細の二段階に分けた。たとえば、humanのカテゴリには、male,female,young,child,old,adultなどの意味属性があるなど、意味的に分類可能な意味属性を整理している。

操作画面を図2.4に示す。

カットに対する意味的属性アノテーションの定義ファイルとして、sceneDefinitions.xmlという定義ファイルを用意した。また、オブジェクトに対するアノテーション定義ファイ

図 2.4: 内容に関する意味的属性アノテーションの操作画面

ルとして、objectDefinitions.xml を用意した。定義ファイルの形式としては以下のリストのようにした。

```
<?xml version ="1.0" encoding="Shift_JIS"?>
<objectDefinitions version="0.1">
  <category id="カテゴリ意味属性" jp="カテゴリ名（日本語）">
    <item id="詳細意味属性" jp="詳細名（日本語）">
      <synonym word="同義語"/>
                ...（synonymが複数存在）
    </item>
     ...（itemが複数存在）
  </category>
     ...（categoryが複数存在）
</objectDefinitions>
```

リスト 1　objectDefinitions.xml の定義ファイル形式

```
<?xml version ="1.0" encoding="Shift_JIS"?>
<sceneDefinitions version="0.1">
  <category id="カテゴリ意味属性" jp="カテゴリ名（日本語）">
    <item id="詳細意味属性" jp="詳細名（日本語）">
      <synonym word="同義語"/>
```

```
                ...
    </item>
     ...
  </category>
     ...
</sceneDefinitions>
```

リスト2 sceneDefinitions.xmlの定義ファイル形式

また、定義ファイルは、ユーザによって追加される場合も想定している。これは、あらかじめ用意された定義ファイルでは適切でないオブジェクトやシーンもあると考えられるためである。しかしながら、単に新しい定義（意味属性）を追加しただけでは、コンピュータはその定義の意味を理解できない。今回は動画コンテンツを自然言語レベルでの取り扱いにしようとしているために、新しく追加される定義（意味属性）に対し、新しい定義に対する同義語を列挙する形でその意味属性の意味を定義した。単なる日本語一語での意味記述では、検索時に完全一致での検索にマッチしないので、複数個同義語を記述することにより一致する可能性が広がるためであると考えている。現段階では複数の意味を手入力で記述しているが、同義語辞典であるシソーラスなどを用いれば、自動入力も可能であると考えている。ただ、単純にシソーラスで追加すると、あまり関連のない語まで追加される可能性があるので、修正を可能にする必要性もあると考えている。

XML定義ファイルには、新たな項目を作るだけでなくその項目の説明をする方法にはRDFスキーマ[5]などグラフ構造を用いた定義の表現方法が存在するが今回はより簡略かつ検索に使いやすいように、新しい項目に関するさまざまな同義語(日本語、英語を含む)を列挙することにした。この方式ならば、手軽に項目追加が可能であるし、検索時に完全一致、あるいは、部分一致が容易であると考えられるからである。

今回は、カテゴリと詳細の二段構成にしたが、かならずしも二段である必然性はなく三段、四段構成である可能性もある。

2.6 オブジェクトトラッキング

ここでいうオブジェクトとは、人や動物、物やテキスト等映像中に登場する独立した物体をいう。テロップや字幕等編集によってつけられた人工データもオブジェクトとする。

オブジェクトにアノテーションをする場合、カットにアノテーションをする場合と同様にそのオブジェクトが出現する時間と消滅する時間を正確に知ることが重要な場合が多い。単なる検索の場合は、その時間は正確である必要性は薄いが、要約や編集等の利用時において重要になる場合が多いからである。そのオブジェクトの動作を追尾する必要があり、手法としては一般的なテンプレートマッチングを採用している。キーフレームのオブジェクト矩形選択範囲との画素間比較により絶対値誤差の合計が閾値以内であればオブジェクトが存在していると認識している。

一般に、MPEG等の圧縮された動画形式はランダムアクセスや逆順シークが苦手である。

図 2.5: オブジェクトアノテーションの操作画面

それは、動画の場合、画面全体の情報を全て記憶しているＩフレームからの差分情報を記憶しているために、Ｉフレームから目的のフレームまで１フレームごとに計算する必要があるためである。しかしながら、動画解析の場合は、時間軸とは逆方向に順に検索したりする必要もあるが、それは非常に効率が悪い。そこで、逆方向にシークする場合は、あらかじめ時間軸過去方向にごとに３秒ごとに無圧縮状態に展開したのちバッファにため処理を行うことにより、高速化する工夫をした。本来ならば、Ｉフレームを基準に展開するのがすじであるがその方法が分からなかったので、３秒ごとにバッファリングすることにした。

また、トラッキングした時間が正確であるかどうかを認識するために、開始フレームと終了フレームそれにその前後 0.1 秒のフレームを表示することにより一目で確認できるように工夫している。

オブジェクトトラッキングの操作画面として図 2.5 に示す。

2.7 アノテータ情報・コンテンツ情報の付与

アノテーションを行う場合、アノテーションを行う人（本研究ではアノテータと呼ぶ）によってアノテーション結果が大きく異なる場合が多い。そのためにだれがアノテーションしたかという情報が重要であり、それを付与する必要がある。そのためのダイアログも作成した。将来的には、同一のコンテンツに複数の人がアノテーションした場合などにそれぞれのアノテーション結果をマージする等の操作が必要となってくるが、その場合にもアノテータ情報は重要になると考えられるので、その意味もこめて付与した。また、コン

図 2.6: コンテンツタイトルダイアログ

テンツ情報を格納するダイアログボックスも作成した。コンテンツのタイトル情報や製作者情報・著作権情報などコンテンツ全体に関わる情報を付与するダイアログである。両ダイアログとも Windows の標準インタフェースを用いている。コンテンツタイトルの操作画面は図 2.6 に、アノテータ情報の操作画面は図 2.7 に示す。

2.8 XML 形式での入出力

編集結果を保存する形式として XML を採用した。これは、将来的に MPEG-7 への対応のしやすさと拡張性の高さを配慮したためである。XML の利点は第一章でも述べたが、拡張性の高さと機種やアプリケーションに依存しない形式、プレーンテキストであるので時代が経ってもデータとしては劣化しない永続性の高さがあげられ、今回みたいな多くの人で共有し残していく事が重要なアノテーションには最適なフォーマットである。欠点として、バイナリデータに比べてファイルサイズが大きくなってしまう事だが、1 年で 2 倍近く伸びるハードディスクの容量を考慮すれば、全くといっていいほど問題ではない。詳しいフォーマットは以下の形式である。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<VideoAnnotation version="0.1">
  <VideoHeader>
    <filepath>ローカルファイルのパス</filepath>
    <url>インターネット上コンテンツの URL</url>
    <filesize unit="byte">ファイルサイズ</filesize>
    <winsize width="352" height="240"/>
    <description>詳細情報の記述</description>
  </VideoHeader>
  <VideoTitleInfo>
```

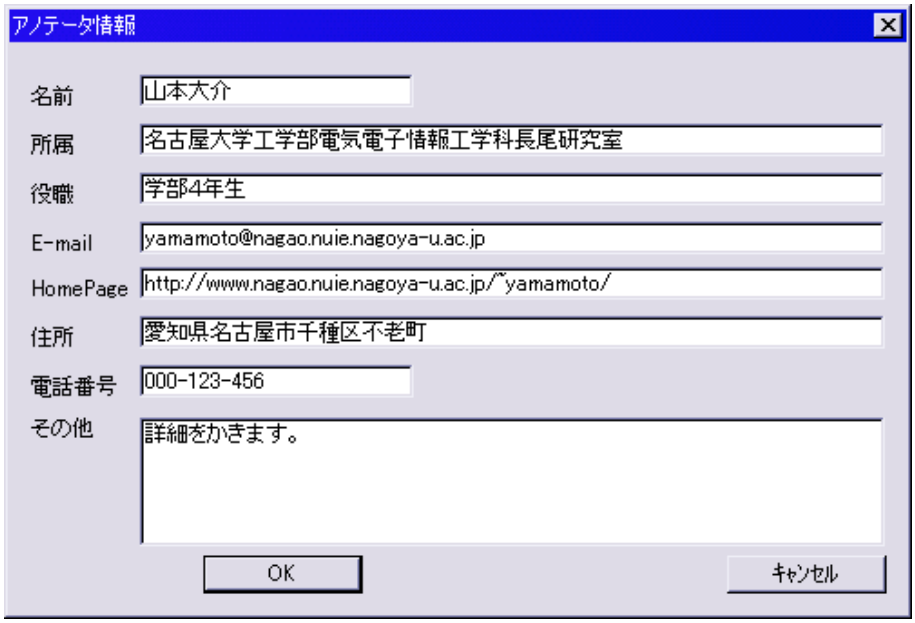

図 2.7: アノテータ情報ダイアログ

```
    <title>ビデオコンテンツのタイトル</title>
    <subtitle>サブタイトル</subtitle>
    <creator>製作者</creator>
    <copyright>著作権表示</copyright>
    <description>詳細情報の記述</description>
  </VideoTitleInfo>
  <AnnotatorInfo>
    <author>アノテーションをした人</author>
    <email>foo@hoge.com</email>
    <group>author の所属グループ</group>
    <url>annotator の HP</url>
    <address>住所</address>
    <tel>電話番号</tel>
    <post>ポスト</post>
    <description>詳細情報の記述</description>
  </AnnotatorInfo>
  <VideoTracks>
    <track stime="開始時間" etime="終了時間" keytime="キーとなるフレームの時
間">
      <histogram r="4" g="4" b="4"> // ヒストグラム情報
        <item r="0" g="0" b="0">0.1112</item>
        ... (item が r × g × b ×個存在)
      </histogram>
      <place>このシーンの場所</place>
```

```
      <annotation>
        <item category="カテゴリ意味属性" detail="詳細意味属性"/>
          ... (itemが複数存在)
      </annotation>
      <description>詳細情報の記述</description>
    </track>
     ... (trackが0以上複数存在)
  </VideoTracks>
  <Objects>
    <object stime="開始時間" etime="終了時間" keytime="キーの時間"> // object
以下は省略
      <histogram r="4" g="4" b="4"> // ヒストグラム情報
        <item r="0" g="0" b="0">0.1112</item>
        ... (itemがr×g×b×個存在)
      </histogram>
      <name>このオブジェクトの名前</name>
      <trackedRect left="左端" right="右端" top="上" bottom="下"/> // オブジ
ェクトの存在矩形範囲
      <annotation>
        <item category="カテゴリ意味属性" detail="詳細意味属性"/>
          ... (itemが複数存在)
      </annotation>
      <description>詳細情報の記述</description>
    </object>
     ... (objectが0以上複数存在)
  </Objects>
  <VoiceItems>
    <w in="開始時間" out="終了時間">ことば</w>
      ... (wタグが複数存在)
  </VoiceItems>
</VideoAnnotation>
```

リスト3　出力XMLのフォーマット

フォーマットは、大きく分けると、4つの項目になる。一つ目は、ヘッダ情報を示すタグの <VideoHeader> タグ。ファイルの存在する位置やファイルサイズ動画フォーマット等を記述している。二つ目は、タイトル情報を示す <VideoTitleInfo> タグ。ビデオのタイトルや著作権情報が記述されている。三つ目は、アノテーションした人の情報をあらわす <AnnotatorInfo> タグ。アノテーションした人の名前やメールアドレス等を記述する。四つ目は、シーン情報をあらわす <VideoTracks> タグ。カットごとにカットの開始時刻と終了時刻およびカラーヒストグラムに選択式アノテーション情報の意味属性等を列挙してい

ファイル名	画面解像度	時間	目視によるカット数
WebGrabber.mpg	320x240	1 分 20 秒	14
MapPlanner.mpg	352x240	45 秒	7
NewsPage.mpg	352x240	2 分 40 秒	11
Aspire.mpg	352x240	3 分 04 秒	28
LED.mpg	352x240	3 分 13 秒	58
VirtualWall.mpg	352x240	3 分 34 秒	24

表 2.1: カット検出に使用した動画ファイル

る。五つ目はオブジェクト情報を示す <Objects> タグ。オブジェクトの存在する開始時刻と終了時刻それにオブジェクトの存在矩形範囲、選択式アノテーションや記述情報を列挙している。六つ目は音声情報である。音声とその開始時間と終了時間を記述してある。

VideoTracks タグと Objects タグの中身を説明する。VideoTracks タグの中には tracks タグが存在する。ひとつの tracks タグでビデオコンテンツのひとつのカットからカットまでのまとまりを示す。Objetcts タグはビデオコンテンツの中に現れるオブジェクトを示す。書式としては以下のようになる。

2.9 アノテーションツールの評価

本ツールの評価であるが、アノテーションツールとして比較になるソフトを持ち合わせていないことと、使いやすさの評価は主観に頼る部分が多く、定量的な評価が難しい。が、カット検出等一部で定量的な評価ができる部分が存在するので、その部分の定量的評価を行ってみたい。

2.9.1 カット検出の評価

カット検出の精度について評価を行う。評価に利用するコンテンツは、56 個のニュース映像から無作為にとりだした 6 つのニュース映像を利用している。

実験方法は、本ツールを使ってカットの自動検出を行った。修正機能等は利用していない。正しく認識したカット数を正検出数、本来カットではない部分をカットであると認識した数は、誤検出数、本来カットである部分をカットであると認識した部分の数を、未検出数として実験を行った。閾値等のパラメータはすべての映像で同一にした。なお検出率は以下の式により計算した。

$$(\text{検出率}) = \frac{(\text{正検出数}) - (\text{誤検出数})}{(\text{正検出数}) + (\text{未検出数})}$$

全ての正検出数・誤検出数・未検出数をそれぞれ足したものでの検出率は 92.5

誤検出の例としては、図 2.8 のように同じカットではあるものの、信号の色が赤になる

ファイル名	正検出数	誤検出数	未検出数	検出率	
WebGrabber.mpg	14	0	0	100	
MapPlanner.mpg	7	0	0	100	
NewsPage.mpg	11	1	0	90.9	
Aspire.mpg	27	1	1	92.9	
LED.mpg	57	1	3	93.3	
VirtualWall.mpg	23	1	3	84.6	

表 2.2: カット検出の実験結果

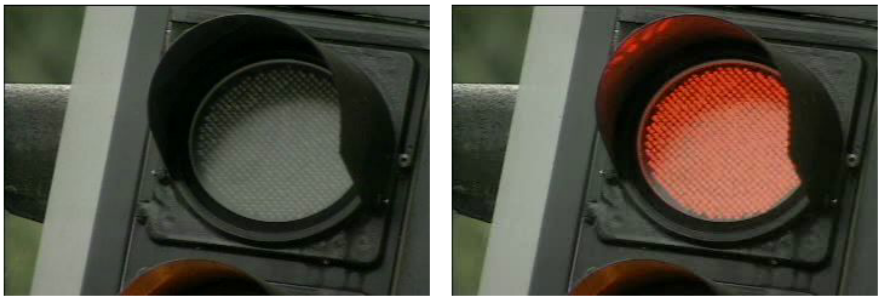

図 2.8: 誤検出例：信号の色が変わってる事により新たなカットであると誤検出している例

などの場合は、カットとして検出してしまう。未検出の例としては。図 2.9 のように、画面全体の色変化がおきないでカットが変わる場合などである。

誤検出や未検出の主要因は、画面全体のヒストグラムを求めているので局所的な変化に対応していない点である。単純なヒストグラムではなく位置情報も考慮した手法を考えなければならないと考えている。

2.9.2　ツールとしての評価

ツールの評価というものは、主観的な要因によるものが多く、数値で表すことが難しい。また、たとえユーザにとって、使いやすいツールだとしても、有用な情報がアノテーショ

図 2.9: 未検出例：映像が左から右へと徐々に変わっていく例

ン結果として反映されていなければ意味がなく、定量的な評価が難しい。そこで、ツールとしての定量的な評価は今後の課題としたい。

第3章　自然言語による意味的ビデオ検索

アノテーションされた動画コンテンツの応用例として、自然言語による検索を取り上げた。これは、アノテーションなしでの動画検索や、アノテーションが不完全な方式では十分な検索ができないと考えられる例であり、本ツールによるアノテーションの有効性を示す上でも、有効な例であると考えられるからである。自然言語による検索の上で重大な特徴は、動画の意味的内容を把握していなければ十分な検索ができない点である。たとえば、人の顔画像をキーとして動画像に全探索をかければ、人の顔を検索することは可能かもしれないが、その人が話しているのかどうかを全自動で解析するのは難しく、特定の条件下でしか検出することはできない。それは動画像から意味的内容を検出することは非常に困難だからである。しかしながら、意味的内容をアノテーションによって記述すれば容易に検索できる。一般に意味的内容を記述するのは困難を極めるが本アノテーションツールによりある程度簡単に記述できる。そこで、従来難しいとされている自然言語での動画像検索をWebブラウザを通して行う実験を試みた。

具体的なシステムとしては、ビデオアノテーションエディタによって作られたXMLアノテーションデータをXMLデータベースに登録し、一般的なWebブラウザを用いてビデオコンテンツを検索するシステム (図3.1) をJava Servletを用いて実現した。検索は、自然言語入力によって行っている。

3.1　使用したデータベースと開発環境

本研究では、XMLデータベースとして、Xindice [6] を用いた。XindiceはApacheによって作られたネイティブXMLデータベースであり、XPathを用いたデータベース検索ができるという特徴があり、MPEG-7や本アノテーションデータなど、XMLデータの蓄積には適したデータベースである。また、サーバサイドのプログラミング環境として、Sun Java Developer Kit 1.41を使用しApatchとTomcatを連携させたサーバでの検索をしている。

しかしながら、XindiceはJavaベースのデータベースであり検索・特に全文部分一致検索 (contains関数) が非常に遅いという欠点がある。そのために、なるべくcontains関数を使わない検索アルゴリズムを心がけているが、Xindiceを使う以上ある程度の遅さは我慢しなければならない。当研究室ではXindiceの高速化に関する研究も行っているが残念ながら、本研究によって作られたアノテーションデータには適用しても十分な高速化が図れないので採用は見送った。

3.2　検索アルゴリズム

検索は、本研究で作られたXMLデータをXindiceに登録したデータを元に検索している。データベースに登録されている情報としては、動画コンテンツに対するタイトル情報とアノテーション情報、カットとオブジェクトに対するカラーヒストグラム情報や選択式アノテーションによる意味属性情報とその意味属性の意味内容記述（ここではその意味属性が意味する言葉とその類義語の列挙）カットとオブジェクトの存在時間範囲とオブジェクトの存在位置、動画の音声部分の音声認識結果とその存在時間が登録されている。

本研究の特徴としては、人間が介在する意味属性の付与と、基本的に全自動解析されるヒストグラム情報やオブジェクトトラッキング結果、カット検出結果のみを用いて検索を行っており、人間と機械の両方のアノテーション結果をうまく使い分けている点である。

本研究では主に、マウスで選択するだけで簡単にアノテーションできる意味属性情報とコンピュータで自動抽出されるカラーヒストグラム情報と、オブジェクトおよびカットの存在時間範囲だけを利用して検索を行っている。音声部分の情報や、キーボードで情報を入力したものは利用しておらず、人間が介在する部分は意味属性の付与とオブジェクトの存在時間範囲の修正のみである。

この条件のもとに、以下のアルゴリズムを用いて検索を行った。

1. まず第一段階として、検索キーワードの形態素解析を行った。形態素解析には、茶筅[8]という形態素解析のソフトを用いて行っている。その茶筅から、動詞・形容詞・名詞・未知語をキーワードとして使用している。なお、未知語とは茶筅に登録されていない単語であり、名詞や英語などの場合が多い。今回は未知語を全て名詞と仮定して使用している。

2. 次に、形容詞・名詞から色や明暗を表す単語（たとえば、赤い・黒い・青い・暗い・明るい等）を色を表現する単語として認識した。色を表現する単語（便宜上、色名詞と呼ぶことにする）は、アノテーションデータの中に含まれるカラーヒストグラムとのマッチングを行うために使用するためである。

3. 検索アルゴリズムには、基本的にアノテーションデータと検索キーワードの形態素解析済み単語（名詞・形容詞・動詞・未知語）との完全一致した場合、そのオブジェクトやシーンに対して得点をつけ、得点が高い順に検索結果をソートする方式を取っている。アノテーションデータには、シーンやカット情報を表す「track」XMLツリーと、人や物などをあらわす「object」XMLツリーの集合が存在している。検索キーワードの中に場面やシーンを表す語、具体的には「場面」「光景」「風景」「シーン」「画面」「様子」等が含まれている場合はオブジェクト情報をあらわす<object>ツリーだけでなくシーン情報も表す<track>ツリー内も検索を検索を行い、特にこれらの言葉が含まれていない場合は、<object>だけを検索している。特に、場面を表す語で検索キーワードが終わっている場合は、<track>内を重点的に検索を行う。このとき、検索キーワードの中に色を表現する単語が含まれている場合は、カラーヒストグラムと色のデータから相関関数を用いて、その相関度に応じて得点を加算している。今回は、キーボードから入力したアノテーションデータは使用していない。

図 3.1: ビデオの検索画面例

なお、得点の重み付けであるが、ひとつのカット及びオブジェクトにおいて、キーワードがひとつヒットしたら１点とし、カラーヒストグラムによる得点の加算については、最大１点最小０点の実数値で加点したのち、さらに検索文で前の方のキーワードの重みづけを小さく、後ろのほうのキーワードの重み付けを大きくすることにした。本来ならば、構文解析を行い係り受けなどを解析して重み付けをする必要があるが、今回の単純な自然言語検索の場合は、「ウェブについて話している若い男」などのように、単純に後方の単語の方が重要であろうという前提で検索してもある程度の精度が得られると考えた。しかしながら、検索文を構文解析することは非常に重要であり、今後の課題としたい。

4. このようにして加点された得点に応じてソートし、順位づけをした上でユーザーにウェブブラウザを使って検索結果を示した。

3.3 評価

検索結果に対する評価を行う。本来ならば非常に多くの動画コンテンツに対してアノテーションを行う必要があるが、現段階では試作段階であり、少数のビデオコンテンツに対してのアノテーションを詳細に検討した方がよいと考えたため、３本の二分程度動画コンテ

ファイル名	画面解像度	時間	カット数	オブジェクトID数	シーンID数
WebGrabber.mpg	320x240	1分20秒	14	12	1
MapPlanner.mpg	352x240	45秒	7	7	4
NewsPage.mpg	352x240	2分40秒	11	9	6

表 3.1: 検索に用いたニュース映像

ンツだけにまとを絞ってアノテーションを行ってデータベースに登録し検索を行った。もちろん将来的には、もっと多くの動画コンテンツにアノテーションを行った後に検索システムの評価を行いと思っている。

3.3.1 データベースに登録されたコンテンツ

今回データベースに登録したビデオ映像は以下の三つであり、いずれもニュース映像である。

これら三つの映像をビデオアノテーションエディタによってアノテーションしたのちXindiceで登録し、検索実験を行った。

3.3.2 検索例「ウェブについて話している男」

この例では、形態素解析を行って、「ウェブ」・「話す」・「男」のキーワードを抜き出した。ウェブについてはウェブというアノテーションされた意味属性は存在しないので無視される。話すはspeakという意味属性が、男はmanという意味属性に変換されるので、それに応じて検索を行う。それとマッチしたオブジェクトを探しだしそのオブジェクトを順位づけして表示している。今回は、speakとmanの両方の意味属性がついたものがより上位にくる検索結果となっている。検索結果を図3.2に示す。

3.3.3 検索例「ウェブについて話している若い野郎」

この検索例では、形態素解析を行って、「ウェブ」・「話す」・「若い」・「野郎」とキーワードを抜き出している。ウェブはマッチする意味属性は存在しないが、話す・若い・野郎はspeak young manにそれぞれ対応している。前回の検索では意味属性のmanは男に対応していたが、今回は野郎に対応している。このように、ひとつの意味属性に複数の単語が結びついている。検索結果を図3.3に示す。

3.3.4 検索例「暗いシーンのテレビ」

この検索例の場合は、暗い・シーン・テレビという三つのキーワードからなっている。シーンの前に色や明度の情報をあらわす暗いが存在するので、まず画面のヒストグラムを

図 3.2: 検索例「ウェブについて話している男」

図 3.3: 検索例「ウェブについて話している若い野郎」

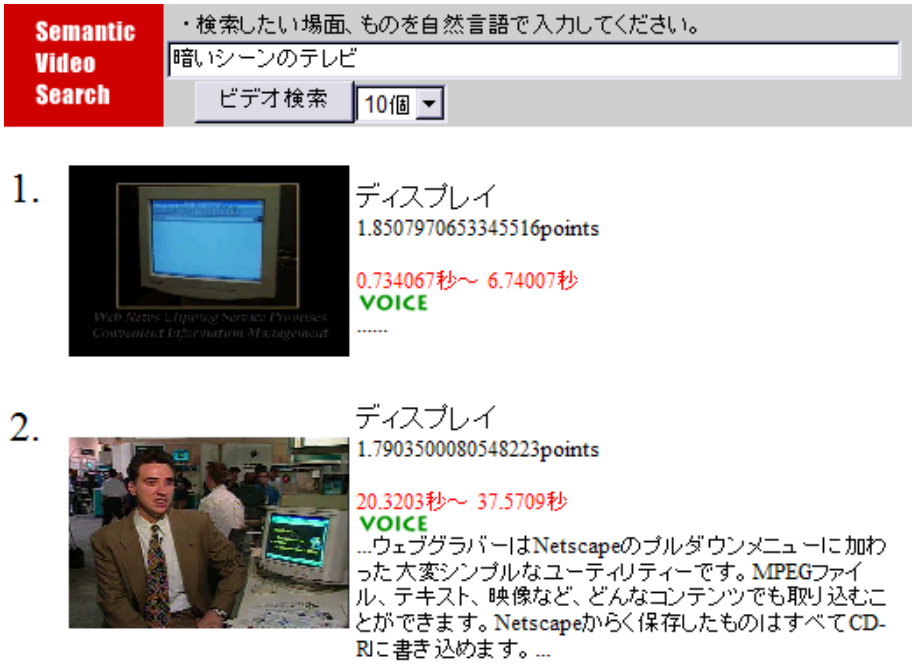

図 3.4: 検索例「暗いシーンのテレビ」

みて暗いシーンを探し、暗さの相関度をもとにして得点計算する。さらに、テレビは TV の意味属性に相当するので、TV の意味属性のオブジェクトを持つオブジェクトを探し出し、そのオブジェクトが登場するシーンの得点も加点して順位付けを行う。この場合は、より暗い場面のディスプレイが検索されている。検索結果は図 3.4 に示している。

3.3.5 まとめ

このように、検索したいオブジェクトやシーンに動作や状態・状況の意味属性が付与されていれば検索可能であることを示した。その意味属性は単純な完全一致ではなく、同義語にも柔軟に対応しており、たとえ同じ意味属性が同じだけ付いていたとしても、色情報などを基にしてソートしなおすこともできることを示した。このようにすれば、動画検索も十分実用的であると考えられる。

今回は、検索には音声部分の情報は一切使っていない。音声部分には検索に非常に役に立つ言語情報などがたくさん含まれているので、音声部分も検索に使用すればさらに高度な情報が得られ、よりよい検索ができると考えている。

第4章　関連研究

4.1　アノテーションに関する研究

4.1.1　Video Annotation Editor

本研究の元となった研究に長尾らが作成したVideo Annotation Editorがある[3]。これは、MPEGコンテンツに対しXML形式のアノテーションデータを付与するツールであり、ViaVoiceによる音声認識や、カット検出・オブジェクトトラッキングなどを行うことができるアノテーションツールである。

音声認識と画像認識をひとつの画面でひとつの時間軸を元に統合して処理できるのが特徴となっている。

4.1.2　MovieTool

これはリコー[10]が開発したMPEG-7記述用のビデオアノテーションツールである。MPEG-7を記述するために開発されたものであり、ビデオコンテンツに対し容易に階層構造表現も含めたMPEG-7データを記述することが可能である。

MovieToolの特長としては以下のとおりである。

- 映像を読み込むだけでMPEG-7記述が作成できる
- 映像の構造をビジュアルに作成でき、それをMPEG-7記述に自動反映する
- 構造化された映像の各シーンと、MPEG-7記述部分との対応をわかりやすく表示する
- MPEG-7記述の編集では、利用可能なタグ候補を表示することにより、文法に即したMPEG-7記述を簡単に作成できる
- MPEG-7の文法を定義したMPEG-7スキーマと、MPEG-7記述との整合性チェックを行える
- MPEG-7スキーマで定義されているすべてのメタデータを記述できる
- MPEG-7スキーマの動的読み込みにより、今後のスキーマの変更や、個別の拡張スキーマにも対応できる

4.1.3 VideoAnnEx Annotation Tool

IBM が作った MPEG-7 対応アノテーションツールである VideoAnnEx Annotation Tool [9] がある。これは、動画コンテンツに対して、Static Scene, Key Objects, Event の情報を各シーンに対して意味属性を振ることができるツールである。Windows 標準のツリービューを利用して、あらかじめ登録されている ID から複数選択したり、ID を増やしたりすることができるツールである。

ツリービューを利用することにより意味属性をカテゴリ分けできる利点がある一方、ツリーが大きくなるにつれどこにどの意味属性があるのか分かりにくく、また、ツリーの一部分を閉じてしまうと、そのツリー以下に意味属性のチェックが入っていても分かりにくいという弱点がある。

このソフトは、IBM から無料でダウンロードして実際に動かしてみることができる点が興味深い。

4.2 ビデオ検索に関する研究

4.2.1 MediaStream

MIT(マサチューセッツ工科大学) のメディアラボでの Daivs ら [12] によって開発された MediaStream は、アイコンによるビジュアル言語によって映像を構造化するようになっている。このツールの特徴は、３０００ものアイコンとその組み合わせによってビデオのオブジェクトにアノテーションすることができる点である。検索を考慮する場合、ビデオに注釈をつける場合自然言語でつけるよりもアイコンを使って意味属性をふった方が分かりやすいためにこの方式を利用したと考えられる。

本ツールの難点は、映像をアイコンで現すために、非常に多くのアイコンが必要となる点である。開発の最終段階にはアイコンの数が６０００個にもなったといい、アイコンは種類分けされてはいるもののアイコンの選択には慣れが必要であるし、そのアイコンの意味するものがアイコンだけでは分かりにくいという欠点があると考えられる。さらに、複雑なオブジェクトには複数のアイコンを組み合わせる必要があるので、複雑なオブジェクトを表現できる一方、煩雑な作業が必要になると考えられる。

4.2.2 時刻印付オーサリンググラフによるビデオ映像のシーン検索

一般に、時間とともに変化していく映像データを、意味属性だけで表現していくのは非常に困難な作業である。そこで、是津耕司ら [13] が開発した時点モデルによる映像区間の合成：時刻付きオーサリンググラフに関する研究を紹介では、各時点において、テキストにより映像のキーワードを記述する方式でアノテーションを行っている。テキストを記述するだけでは複雑な検索に不向きであるが、それぞれのキーワード間に互いに関係のあると思われるものに対して、時刻印付きオーサリンググラフという無向グラフを記述することによりその問題を解決している。さらに、その関連付けはテキストに含まれるキーワードの類似性を用いて自動的に関連付けを行っている。

このグラフを用いて、自然言語検索文による検索が行えるようになっている。検索アルゴリズムは自然言語検索文に含まれるキーワードとノードに含まれるキーワードの類似計算を行って関連のあるノードを見つける。そして、それらのノードにおいて時間的つながりを考慮しつつ、極小部分グラフを求める。この極小部分グラフが求める検索結果である。

映像に対し、無作為にコメントを自然言語で書いていくだけで検索に十分なアノテーション情報が追加できることが特徴である。

4.2.3 Informedia

画像認識、音声認識、自然言語処理の技術を統合して１０００時間ものビデオ映像の自動索引付けを行ったシステムとして、カーネギーメロン大学の Informedia プロジェクト [14] がある。クローズドキャプションの利用、文字・音声の自動認識を駆使してキーワード索引の自動生成を行い、キーワードにもとづくビデオ映像検索が可能になっている。また、tr-idf 法 (term frequency/inverse document frequency) を用いてビデオ映像の要約も実現している。

実際に１０００時間もの動画像に対して索引付けを行っており、非常に大規模なシステムでの自動認識と検索・要約を行っている点が興味深い。

第5章　おわりに

5.1　まとめ

今回は、ビデオコンテンツに対するアノテーションツールと、自然言語による検索ツールを試作した。従来難しいと思われていた自然言語によるビデオコンテンツ検索が、ヒストグラム等を基にした自動解析手法に加え動画像のオブジェクトやシーンの意味内容へのアノテーションを併用することにより比較的容易になることを示した。これにより、ユーザはGoogleと同様の感覚で動画像データを意味的に検索できるようになる。

5.2　今後の課題

本研究ではまだまだ、カット検出やオブジェクトトラッキング、それに検索部分など未熟な部分が多い。たとえば、フェードイン・フェードアウトしながらカットが変わる部分に対しての認識率が悪い。検索に関しても検索文の構文解析を行っていないなどの弱点がある。個々の要素の認識率や精度の向上にも勤めたいと考えている。

また、今回の研究ではビデオコンテンツを映像の立場からしか分析していない。それは、音声解析まで考慮すると非常にやらなければならない事が多くなり、帰って個々の研究が疎かになる恐れがあるためである。今回の研究では映像のみを利用してビデオコンテンツのアノテーションや検索をするという立場で研究を行った。もしも音声部分も統合的に可能であるのならばカット検出や検索などありとあらゆる部分で非常に多くの利益をもたらすと考えている。たとえば、カット検出の部分ならば映像では途切れていたとしても、音声で全く途切れていなければその二つの映像のカットは同じシーンであるという予想が立てられるし、検索時にも音声認識を利用しさえすれば、動画検索に役立つのはいうまでもない。音声部分に関しては今後の課題としたい。

さらに、今回は実験したデータが少なかったので、１００以上のビデオコンテンツに対してアノテーションを行い、データベースに登録することで、ある程度大規模なシステムに関しても有用であることを示す実験を行う予定である。

また、アノテーションツールに関しても、本当にいまのインタフェースが適しているのかも十分な議論ができていないのも事実である。本当にビデオアノテーションに適したインタフェースが存在する可能性もありツールの評価も含め、そのあたりの議論も今後できればよいと考えている。

謝辞

熱心で誠実な指導と研究のあり方を教えていただいた末永先生と、ミーティングで的確な意見をいただいた末永研究室の諸先輩方、さらに研究室の枠を超えて指導していただいた長尾先生と長尾研究室の皆様に心から感謝します。

本研究を進めるにあたり、指導教員である末永康仁教授、森健策助教授には研究の心構え等基礎的な考え方から、ミーティング等を通して貴重なご意見を多数いただき様々な面でお世話になりました。さらに、長尾確教授にはミーティングやディスカッション・論文指導にコンセプトメイキング等研究室の枠を超えて、様々なご指導を賜り大変お世話になりました。

末永研究室のテーマ企画やその他論文の調べ方や研究のやり方を教えていただいた豊住健一先輩・藍口孝之先輩・山田直也先輩をはじめとした末永研究室の諸先輩方にも大変お世話になりました。

さらに、長尾研究室の梶克彦先輩・山根隼人先輩には論文の書き方や研究の仕方・ミーティングでの助言はもとより、研究生活面でも生活環境の充実や行事等を通して楽しい研究室作りをもしていただき、大変お世話になりました。特に山根先輩には飲み会や論文の書き方・研究の仕方を、梶先輩には ShopNavi などを通して研究室環境の充実を図っていただいています。

研究室環境の充実といえば、長尾研究室秘書の兼松英代さんには通常の秘書業務に加え、英語の添削や、コーヒーやお菓子の差し入れ、掃除やその他の細やかな心配りをしていただき非常に感謝しています。

慶応大学の福岡俊樹様には論文の謝辞の書き方を教えていただきお世話になりました。

そして、困った時に助け合ったり遊んだりした末永研究室のＢ４である、池崎正和君・澤田大輔君・根木大輔君・宮本秀昭君・脇田悠樹君にも研究面や生活面にも非常にお世話になりました。

また、長尾研究室のＢ４の友人の皆様にも非常にお世話になりました。松浦真治君にはサーバや Java 関連ではいろいろと質問ばかりして非常にお世話をかけました。さらに、私生活面でもとあるプロジェクトを通して非常に多くの迷惑をかけてしまい申し訳なく思っています。清水敏之君にはことあるごとに研究や私生活面で相談に乗ってもらい非常に感謝しています。加藤範彦君にも彼のあふれるアイデアや新しい技術を積極的に紹介してもらい研究を通して非常に多くの刺激をいただきました。細野祥代さんには、研究に関することだけでなく、研究室旅行や研究室生活環境において充実を図っていただき感謝しています。

最後に、拙いながらも大学で論文を書けるまでに育てていただいた両親にも最大限の感謝の気持ちを表します。

ここに書ききれなかった人たちも含め様々な人たちのおかげで今の自分があり、この論

文を書く事ができたと思っています。
　ありがとうございました。

関連図書

[1] W3C. Extensible Markup Language (XML). http://www.w3.org/XML/. 1996.

[2] Katashi Nagao,Yoshinari Shirai,Kevin Squire. Semantic Annotation and Transcoding: Making Web Content More Accessible. IEEE MultiMedia,Vol.8,No.2,pp69-81. 2001.

[3] Katashi Nagao,Shigeki Ohira,Mitsuhiro Yoneoka. Annotation-based multimedia summarization and translation. In Proceedings of the Nineteenth International Conference on Computational Linguistics (COLING-2002). 2002.

[4] MPEG. MPEG-7. http://ipsi.fraunhofer.de/delite/Projects/MPEG7/. 2002.

[5] W3C. Resource Description Framework (RDF). http://www.w3.org/RDF/. 2001.

[6] The Apache Software Foundation. Apache Xindice. http://xml.apache.org/xindice/. 2001.

[7] 西尾章治朗, 田中克巳, 上原邦昭, 有木康雄, 加藤俊一, 河野浩之. 情報の構造化と検索. 2002.

[8] 奈良先端科学技術大学院大学自然言語処理学講座. 形態素解析システム 茶筌. http://chasen.aist-nara.ac.jp/. 1997.

[9] IBM. VideoAnnEx Annotation Tool. http://www.alphaworks.ibm.com/tech/videoannex. 2002.

[10] Richo. Ricoh MovieTool. http://www.ricoh.co.jp/src/multimedia/MovieTool/. 2002.

[11] Canon. MPEG-7 Audio 'Spoken Content'. http://www.cre.canon.co.uk/mpeg7asr/. 2002.

[12] Davis, M.. An Iconic Visual Language for Video Annotation. Proceedings of IEEE Symposium on Visual Language. pp.196-202, 1993.

[13] 是津耕治, 上原邦昭, 田中克己. 時刻印付オーサリンググラフによるビデオ映像のシーン検索. 情報学会論文誌,Vol.39,No.4,pp.923-932. 1998.

[14] Wactlar, H. D., Kanade, T.,Smith, M.A. and Stevens, S.M.. Intelligent Access to Digital Video: Informedia Project. IEEE Computer,Vol.29,No.5,pp140-151. 1996.

[15] Google. http://www.google.com/.

第6章　付録

6.1　実験で用いたobjectDefinitions.xmlの全リスト

```
<?xml version="1.0" encoding="Shift_JIS"?>
<objectDefinitions version="0.1">
  <category id="stasis" jp="静止">
    <item id="sitting" jp="座る">
      <synonym word="座る"/>
      <synonym word="すわる"/>
      <synonym word="腰掛ける"/>
      <synonym word="sit"/>
      <synonym word="sits"/>
    </item>
    <item id="standing" jp="立つ">
      <synonym word="立つ"/>
      <synonym word="stand"/>
      <synonym word="stands"/>
      <synonym word="standed"/>
    </item>
    <item id="sleeping" jp="寝る">
      <synonym word="寝る"/>
      <synonym word="横になる"/>
      <synonym word="lay"/>
      <synonym word="sleep"/>
      <synonym word="sleeps"/>
    </item>
    <item id="other" jp="その他">
    </item>
  </category>
  <category id="move" jp="移動">
    <item id="walk" jp="歩く">
      <synonym word="歩く"/>
      <synonym word="歩行"/>
      <synonym word="walk"/>
      <synonym word="walked"/>
```

```
      <synonym word="walks"/>
    </item>
    <item id="run" jp="走る">
      <synonym word="走る"/>
      <synonym word="駆ける"/>
      <synonym word="run"/>
      <synonym word="runs"/>
    </item>
    <item id="fall" jp="落ちる">
      <synonym word="落ちる"/>
      <synonym word="落下"/>
      <synonym word="fall"/>
      <synonym word="falls"/>
    </item>
    <item id="float" jp="浮く">
      <synonym word="浮く"/>
      <synonym word="ぷかぷか"/>
    </item>
    <item id="swim" jp="泳ぐ">
      <synonym word="泳ぐ"/>
      <synonym word="およぐ"/>
      <synonym word="swim"/>
      <synonym word="swims"/>
      <synonym word="swimed"/>
    </item>
    <item id="fly" jp="飛ぶ">
      <synonym word="飛ぶ"/>
      <synonym word="飛行"/>
      <synonym word="ジャンプ"/>
      <synonym word="jump"/>
      <synonym word="jumpes"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="direction" jp="方向">
    <item id="left" jp="左">
      <synonym word="左"/>
      <synonym word="ひだり"/>
    </item>
```

```
    <item id="right" jp="右">
      <synonym word="右"/>
      <synonym word="みぎ"/>
    </item>
    <item id="up" jp="上">
      <synonym word="上"/>
      <synonym word="うえ"/>
    </item>
    <item id="down" jp="下">
      <synonym word="下"/>
      <synonym word="した"/>
    </item>
    <item id="foward" jp="前">
      <synonym word="前"/>
    </item>
    <item id="backward" jp="後ろ">
      <synonym word="後ろ"/>
    </item>
    <item id="random" jp="うろちょろ">
      <synonym word="うろちょろ"/>
    </item>
    <item id="inside" jp="中">
      <synonym word="中"/>
      <synonym word="内部"/>
      <synonym word="真ん中"/>
      <synonym word="内側"/>
    </item>
    <item id="outside" jp="外">
      <synonym word="外"/>
      <synonym word="外部"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="sound" jp="音を出す">
    <item id="speak" jp="話す">
      <synonym word="話す"/>
      <synonym word="しゃべる"/>
      <synonym word="語る"/>
      <synonym word="演説"/>
```

```
      <synonym word="speak"/>
      <synonym word="speaks"/>
      <synonym word="speaked"/>
    </item>
    <item id="roar" jp="吼える">
      <synonym word="吼える"/>
      <synonym word="わめく"/>
      <synonym word="叫ぶ"/>
    </item>
    <item id="cry" jp="泣く">
      <synonym word="泣く"/>
      <synonym word="涙"/>
    </item>
    <item id="sing" jp="歌う">
      <synonym word="歌う"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="behavior" jp="振る舞う">
    <item id="eat" jp="食べる">
      <synonym word="食べる"/>
    </item>
    <item id="drink" jp="飲む">
      <synonym word="飲む"/>
    </item>
    <item id="write" jp="描く">
      <synonym word="描く"/>
      <synonym word="かく"/>
      <synonym word="お絵かき"/>
    </item>
    <item id="read" jp="読む">
      <synonym word="読む"/>
      <synonym word="読書"/>
    </item>
    <item id="cut" jp="切る">
      <synonym word="切る"/>
      <synonym word="切断"/>
      <synonym word="削除"/>
    </item>
```

```
    <item id="hit" jp="たたく">
      <synonym word="叩く"/>
      <synonym word="たたく"/>
    </item>
    <item id="buy" jp="買う">
      <synonym word="買う"/>
      <synonym word="購入"/>
    </item>
    <item id="sell" jp="売る">
      <synonym word="売る"/>
      <synonym word="売却"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="human" jp="人">
    <item id="male" jp="男">
      <synonym word="男"/>
      <synonym word="男性"/>
      <synonym word="野郎"/>
      <synonym word="man"/>
      <synonym word="human"/>
      <synonym word="male"/>
      <synonym word="men"/>
      <synonym word="Man"/>
      <synonym word="Human"/>
    </item>
    <item id="female" jp="女">
      <synonym word="女"/>
      <synonym word="女性"/>
      <synonym word="少女"/>
    </item>
    <item id="baby" jp="赤ちゃん">
      <synonym word="赤ちゃん"/>
    </item>
    <item id="child" jp="子供">
      <synonym word="子供"/>
      <synonym word="子"/>
    </item>
    <item id="young" jp="若者">
```

```
      <synonym word="若者"/>
      <synonym word="若い"/>
      <synonym word="young"/>
      <synonym word="Young"/>
    </item>
    <item id="adult" jp="大人">
      <synonym word="大人"/>
      <synonym word="adult"/>
    </item>
    <item id="elder" jp="老人">
      <synonym word="老人"/>
      <synonym word="おじいさん"/>
      <synonym word="おばあさん"/>
    </item>
    <item id="dead" jp="死亡">
      <synonym word="死亡"/>
      <synonym word="死体"/>
    </item>
  </category>
  <category id="animal" jp="動物">
    <item id="dog" jp="犬">
      <synonym word="犬"/>
      <synonym word="いぬ"/>
    </item>
    <item id="cat" jp="猫">
      <synonym word="猫"/>
      <synonym word="ねこ"/>
    </item>
    <item id="bird" jp="鳥">
      <synonym word="鳥"/>
      <synonym word="とり"/>
    </item>
    <item id="bug" jp="虫">
      <synonym word="虫"/>
      <synonym word="昆虫"/>
    </item>
    <item id="fish" jp="魚">
      <synonym word="魚"/>
      <synonym word="魚介類"/>
    </item>
    <item id="other" jp="その他">
```

```
      <synonym word=""/>
    </item>
  </category>
  <category id="plant" jp="植物">
    <item id="tree" jp="木">
      <synonym word="木"/>
    </item>
    <item id="flower" jp="花">
      <synonym word="花"/>
      <synonym word="フラワー"/>
    </item>
    <item id="bush" jp="茂み">
      <synonym word="茂み"/>
      <synonym word="草むら"/>
    </item>
    <item id="leaf" jp="葉">
      <synonym word="葉"/>
      <synonym word="葉っぱ"/>
    </item>
    <item id="seed" jp="種">
      <synonym word="種"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="food" jp="食べ物">
    <item id="noodle" jp="麺">
      <synonym word="麺"/>
    </item>
    <item id="rice" jp="ご飯">
      <synonym word="ご飯"/>
      <synonym word="飯"/>
    </item>
    <item id="soup" jp="スープ">
      <synonym word="スープ"/>
      <synonym word="汁"/>
    </item>
    <item id="meat" jp="肉">
      <synonym word="肉"/>
    </item>
```

```
    <item id="vegetable" jp="野菜">
      <synonym word="野菜"/>
      <synonym word="菜っ葉"/>
    </item>
    <item id="fish" jp="魚">
      <synonym word="魚"/>
    </item>
    <item id="juice" jp="ジュース">
      <synonym word="ジュース"/>
      <synonym word="清涼飲料水"/>
    </item>
    <item id="alcohol" jp="アルコール">
      <synonym word="アルコール"/>
      <synonym word="お酒"/>
      <synonym word="酒"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="hardware" jp="機械">
    <item id="car" jp="車">
      <synonym word="車"/>
      <synonym word="自動車"/>
    </item>
    <item id="TV" jp="テレビ">
      <synonym word="テレビ"/>
      <synonym word="ディスプレイ"/>
      <synonym word="TV"/>
      <synonym word="display"/>
      <synonym word="Display"/>
    </item>
    <item id="computer" jp="コンピュータ">
      <synonym word="コンピュータ"/>
      <synonym word="計算機"/>
      <synonym word="Computer"/>
      <synonym word="computer"/>
    </item>
    <item id="clock" jp="時計">
      <synonym word="時計"/>
    </item>
```

```
    <item id="fan" jp="フン">
      <synonym word="ファン"/>
      <synonym word="扇風機"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="shape" jp="形">
    <item id="circle" jp="円">
      <synonym word="円"/>
      <synonym word="丸"/>
      <synonym word="まる"/>
    </item>
    <item id="square" jp="四角">
      <synonym word="四角"/>
      <synonym word="正方形"/>
      <synonym word="長方形"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="image" jp="絵">
    <item id="image" jp="絵">
      <synonym word="絵"/>
      <synonym word="image"/>
      <synonym word="picture"/>
      <synonym word="Image"/>
    </item>
    <item id="picture" jp="写真">
      <synonym word="写真"/>
      <synonym word="image"/>
      <synonym word="picture"/>
      <synonym word="Picture"/>
    </item>
    <item id="movie" jp="動画">
      <synonym word="動画"/>
      <synonym word="ビデオ"/>
      <synonym word="ムービー"/>
      <synonym word="movie"/>
```

```
      <synonym word="video"/>
      <synonym word="Movie"/>
      <synonym word="Video"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="icon" jp="アイコン">
    <item id="logo" jp="ロゴ">
      <synonym word="ロゴ"/>
      <synonym word="logo"/>
      <synonym word="Logo"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="string" jp="文字">
    <item id="title" jp="タイトル">
      <synonym word="タイトル"/>
      <synonym word="題名"/>
      <synonym word="題"/>
      <synonym word="title"/>
      <synonym word="Title"/>
    </item>
    <item id="name" jp="名前">
      <synonym word="名前"/>
      <synonym word="苗字"/>
      <synonym word="name"/>
      <synonym word="Name"/>
    </item>
    <item id="display" jp="表示">
      <synonym word="表示"/>
      <synonym word="ディスプレイ"/>
      <synonym word="display"/>
      <synonym word="Display"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
```

```
  </category>
</objectDefinitionss>
```

6.2 実験で用いたsceneDefinitions.xmlの全リスト

```
<?xml version="1.0" encoding="Shift_JIS"?>
<sceneDefinitions version="0.1">
  <category id="place" jp="場所">
    <item id="office" jp="オフィス">
      <synonym word="オフィス"/>
      <synonym word="事務所"/>
      <synonym word="会社"/>
    </item>
    <item id="out" jp="外">
      <synonym word="外"/>
      <synonym word="外部"/>
      <synonym word="道端"/>
    </item>
    <item id="home" jp="家">
      <synonym word="家"/>
    </item>
    <item id="road" jp="道">
      <synonym word="道"/>
      <synonym word="道路"/>
      <synonym word="道端"/>
    </item>
    <item id="sea" jp="海">
      <synonym word="海"/>
      <synonym word="海洋"/>
      <synonym word="海辺"/>
    </item>
    <item id="sky" jp="空">
      <synonym word="空"/>
      <synonym word="そら"/>
    </item>
    <item id="monitor" jp="画面">
      <synonym word="画面"/>
      <synonym word="ディスプレイ"/>
      <synonym word="表示"/>
    </item>
```

```
    <item id="web" jp="ウェブページ">
      <synonym word="インターネット"/>
      <synonym word="ウェブページ"/>
      <synonym word="ページ"/>
    </item>
    <item id="chaos" jp="混沌">
      <synonym word="混沌"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="weather" jp="天気">
    <item id="fine" jp="晴れ">
      <synonym word="晴れ"/>
    </item>
    <item id="cloud" jp="曇り">
      <synonym word="曇り"/>
    </item>
    <item id="rain" jp="雨">
      <synonym word="雨"/>
    </item>
    <item id="wind" jp="風">
      <synonym word="風"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
  <category id="category" jp="カテゴリ">
    <item id="title" jp="タイトル">
      <synonym word="タイトル"/>
    </item>
    <item id="opening" jp="オープニング">
      <synonym word="オープニング"/>
    </item>
    <item id="ending" jp="エンディング">
      <synonym word="エンディング"/>
    </item>
  </category>
  <category id="mood" jp="雰囲気">
```

```
    <item id="fine" jp="良い">
      <synonym word="良い"/>
    </item>
    <item id="heavy" jp="重い">
      <synonym word="重い"/>
    </item>
    <item id="other" jp="その他">
      <synonym word=""/>
    </item>
  </category>
</sceneDefinitions>
```