

第4章 3次元表示の動画検索への応用

本章では、本研究で開発した、YouTube の API を利用することで得られる XML 形式のメタデータを 3次元空間で可視化したシステムについての説明を行う。

4.1. 概要

現在、YouTube と呼ばれるインターネットで動画共有サービスを提供するサイトが注目を集めている。YouTube は 2005 年 12 月に公式にサービスを開始して以来、急速に利用者を増やし、2007 年 5 月 21 日時点で約 8000 万の動画があり、1 日に 3 万 5 千の動画がアップロードされているとプレスリリースでも発表されている²⁴⁾。

まず、現行の YouTube サイトでの一般的な検索プロセスや、用いられている技術として代表的な Ajax、そして最後に YouTube が提供する API に関する説明を行う。

検索プロセス

- 1、トップページの検索フォームから特定のクエリを入力する。
- 2、クエリに応じた動画が関連度の高い順に一覧表示される。
- 3、動画のタイトル、詳細説明、画像、再生時間、等を加味し、見たい動画をクリックする。
- 4、ページがリロードされ、動画再生ページに飛ぶ。

図 4.1 に検索結果例を示す。

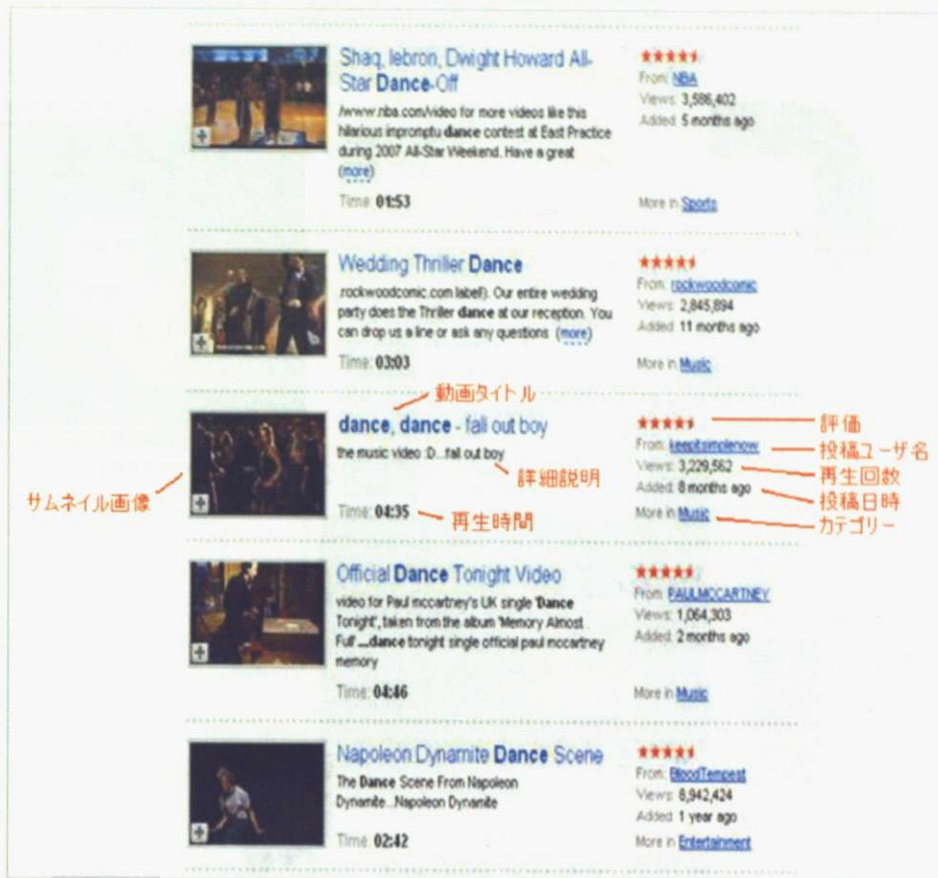


図 4.1 YouTube による検索結果表示例

この例からもわかるように、1つの検索結果に対し非常に多様なデータが付与されている。ただし、情報提示の仕方は各データによって異なる。動画タイトルやサムネイル画像といった重要な情報は大きく表示され、その他のサブ的な情報は小さく表示されている。

Ajax

YouTube では代表的な技術として Ajax が用いられている。Ajax とは 'Asynchronous JavaScript + XML' の略で、Web ブラウザがもつ JavaScript の機能を使ってサーバーと非同期通信を行うことで対話的な Web アプリケーションを実装するための技術の総称である。従来、Web ブラウザを使った Web アプリケーションでは、データをサーバーに通知して処理結果を得るにはページ全体をロードしなればいけなかった。Ajax では、ユーザーの操作や画面描画などと並行してサーバーと非同期に通信を行うことで、サーバーの存在を感じさせないシームレスな Web アプリケーションを実現することができる。図 4.2 に実際に YouTube で使われている Ajax の使用例を示す。



図 4.2 YouTube での Ajax 使用例

この例で示すように、ページを遷移させることなく同一ページ内の要素を動的に変化させている。YouTube ではこの他にも、動画の詳細情報が長すぎる場合にはデフォルトでその一部だけを表示させるようになっており、さらに詳細情報を確認したい場合には、青色で示される(more)という文字をクリックすれば、ページの一部分だけを書き換えてその情報が表示されるようになっている。この技術を用いることによって、ユーザーが見たい時に見たい情報だけを表示させることに成功している。

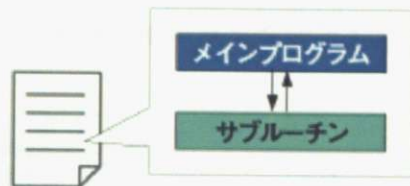
YouTube API

まず、序論で述べた Web API についてももう少し詳しく説明を述べた上で、実際の YouTube における API 使用例を示す。

序論では公開されている Web API の紹介をしたが、結局 Web API が何をしているかという、「HTTP を利用してネットワーク越しに処理を実行して結果を受け取る。」ということをやっているのである。HTTP とはブラウザとサーバーの間で HTML (もしくはウェブ文書) を送受信するためのプロトコルである。

例えば、Perl ではサブルーチンを作成してプログラム内のほかの場所から呼び出すことができる。Web API はこれと同様に、ネットワーク越しにほかのコンピューター上にあるサブルーチンを実行して、その結果を受け取ることができるものである。図 4.3 はそのイメージ図である。

一般的なプログラム



Web APIを利用したプログラム

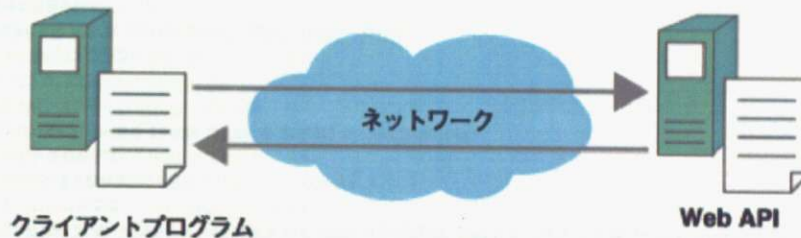


図 4.3 Web API のイメージ図

例えば Google や Yahoo では検索フォームから検索キーワードを送ることで検索結果を受け取ることができる。Amazon では販売されている商品情報を検索したり、ショッピングカートを利用できる仕組みがある。これらのサイトではそういった機能を API として公開しており、自分のプログラムからその API を呼び出すことで受け取ったデータを利用することができる。このように Web API には、自分自身のシステム内だけでなく、インターネット越しに外部サービスが提供する API を呼び出して、その処理結果やデータを活用することができるという特徴がある。

また、Web API では HTTP が利用されるというのも大きな特徴と言える。通常の Web ページなどへのアクセスと同様に URL でアクセス先を指定してデータを送信し、結果を受け取ることができる。

そして、受け取るデータ形式として最も一般的なのが XML である。XML とはデータ記述言語の 1 つで、要素とタグによって記述される。ユーザーは自由に要素を特徴づけるタグを定義することができ、文章中に含まれるデータに意味づけができる、というのが XML の最大の特徴である。

YouTube においても API を利用して得られるデータ形式に XML を使用しており、以下のような URL をブラウザから送信することで、

```
http://www.youtube.com/api2_rest?method=youtube.videos.list_by_tag&dev_id=*****  
&tag=Japan&page=1&per_page=50
```

図 4.4 のような XML 形式のデータを受け取ることができる。

```
<?xml version="1.0" encoding="utf-8" ?>  
- <ut_response status="ok">  
- <video_list>  
  <total>202591</total>  
  - <video>  
    <author>nealon14</author>  
    <id>eslA4xU9bNk</id>  
    <title>X-Japan - I.V (Full Song DVD Rip)</title>  
    <length_seconds>297</length_seconds>  
    <rating_avg>4.91</rating_avg>  
    <rating_count>134</rating_count>  
    <description>Full song from Saw 4 DVD!</description>  
    <view_count>60572</view_count>  
    <upload_time>1199884656</upload_time>  
    <comment_count>73</comment_count>  
    <tags>X-Japan I.V. xjapan rock saw4 saw thebigjapan 192 kbps lv i.v nealon24 dvd</tags>  
    <url>http://www.youtube.com/?v=eslA4xU9bNk</url>  
    <thumbnail_url>http://img.youtube.com/vi/eslA4xU9bNk/default.jpg</thumbnail_url>  
    <embed_status>ok</embed_status>  
  </video>  
  - <video>  
    <author>cooptvberlin</author>  
    <id>AH1UA3ljXug</id>  
    <title>9/11 Japan Parliament 1/10/2008 Pt 1/8 OLD VERSION subs</title>  
    <length_seconds>286</length_seconds>  
    <rating_avg>4.86</rating_avg>  
    <rating_count>155</rating_count>  
    <description>WATCH UPDATE http://www.youtube.com/watch?v=Oq4_07FmCyA "The 'War on  
    Terror' started with the attacks of 9/11." is incorrect and has to be "And as we all know, the  
    starting point [or premise] of this bill is 9/11" Video covers: War on Terror - who is the enemy?
```

図 4.4 YouTube API による XML データ

この図からもわかるように、各要素はその要素を特徴づけるタグによって挟まれているので何を意味するデータなのかがわかる。例えば図中の<title>タグで挟まれている要素は動画のタイトルを表すものであるということが容易に想像できる。

4.2. 問題点

以下、YouTube における問題点を記述する。

通常、動画・音楽・Web サイト等の検索を行うインターフェースとして、検索クエリを入力するフォームが用意されている。このフォームから目的の情報を探し出すのに適切と思われるクエリをユーザーが入力し、検索結果を吟味する。そもそもこの検索クエリを入力する段階として、ユーザーが自分の探したいものが決まっており、それに対する情報が明確にわかっている場合と、ユーザーが自分の探したいものが具体的に決まっておらず、漠然とこんな感じの動画を探したいという目的だけもっている場合とがある。前者を検索パターン1、後者を検索パターン2と名付ける。

検索パターン1：探したいものが決まっている

検索パターン2：探したいものが決まっていない

検索パターン1の場合は、その動画に関する複数のキーワードで絞り込み、少数の検索結果を吟味すればよい。また、検索結果が多い場合でも、コンテンツの内容を逐一確認せずとも、ディスプレイに表示されたメタデータを流し読みするだけで目的のコンテンツにたどり着くことができる。例えば、〇〇というアーティストの〇〇というタイトルのPV動画が見たいとする。そのような場合、検索クエリとしてアーティスト名とタイトルを同時に入力するのが普通であるし、仮にアーティスト名しか正確に思い出せない場合でも、検索結果として表示されるタイトル名だけを確認して動画を探せばよい。

検索パターン2の場合では、漠然としたキーワードで検索するしかないため、返ってくる検索結果も必然的に多くなる。この中から目的のコンテンツを探し出すためには、スクロールやページ切り替えを何度も行ったり、コンテンツに逐一アクセスしてその内容を確認する作業が必要になる。YouTube のサイトのように多様な情報が混在する場合には一画面内に表示される検索結果も少なく、ユーザーが満足するコンテンツを探し出すには長い時間がかかってしまう。例えば、漠然と何かサッカーに関する面白い動画を発見したいとする。そのような場合は、検索クエリとしてサッカーとだけ入力するのが普通である。しかし、サッカーという検索クエリは非常に幅の広いワードであり、検索結果はその分多くなる。YouTube では検索クエリとの関連度が高い順に、検索結果が縦一列に表示されているだけであり、検索結果に含まれるサムネイル画像・動画タイトル・詳細説明・再生時間等を総合的に判断して、見る価値があるかどうかを判断しなければならない。このような場合は何か動画情報の方からおもしろい動画であることを語りかけてくれるような仕組み

が必要と言える。

本章では、上記した検索パターン2における問題点を解決すべく、YouTube の API を利用することで得られる多次元の XML データを3次元表示で可視化するシステムの開発を行った。その際、処理結果をリアルタイムで3D化できるように、第3章で用いた Web3D の1つである Viewpoint で3D オブジェクトを記述した。

4.3. 目的

第3章で述べた音楽情報可視化システムの開発を通し、3次元表示による利点として以下の2点があげられた。

- ・ 2次元表示よりも多くのデータを一度に表示できる
- ・ 複数の軸を設定できることにより多様な属性を表現できる

また XML という複雑な情報を可視化するにあたり、これら3次元表示の特性を最大限に生かすためには以下の2点にも考慮する必要があることがわかった。

- ・ データのオーバーラップの解消、かつ、仮想3次元表示における空間把握の実現方法として、アニメーションの効果的な利用を考える。
- ・ 文字情報のような詳細情報を常に確認することができ、かつ、情報の特性をサブ的に表現できるような手法を考える。

本章ではこれらの要素を実現したシステムが果たして XML 情報の可視化に本当に有効であるのかどうかを検証することが目的であり、そのために以下のようなシステムを開発した。

YouTube の API を利用し、動画に含まれる複雑な XML 形式のメタデータを仮想3次元空間で効率よく可視化することのできるシステム。

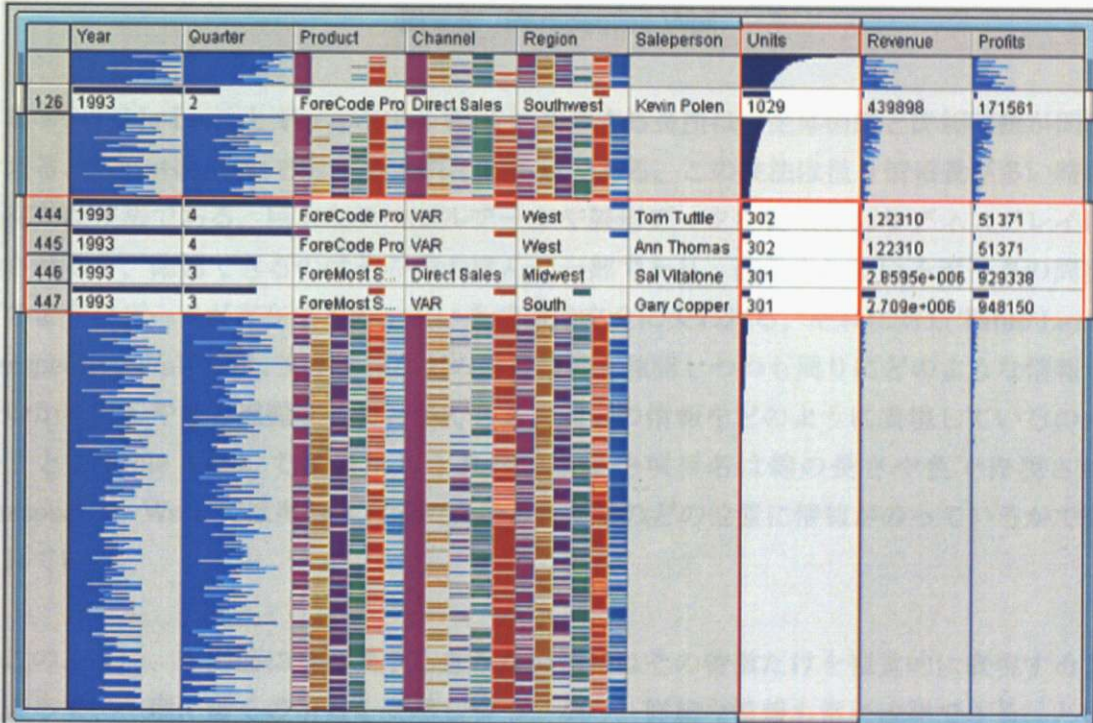
4.4. 設計コンセプト

本章で開発したシステムは、先行研究でも述べた Table Lens と Perspective Wall の長所を組み合わせで設計されている。

まず、Table Lens と Perspective Wall の長所・特徴をそれぞれ簡潔に述べる

Table Lens

- ・ 表形式で管理されるような多次元情報を 2 次元平面で可視化したシステム
- ・ 全体と詳細が同時に確認できる
- ・ 表示しきれない数値や項目名を線の長さや色で表現



	Year	Quarter	Product	Channel	Region	Saleperson	Units	Revenue	Profits
126	1993	2	ForeCode Pro	Direct Sales	Southwest	Kevin Polen	1029	439898	171561
444	1993	4	ForeCode Pro	VAR	West	Tom Tuttle	302	122310	51371
445	1993	4	ForeCode Pro	VAR	West	Ann Thomas	302	122310	61371
446	1993	3	ForeMost S...	Direct Sales	Midwest	Sal Vitatone	301	2.8595e+006	929338
447	1993	3	ForeMost S...	VAR	South	Gary Copper	301	2.709e+006	948150

図 4.5 Table Lens

Perspective Wall

- ・ 時系列データのような線形情報を 3 次元表示を用いて可視化したシステム
- ・ 全体と詳細が同時に確認できる
- ・ 詳細を確認したい部分は壁の中央に表示され、その他の情報は壁の奥行き方向に表示される。

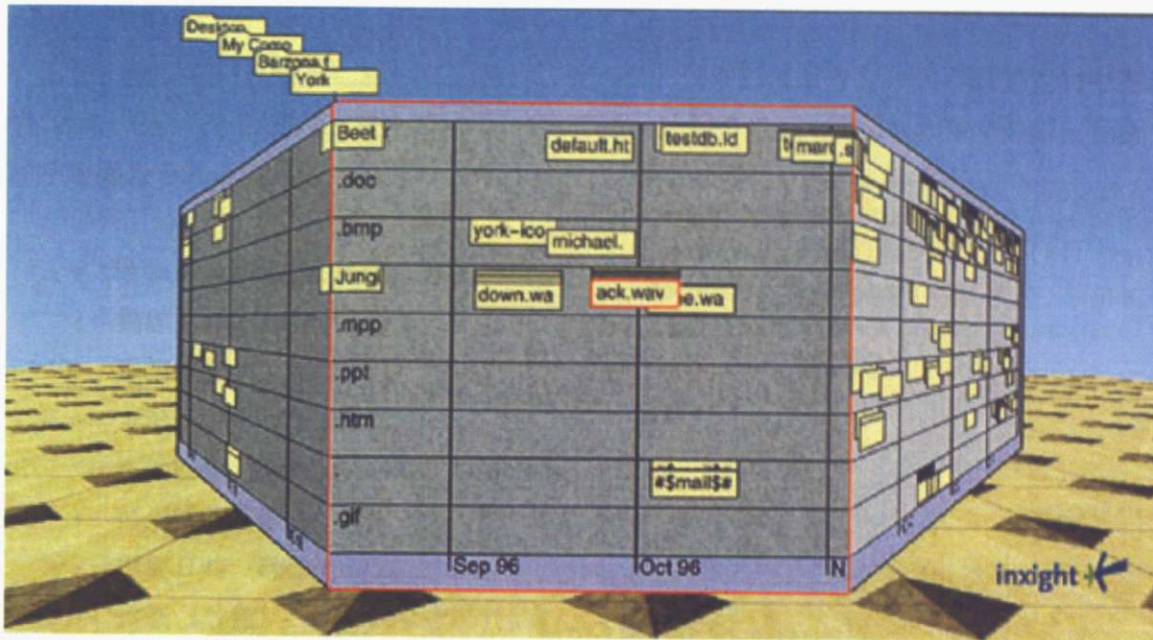


図 4.6 Perspective Wall

まず、Table Lens と Perspective Wall に共通する長所は、全体俯瞰と詳細確認が同時に行える、focus+context 技法が使われている点である。この技法は扱う情報量が多い時に特に有効な技法である。巨大なテーブルデータや線形データを従来どおりディスプレイに表示させると、確認できるのはその中のほんの一部であり、表示されているデータの周囲にどのようなデータが存在するのかという情報は完全に失われる。それに対し Table Lens や Perspective Wall では、文字情報などの詳細情報を確認しつつも周りにどのような情報があるのかをぼんやりと把握することができる。周囲の情報をどのように表現しているのかというと、Table Lens では表示しきれない数値や項目名は線の長さや色で表現され、Perspective Wall では奥行き方向に表示された壁のどの位置に情報がのっているかで表現されている。

このように、1画面に全て表示しきれない情報はその特徴だけを視覚的に表現することによって、一度に多くの情報を表示しつつ、かつ、詳細な情報も常に確認できる、というアドバンテージを生んでいる。

YouTube における検索、特に検索パターン 2（あいまい検索）の場合は、多くの動画情報を吟味したり再生したりしなければならず、検索結果の中にどのような動画があるのかを俯瞰的に眺め、おおまかな特徴を洞察することができるのは非常に大切なことである。また、知らない動画の内容を把握するためには詳細情報の確認作業も必須であり、それらを同時に行える Table Lens や Perspective Wall のコンセプトを応用することは動画検索においても高いパフォーマンスを発揮するものと考えられる。

ただ、動画という情報において、表示しきれない周りの情報をどのように視覚的に表現するか、というところは工夫を要する。YouTube の API が返す XML の中で動画の特徴づけると考えられる要素には以下のものがある。

- サムネイル画像 (画像)
- 動画タイトル (文字)
- 詳細情報 (文字)
- カテゴリー (文字)
- 再生回数 (数値)
- 再生時間 (数値)
- 投稿日時 (数値)
- 評価 (数値)

この中で動画の内容を視覚的に表現するのに最も有効と考えられる要素は、やはりサムネイル画像であろう。動画タイトルも端的に文字で内容を表してくれるので、内容を把握するのに優れているとは言えるが、周りに文字情報を大量に表示させてしまうと、各文字が読みづらくなりその有効性は失われてしまう。

そこで、サムネイル画像をその動画を代表する要素として詳細情報を確認するエリアの周りに配置することを考える。しかし、サムネイル画像はある程度の大きさを保たなければ識別不可能となり用途をなさない。Table Lens のように単純に一行に表示させるには余りにも表示領域が大きすぎる。サムネイル画像であればその一部が失われても、人間の視覚的補間能力によってある程度オリジナル画像を想像して認識することができる²⁵⁾。そのため、サムネイル画像を 3D の奥行き方向にある程度のオーバーラップを許しながらも表示させるのが最も効率の良い表示方法であると考えた。この方法では奥深くにある画像はオーバーラップが強すぎて認識することが困難になってしまうが、アニメーションによってそれらを動かすことができれば解消できる。3D の奥行き方向に情報を表示させ、アニメーションで情報を動かす方法は Perspective Wall で紹介された手法であり、この手法を応用する。

さらに、3D 表示を用いることで複数の軸が設定できる。つまり、奥行きの軸+高さの軸を利用することができる。3D の奥行きの軸にサムネイル画像を等間隔に表示しつつも、高さという軸に他のパラメータを割り当てるということが可能なのである。この時のサムネイル画像の並び順は、デフォルトで YouTube が返す検索クエリとの関連度の高い順である。

ここで重要なのが、高さ軸にどういったパラメータを割り当てるかということである。サムネイル画像が表示されている高さによって、その動画の特徴を表現する要素を割り当てる必要がある。高さがずれるということは、オーバーラップすることで失われるサムネイル画像の損失領域が少なくすむということである。つまりサムネイル画像のほぼ全体を正確に把握することができるようになり、他の情報との差異化が生まれる。また、高さで何かを表現するということは扱う情報は数値でなければならず、第3章の考察でも述べたように文字情報などはその高さで何かを表現するのに向いていない。Table Lens では数値情報を線の長さで表現しており、高さに数値パラメータを割り当てる手法は Table Lens で用いられているコンセプトの応用と言える。

YouTube API によって得られる XML の要素として数値として扱えるものは

- 再生回数 (数値)
- 再生時間 (数値)
- 投稿日時 (数値)
- 評価 (数値)

の4つであり、今回、動画を特徴づけるパラメータとして再生回数を採用することにした。再生回数が多い動画とはつまり、多くのユーザーがその動画を見たということであり、つまらない動画であれば再生回数が増えることはまずあり得ないと言える。そのユーザーの好みにもよるが、再生回数が多い=面白い動画である可能性は高くなる。また、上記したように、高さがずれるということはデータのオーバーラップが少なく動画全体が表示されるということでもあり、そのサムネイル画像をクリックできる領域も広がる。これにより面白い動画へのアクセスをより容易にすることが可能となる。ただし、投稿日時の新しい動画は必然的に再生回数も少なくなり、たとえ内容的に面白いものであったとしても視覚的に現れてこないこともある。

このように Table Lens と Perspective Wall を組み合わせることによって、検索パターン1のような詳細情報だけを確認する必要がある場合にも対応しつつ、検索パターン2のような漠然と面白い動画を探すというシーンにも対応できる仕組みを考案した。

4.5. システム

4.5.1. システムアーキテクチャ

この項では4. 4の設計コンセプトをどのように実装したのか説明を行う。

本システムは、クライアントサイドはHTML ファイル、MTX ファイル（ブラウザに表示する3D情報を保持するファイル）からなり、サーバーサイドはAPIから返されるXMLを処理するためのCGIファイルからなる。XMLの処理結果をリアルタイムでブラウザに表示させるため、Web3D技術の1つであるViewpointを用いて3Dオブジェクト情報(MTXファイル)を記述した。

図4.7に可視化結果を得るまでの処理のフローチャートを示す。

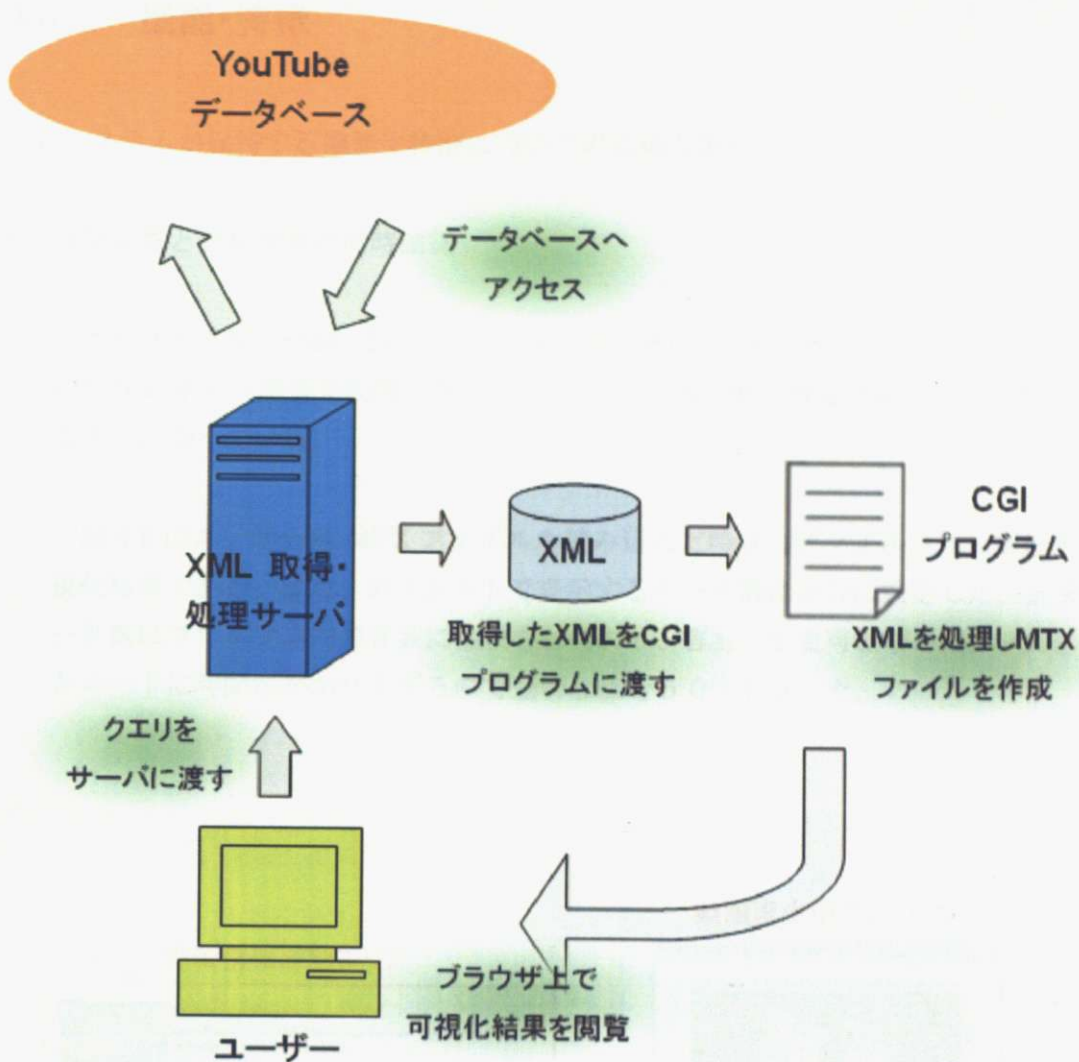


図 4.7 システムアーキテクチャ

以下、フローチャートの詳細説明を述べる。

- (1) ユーザーが規定の Web ページから検索クエリをサーバーに送る。
- (2) サーバーは受け取ったクエリをもとに URL を作成し、YouTube の API をたたく。
- (3) XML を取得し、CGI プログラムに渡す。
- (4) XML から必要な要素だけを抽出し、MTX ファイルを書き出す。その際、再生回数などから 3 次元空間内での位置等を計算により決定する。
- (5) ユーザーが HTML から作成された MTX ファイルを読み込むことでブラウザ上に 3D の可視化結果が表示される。

4.5.2. 機能・特徴

本システムが保持する機能や特徴についての詳細な説明を行う。

☆ 全体俯瞰と詳細情報の同時確認

本システムは、Table Lens と Perspective Wall のコンセプトを応用し、3次元空間内にサムネイル画像を配置したことによって全体俯瞰と詳細情報の確認が同時に行えるようになっている。

図 4.8 はユーザーが MTX ファイルを読み込んだ時点（デフォルト）で表示される可視化結果である。また、デフォルトで表示するデータ数は50に設定した。表示するデータ数はプログラム内の変数に格納されており、容易に変更可能なのだが、大きすぎるとロードに時間がかかりすぎるので適切な値を50と判断した。

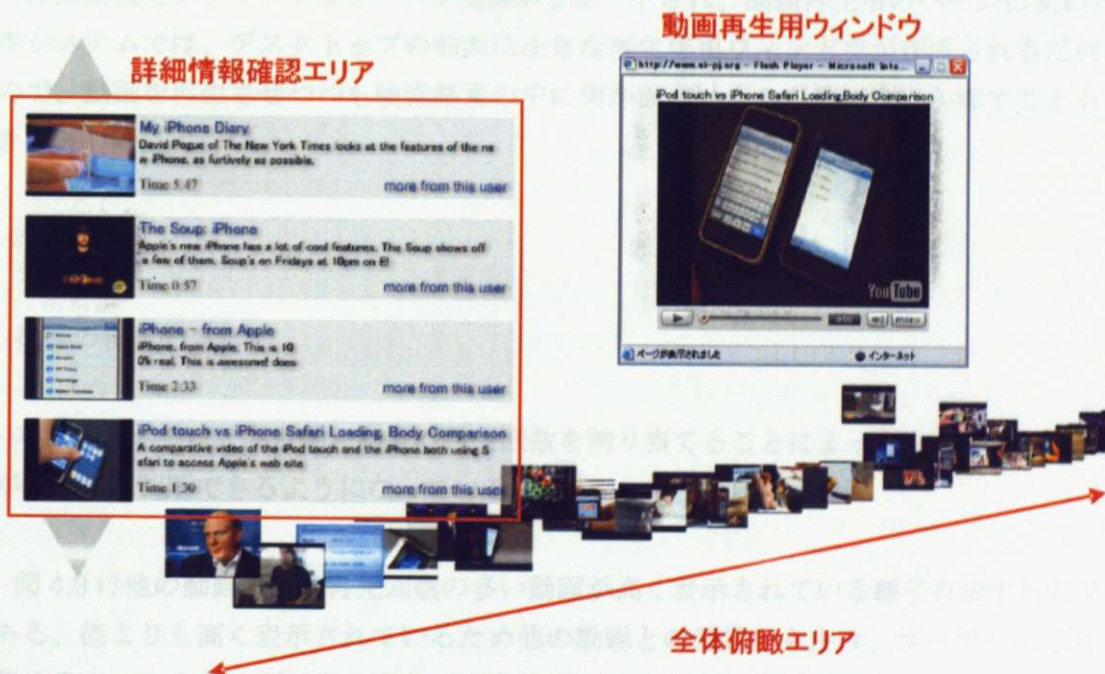


図 4.8 可視化例

ユーザーは詳細情報確認エリアで、サムネイル画像・動画タイトル・詳細説明・再生時間等の文字情報を常に確認することができ、全体俯瞰エリアで周囲にどんな情報が存在しているのかを確認することができる。詳細情報確認エリアが Perspective Wall における正面の壁と同じ役割を果たしており、奥行きで表示されるサムネイルは正面から遠ざかるにつれ小さく表示される。

まず、検索クエリとの関連度が高い検索結果 4 件を詳細情報確認エリアに表示させ、その次の検索結果からは奥行き方向に順番に配置させている。画面のディスプレイ上では L 字型にサムネイルが表示される。

動画を再生させたい場合はサムネイル画像をクリックする。クリックすると動画再生用の別ウィンドウが開かれ、そのウィンドウ内で動画が自動的に再生される。ユーザーにクリックできるということを学習させるため、サムネイル内にマウスカーソルが入るとマウスカーソルの形が変わるように設計した。また、サムネイルをクリックすることによって作成されたウィンドウは自動的に一番前のウィンドウとなるように設定しているため、作成されたウィンドウがどこにあるのか探す必要はない。

従来では、検索ページと動画再生用のページが完全に分離しており、検索ページで見つけた動画をクリックするとページ全体がリロードされ、動画再生用のページに変わる。本システムでは、デスクトップの前面に小さな再生専用ウィンドウが作成されるだけなので、動画を再生させつつも検索結果の中に何か面白いものが他にないか探すこともできる。

☆ 指標の視覚的表示

本システムでは、高さ軸に動画の再生回数を割り当てることによって、人気のある動画が視覚的に判断できるようになっている。

図 4.9 は他の動画よりも再生回数の多い動画が高く表示されている様子を示すものである。他よりも高く表示されているため他の動画との差異化を生み、ユーザーの注目を集めやすい。また、データのオーバーラップにより失われる領域が少ないため、クリックしやすくなるという利便性も生まれる。

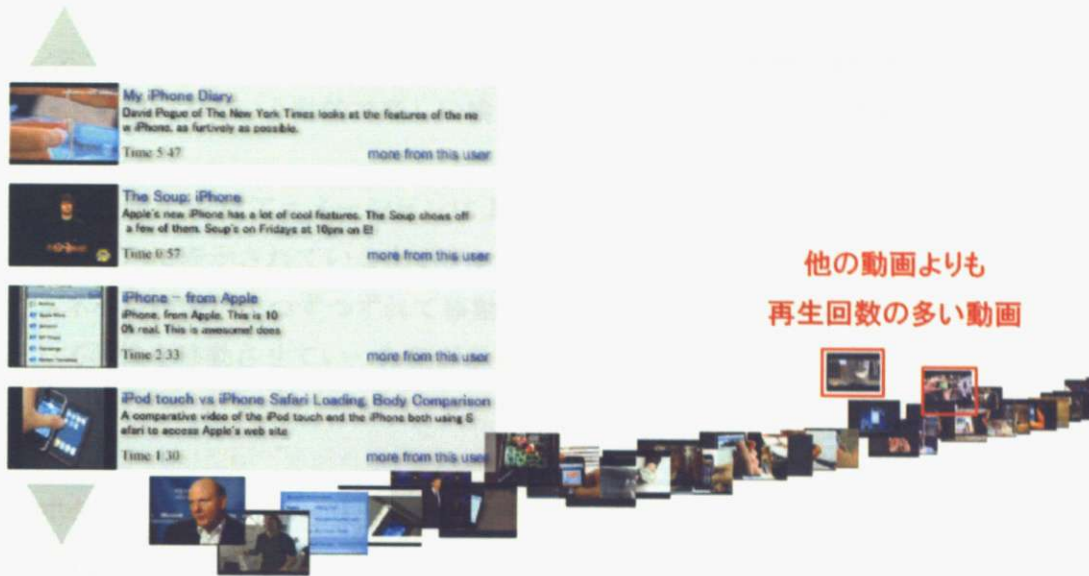


図 4.9 指標の視覚的表示の例

検索パターン2のように漠然と何か面白いものを見つけない時は、意味の広い、漠然としたキーワードで絞り込むしかないため必然的に検索結果も多くなる。従来のYouTube サイトでは1次元、つまり情報の一覧表示であるため、各情報が均一化されてしまい、差異を見つけることが難しい。そこで再生回数の多さを面白い動画の指標とし、3次元の高さ軸にそのパラメータを割り当てることによって、情報の方から面白いことをアピールしてくれるような仕組みを構築した。

ただし、高さを決める際、再生回数そのものを利用するのではなく、検索結果（デフォルトではデータ数50）の中の割合で決めることにした。つまり以下の式で高さは計算されている。なお、式中の $height$ は高さ、 a は高さ制御定数、 $view_count$ は各動画の再生回数、 MAX はデータ中で最も再生回数の多い動画の再生回数、を表す。

$$height[i] = a \times \frac{view_count[i]}{MAX}$$

再生回数は非常に分散の大きなパラメータであるので、絶対的な再生回数を使用してしまうと位置のずれがあまりにも大きくなってしまい、レイアウトが煩雑になってしまう。そこでデータ中の割合により位置を定めたのであるが、全体的に再生回数の低いデータ群であれば、その絶対値が小さい場合でも高く配置されてしまう場合もある。

☆ アニメーションによる動的な検索

本システムでは、スムーズなアニメーションをサムネイル画像に設定することにより、検索対象を動かすという動的で楽しい検索が行えるような仕組みを実現している。

図 4.10～図 4.12 はアニメーションによりサムネイル画像が移動していく様子を示している。画面内に表示されている上ボタン、または、下ボタンをクリックすることによってサムネイル画像が1つずつずれて移動していく。図では初期画面から上ボタンのみを使用して画像を移動させていった例が示されている。



図 4.10 アニメーションによるサムネイル画像の移動1



図 4.11 アニメーションによるサムネイル画像の移動2



図 4.12 アニメーションによるサムネイル画像の移動3

上下ボタンで画像を移動させていく際に、詳細情報確認エリアに入っている動画の文

字情報だけが横のパネルに表示される。必要以上に文字情報表示させないことにより、画面全体の視認性が保たれている。

従来の検索では2次元平面に貼り付けられた情報をスクロール操作などで画面送りしていくことで確認していた。本システムでは画面の方を固定し、情報の方を動かして検索を行う。それによって画面内にダイナミックな動きが生じ、ユーザーに自分が情報を操作しているのだという実感を与えることができる。動画を見るという娯楽行為において楽しく検索を行えるというのはYouTubeサイトでは実現されていないことである。

また、アニメーションによりサムネイル画像の大きさが一定の比率で変化する。例えば、上ボタンを押すごとに奥に配置されているサムネイル画像のサイズは少しずつ大きくなっていく。これにより JJGibson の主張する面の変化（勾配）が発生し、人間が実際に空間を知覚している所作をコンピューターのディスプレイ上に表示された仮想3次元空間でも実現している。

☆ 3次元表示における Ajax の応用

本システムでは、Ajax 技術と3次元アニメーションを組み合わせることで、ダイナミックな動きの中でシームレスな情報取得を行うことができる。

図 4.13～図 4.17 は3次元のアニメーション中に Ajax でシームレスな情報取得を行っている様子である。このアニメーションは約3秒間で完結する。“more from this user”という文字列をクリックすることによって、3Dのアニメーション中に投稿ユーザーの動画一覧が取得される。

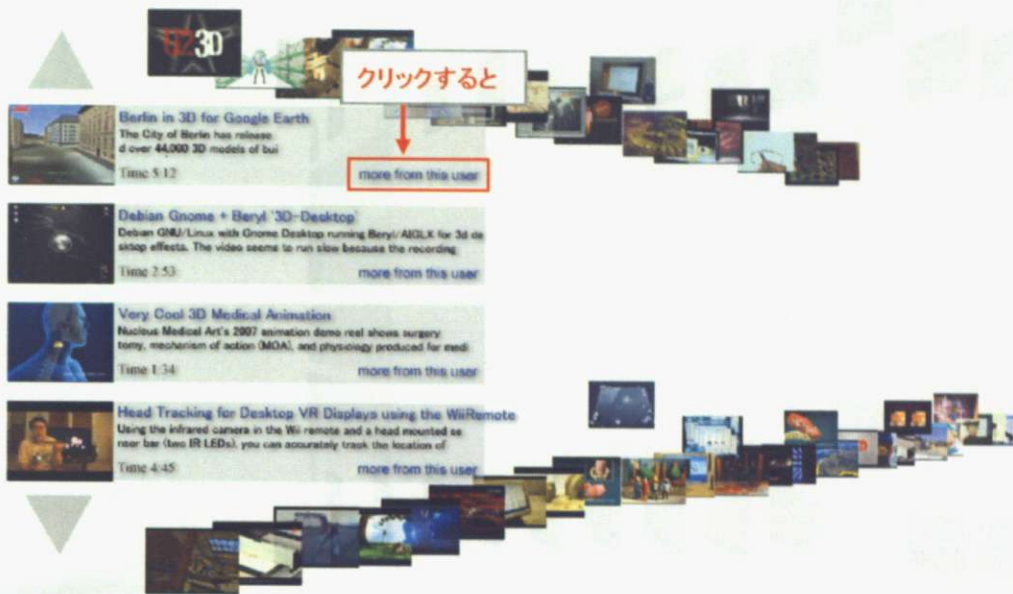


図 4.13 Ajax による情報取得1



図 4.14 Ajax による情報取得2

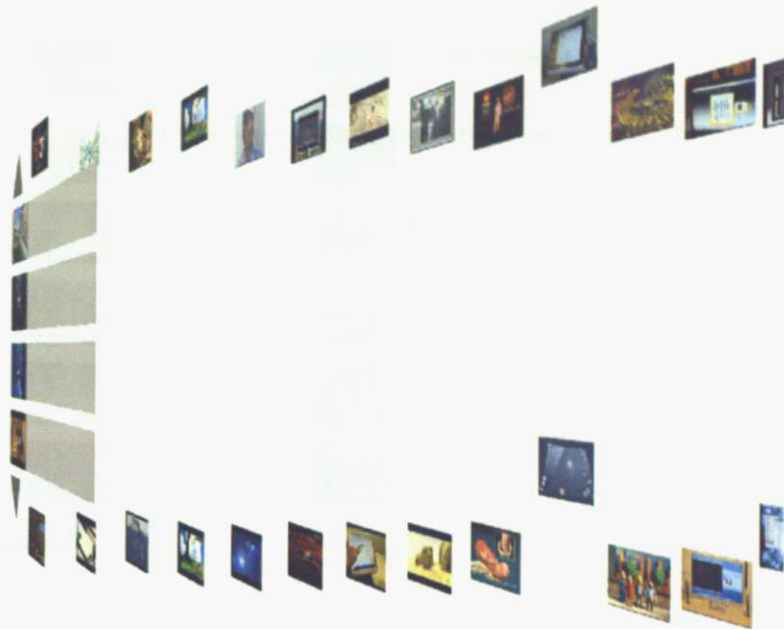


図 4.15 Ajax による情報取得3

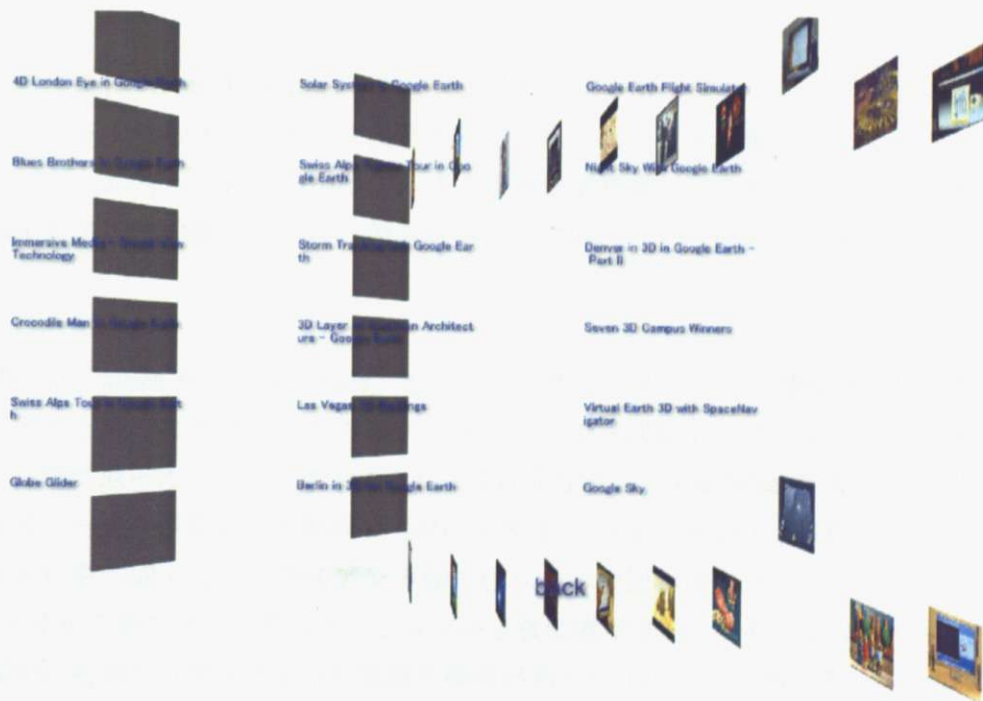


図 4.16 Ajax による情報取得4



図 4.17 Ajax による情報取得5

図??の“more from this user”という文字列をクリックすると、シーン全体が回転を始める。回転が進むにつれ最初に表示されていたサムネイル画像は消えてゆき、新しく投稿ユーザーがもっている動画のサムネイル画像が登場してくる。これは面をテクスチャの張られていない側から見ると何も表示されない、という性質を利用したものである。

ユーザーが“more from this user”をクリックするとサーバーに投稿ユーザー名が送られる。それと同時にクライアントのディスプレイでは回転アニメーションが始まる。サーバーは受け取ったユーザー名をもとに URL を作成し、YouTube の API をたたく。それによりユーザーが保持する動画の XML を取得し、それを処理し、MTX ファイル (3D ファイル) を作成する。クライアント側ではシーンの回転中にサーバーで作成された MTX ファイルを読み込み、アニメーション終了後に表示する。これによりユーザーはまるで回転先に初めからサムネイル画像等の情報があつたかのように感じる。

4.6. 評価

本システムがユーザブルであるかどうかを評価するため、ユーザビリティテストを行った。ユーザビリティテストとはユーザーが実際にシステムとどのようにインタラクションを行うかを調べる手法のことである。

PCや携帯電話など情報処理機器・システムの急速な普及とそのユーザー層の拡大に伴い、ユーザビリティ（使いやすさ、操作性）の重要性がますます高まっている。使いやすい機器・システムを開発するためにはそのユーザビリティを評価し、製品に対してフィードバックをかける必要がある。

ユーザビリティとは一般的に使いやすさや操作性のことであるが、使いやすさには様々な属性が含まれると考えられる。このユーザビリティ属性の定義としては、「Usability Engineering²⁶⁾」の著者として知られる Nielsen による定義や、ISO9241 Part11²⁷⁾における定義が代表的である。以下それぞれの定義をまとめる。

Nielsen

- ・ 学びやすさ(Learnability)
- ・ 効率の良さ(Efficiency)
- ・ 記憶しやすさ(Memorability)
- ・ 間違いにくさ(Errors)
- ・ 主観的な満足度(Satisfaction)

ISO9241

- ・ 有効さ(Effectiveness)
- ・ 効率(Efficiency)
- ・ 満足度(Satisfaction)

機器・システムのユーザビリティ評価には、ユーザーの作業時間や操作回数を計測してユーザビリティ属性を評価するユーザビリティテストと、設計段階を分析して評価するユーザビリティインスペクションがある。

ユーザビリティテストには、ユーザーの作業時間や操作回数などの定量的データを測定するパフォーマンス測定や、ユーザーの発話内容を分析して使いにくさの原因を特定する発話分析法などが含まれる。ユーザビリティインスペクションには、一貫性やエラ

一予防などのユーザビリティ原則が守られているかどうかを検査するヒューリスティック評価法や、ユーザーがその操作を容易に類推可能かどうかなどの認知的な観点で評価する認知的ウォークスルー方などが含まれる。

ユーザビリティテストでは、テストユーザーを集めたり実際に実施したりするのに時間がかかりすぎるというデメリットもあるが、それ以上に、ユーザーの口から、またユーザーの行動から生の情報を得られることは大切である。そこで、本システムにおいてもそのユーザビリティを評価するため、テストユーザーを集め、こちらが課すタスクを遂行してもらった。

4.6.1. 評価手法

本システムは設計コンセプトの節で述べたように、Table Lens と Perspective Wall を組み合わせることによって、検索パターン1のような詳細情報だけを確認する必要がある場合にも対応しつつ、検索パターン2のような漠然と面白い動画を探すというシーンにも対応できる仕組みである。これを評価するためにパフォーマンス測定を行い、タスクを遂行するまでの作業時間を計測した。タスク終了後、発話分析法の一環として、被験者に良い印象を持った点と悪い印象を持った点を紙に自由に記述してもらった。以下、被験者に課したタスクの説明を行う。

タスクは2つからなり、それぞれのタスクを本システム、YouTube の両方で遂行してもらった。つまりユーザーは計4つの作業をすることになる。以下、それぞれのタスクの説明を行う。

タスク1

特定のCMを探せ

ある特定のクエリ（例：ナイキ CM）で検索した状態からタスクを始める。そのキーワードはこちらが指定する。検索結果として40件目付近にある特定の動画（例：ナイキのCMでマイケルジョーダンが出てくるもの）を探してもらい、その時間を計測した。動画タイトル、サムネイル画像、詳細説明から目的の動画かどうかは判断できる。それを両システムで行う。ただし、両システムでクエリを変え、異なる動画を探してもらった。以下に実際に用いたクエリ、検索結果の中から探してもらった動画の1例をあげる。

クエリ：ナイキ CM

探してもらう動画：野球のCM

なぜこれが1例にすぎないかと言うと、YouTube では日々動画がアップロードされ、ある日40件目に返ってきていた結果も、次の日になると50件目が変わっていたりするためである。このため、タスクを行う前に毎回、適切なクエリ・探す動画を調査してタスクを実施した。どちらのシステムで先にタスクを行うかは毎回交互にした。

タスク2

面白いCMを探せ

各ユーザーに面白いCMを探してもらい、発見するまでの時間を計測した。その際ユーザーは自由に検索クエリを選び、入力できる。面白いかどうかの判断はユーザーの感性に委ねられており、ユーザーが満足した時点でタスク終了とした。それを両システムで行う。ただし、用いる検索クエリは両システムで全く同じものは使えないものとした。かつ、過去に見たことのあるCMを探してはいけないものとした。どちらのシステムで先にタスクを行うかは毎回交互にした。

次に、ユーザーテストの手順を述べる。

- 1、氏名、年齢、YouTube 使用経験の有無に関するアンケートに回答してもらう
- 2、操作説明をしないで、まずユーザーに自由に使ってもらう
- 3、本システムの操作説明を行う
- 4、タスク1の実施
- 5、タスク2の実施
- 6、よかった点、よくなかった点をそれぞれ紙に記述してもらう

被験者を集め、上記の一連のユーザーテストを実施した。被験者は29名で以下、被験者の構成は以下のとおりである。

表 4.1 被験者の年齢構成

年齢 (才)	人数(人)
10～20	2
20～30	27

表 4.2 YouTube の使用経験

使用経験	人数(人)
有	21
無	8

マシンの性能による公平性を期すため、ユーザーテストは全て1台のマシンで行った。マシンのスペックは表 4.3 のとおりである。

表 4.3 マシンのスペック

OS	Windows XP Professional
CPU	Intel(R) Core™2 1.66GHz 1.66GHz
メモリ	2GB

4.6.2. 各タスクの狙い

結果をまとめる前に各タスクの狙いを説明しておく。

タスク 1

タスク 1 では以下のことを調査する狙いがある。

- ・ 上下ボタンインターフェースの操作性
- ・ 詳細情報の確認のしやすさ

タスク 1 はある特定の動画を探すという作業であり、これは検索パターン 1 に合致する。検索パターン 1 では、目的の動画がはっきりとイメージできているため、動画タイトル・サムネイル画像・詳細情報を確認すれば目的の動画かどうかを判断できる。そのため、画

面内に表示された情報の操りやすさ、詳細情報の確認のしやすさがタスク達成までの時間に影響を与えると考えられる。

従来の検索では目的の動画を探すのに画面スクロール機能やページ切り替え機能をつかって検索を行っている。これに対し本システムでは、情報側を動かすという検索方式を用いており、そのためのインターフェースとして上下ボタンインターフェースを設置してある。また、詳細情報確認エリアを設け、そのエリア内にある動画だけの詳細情報を表示するという方法を用いている。

当初、ある特定の CM を探せというタスクを予定していたが、これはインターフェースの操作性というよりは検索クエリの選び方のうまさが大きく関係してくるため、こちらでクエリを指定し、検索を行った状態からタスクを開始してもらった。

タスク 1 を終了するまでの時間を計測することで、検索パターン 1 における両システムの有効性を評価する。タスク 1 では、本システムと従来システムで同程度のパフォーマンスを発揮することを期待する。

タスク 2

タスク 2 では以下のことを調査する狙いがある。

- ・ 再生回数の多い動画を視覚的に目立つように表示させた有効性
- ・ 全体俯瞰エリアを設け多くの情報を一度に表示させたことの有効性

タスク 2 は漠然と何か面白い動画を探すという作業であり、これは検索パターン 2 での検索に合致する。検索パターン 2 では、意味の広いキーワードで検索するしかないため、返ってくる検索結果も必然的に多くなる。この中からサムネイル画像・動画タイトル・詳細説明・再生時間等を総合的に判断して、見る価値があるかどうかを判断しなければならない。中にはそれだけでは判断できず、動画を再生してみないと面白いかどうか判断できないものもある。そのため、他のデータとの差異化、多くのデータへのアクセスのしやすさがタスクの達成時間に影響を及ぼすものと考えられる。

従来の検索結果の表示方法は 1 次元かつ少数の情報表示であり、検索パターン 2 においては問題となる場合があった。これに対し本システムでは、3 次元表示を用いることで複数の軸を設定し、多次元情報のより多様な属性を表現している。具体的には 3 次元空間の高さ軸に再生回数を割り当てることで視覚的に面白い動画を判断できる仕組みを考案した。

また、全体俯瞰エリアに周囲の情報も一度に表示させることで、全体の中での位置把握、過去に確認した情報・新しい情報へのアクセスが容易にできるよう設計してある。

タスク2を終了するまでの時間を計測することで、検索パターン2における両システムの有効性を評価する。タスク2では、本システムが高いパフォーマンスを発揮することを期待する。

4.6.3. 自由記述アンケートの狙い

タスク1とタスク2で定量的に測定できなかった、有効性や改善点を発見する。具体的には、細かい機能に対する良し悪しや全体的な印象などを判断する。

4.6. 結果・考察

タスク1における考察

まず、タスク1の結果から示し、それに対する考察を述べる。以下2Dは現行のYouTubeシステム、3Dは本システムのことを指すこととする。

図4.18に全被験者の実験データを示し、表4.4にその平均、分散、標準偏差を示す。

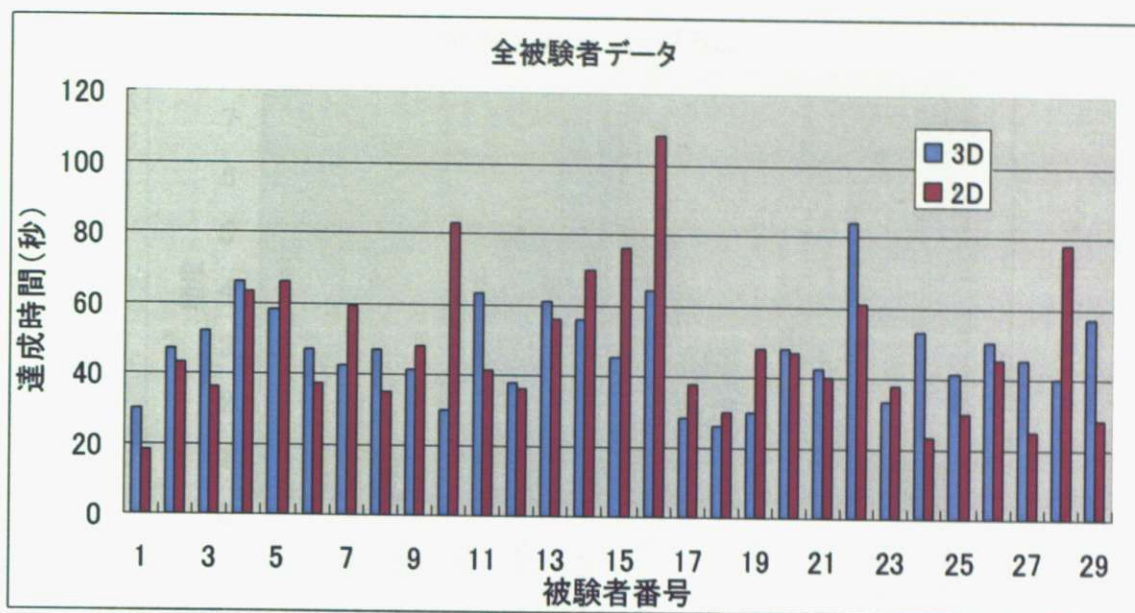


図 4.18 全被験者データ(タスク1)

表 4.4 達成時間の平均、分散、標準偏差(タスク1)

	3D	2D
平均	47.0 秒	48.5 秒
分散	171.3	410.9
標準偏差	13.1	20.3

表 4.4 の平均値より両システムにおいてほぼ同程度のパフォーマンスが得られたように見えるが、平均の値に対して標準偏差が十分に小さいとは言いきく、平均値の信頼性は薄れる。そこで、どちらのシステムがどれだけ速かったのかを調べるため、2D から3Dの時間差分をとり、頻度分布をとった。つまりマイナス値=作業時間が長くなったことを意味する。図 4.19 にそのグラフを示す。ただし、パフォーマンスが向上した部分(プラス値)を濃い色で、パフォーマンスが低下した部分(マイナス値)を薄い色で表している。

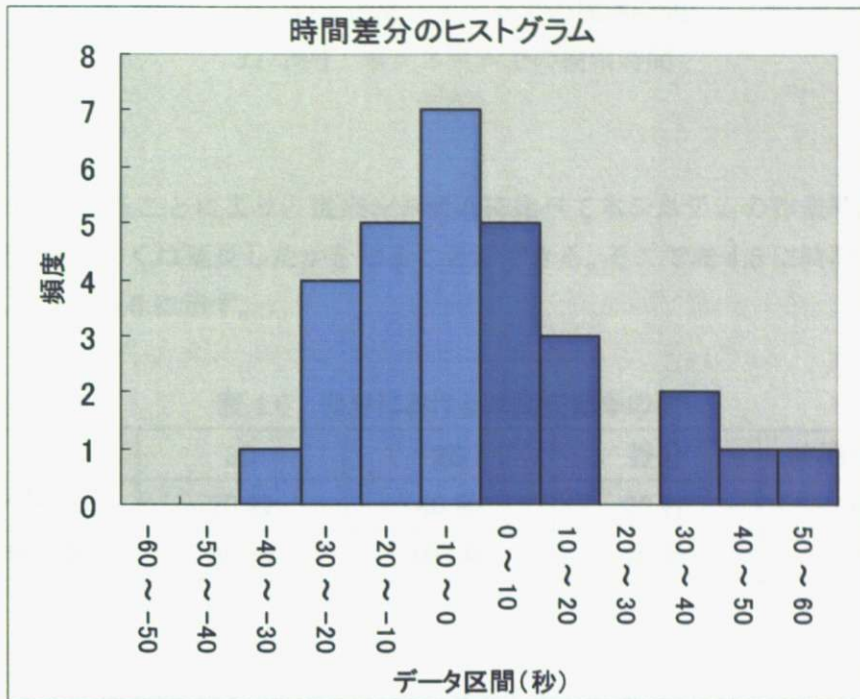


図 4.19 時間差分(2D-3D)のヒストグラム(タスク1)

この図からわかることは、やや3Dの方でパフォーマンスが低下しているということである。しかし、このヒストグラムでは時間差分をそのまま用いている。その値が果たして信用できるのかどうか考察してみる。たとえば以下の表のようなデータがあったとする。

表 4.5 差分検証の例

	3D	2D	差分
ユーザーA	20 秒	40 秒	20 秒
ユーザーB	80 秒	100 秒	20 秒

ユーザーA、ユーザーBともに20秒ずつ速くなっている。しかし、両者におけるこの20秒は等価値ではない。従来のシステムに対して本システムがどれだけ速くなったのかという観点から見れば、圧倒的にユーザーAの方が良いパフォーマンスを発揮したと言える。そこで以下の式のように短縮された時間の割合（時間短縮率）を定義する。

$$\text{時間短縮率 [\%]} = \frac{2D - 3D}{2D} \times 100$$

2D[秒]: 従来システムでの検索時間

3D[秒]: 本システムでの検索時間

分母を2D にすることにより、従来システムに比べて本システムの作業時間がどれぐらいの割合で短縮もしくは延長したかを知ることができる。そこで表 4.5 に時間短縮率の要素を加えたものを表 4.6 に示す。

表 4.6 差分における時間短縮率の例

	3D	2D	差分	時間短縮率
ユーザーA	20 秒	40 秒	20 秒	50% ↑
ユーザーB	80 秒	100 秒	20 秒	20% ↑

そこで、この時間短縮率についてもヒストグラムをとり図 4.20 にその結果を示す。ただし、パフォーマンスが向上した部分（プラス値）を濃い色で、パフォーマンスが低下した部分（マイナス値）を薄い色で表している。

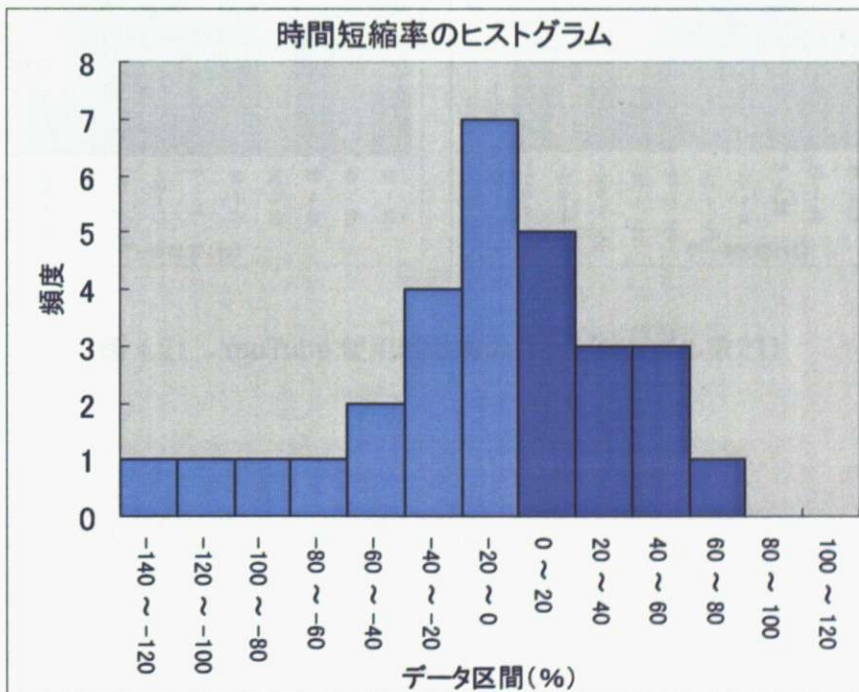


図 4.20 時間短縮率のヒストグラム(タスク1)

この図からも、やや3D の方でパフォーマンスが低下していることが読み取れる。差分

そのものによる評価に加え、時間短縮率という観点からも同じことが言えるということは、タスク1において本システムではやや不利となることがわかったと言ってよい。

それではなぜタスク1において本システムが不利に働いたのか考察を行う。タスク1は情報を確認し、それが目的の動画かどうかを判断する、という極めてシンプルなタスクのため、インターフェースの扱いやすさが強く影響する。そのため、従来のYouTubeサイトを使用したことのある被験者は、その操作に慣れており有利に働いたと言える。しかし、YouTubeサイトを利用したことのない被験者は、その操作や目的のものを見つけ出すまでの一連の流れを経験したことがないため、タスクにバイアスがかからない。

そこでYouTubeの使用経験の有無でデータを分け、再分析を行う必要があると言える。YouTube使用経験の有無でデータを分け、時間差分のヒストグラムと時間短縮率のヒストグラムの両方を作成した。以下、図4.21~4.22にその結果を示す。

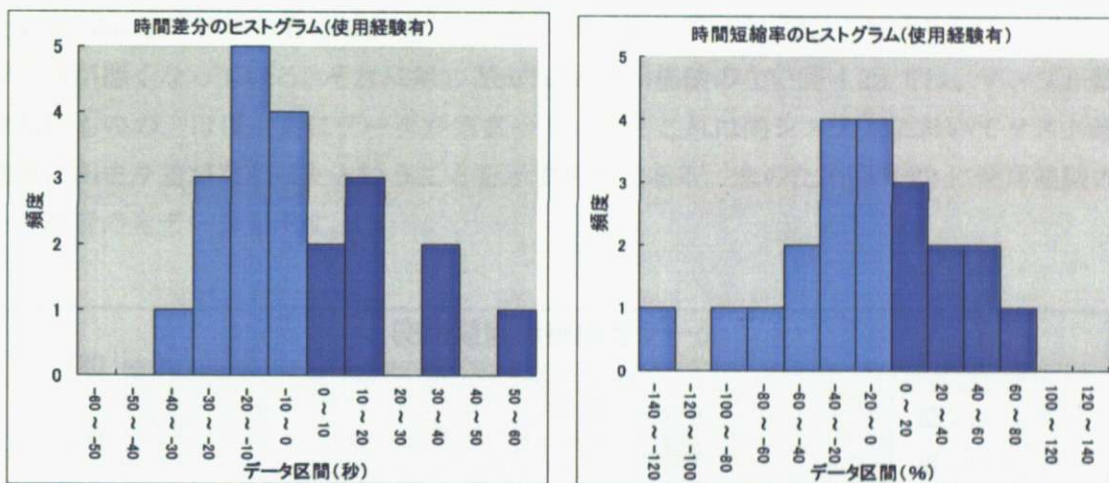


図 4.21 YouTube 使用経験有のグラフ(サンプル数21)

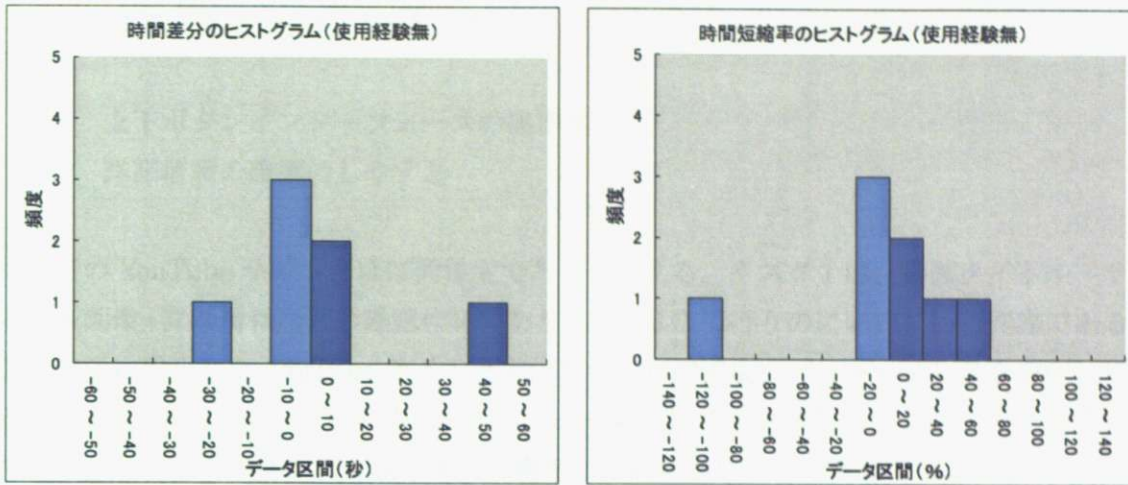


図 4.22 YouTube 使用経験無のグラフ(サンプル数8)

YouTube 使用経験の有る被験者だけのデータである図 4.21 では、やはり 3D のパフォーマンスが悪くなっている。それに対し YouTube 使用経験のない図 4.22 では、サンプル数は少ないものの、ほぼ中央にデータが集まっている。これは両システムにおいてタスク遂行結果にあまり差がなかったということを示すものである。念のため図 4.23 に使用経験の無い被験者の全データを示す。

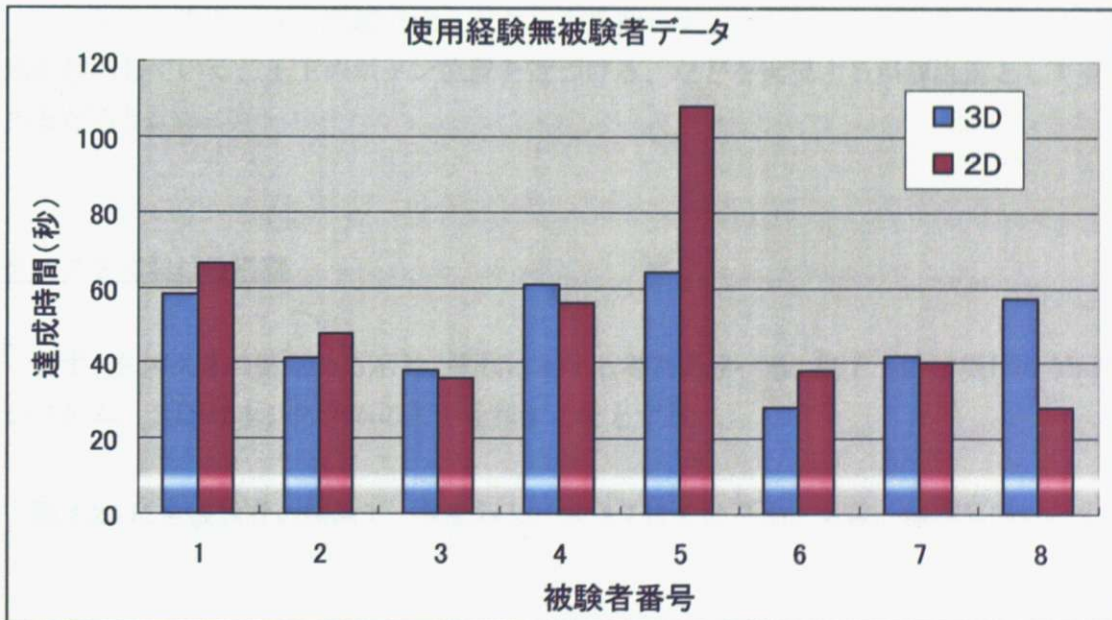


図 4.23 YouTube 使用経験が無い被験者のみのデータ

これによりタスクの狙いである以下の

- ・ 上下ボタンインターフェースの操作性
- ・ 詳細情報の確認のしやすさ

は現行の YouTube サイトとほぼ同程度であると言える。タスク 1 は、動画タイトル・サムネイル画像・詳細情報などの確認作業であり、本来 2D で行うのに向いている作業である。2D 表示に特化した YouTube サイトと本システムが同程度のパフォーマンスを発揮しているということは、本システムがインターフェースとして十分機能していることを示すものである。このことがサンプル数を十分に増やした時においても成り立てばより磐石となる。

ただし、あくまで同程度のパフォーマンスであるということは一考の余地があるということである。本システムでは、ボタンインターフェースの発動きっかけとしてマウスクリックを用いている。動画を動かす検索を行うためにはマウスを何度もクリックする必要がある。マウスをクリックする速さと画面をスクロールさせる速さとは速度の面からもそうであるが、仕事量の面からもかなりの不利がある。自由記述アンケートからも使いにくい点としてマウスクリックの問題が多くあげられていた。システムを数回使うだけのインターフェースなら問題ないが、長く使うシステムにおいては操作のわかりやすさ・覚えやすさに加え、この仕事量の観点が重要であろう。

このマウスクリックの問題では、ボタンを長押しすることによって一定のスピードで動画が移動していく、上下のボタン位置を近づける、などを実現するが解決策として有効であるだろう。

タスク 2 における考察

まず、タスク 2 の結果から示し、それに対する考察を述べる。以下 2D は現行の YouTube システム、3D は本システムのことを指すこととする。

図 4.24 に全被験者の実験データを示し、表 4.7 にその平均、分散、標準偏差を示す。

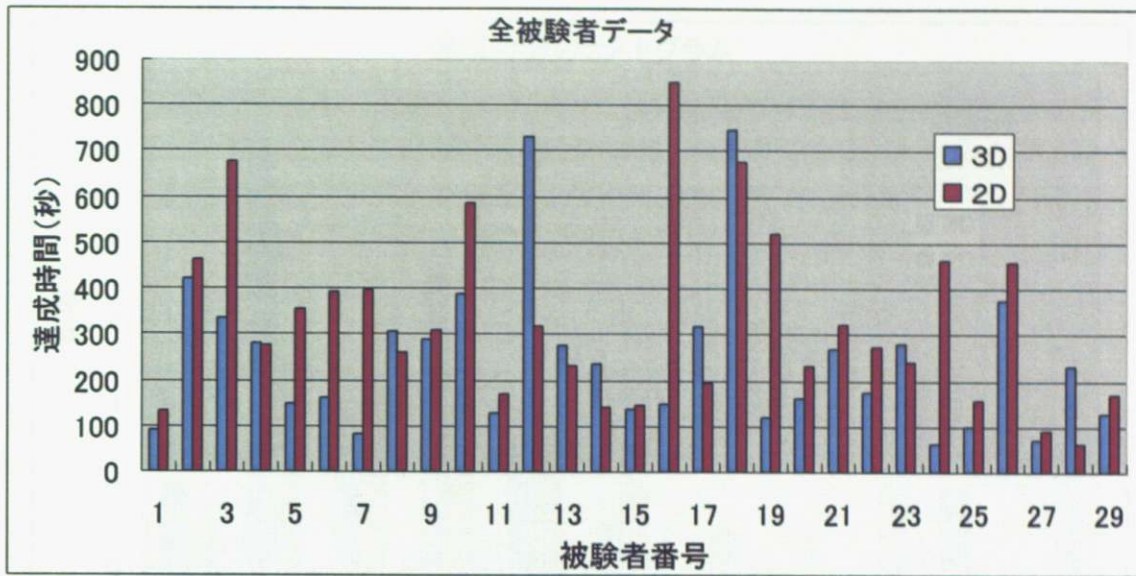


図 4.24 全被験者データ(タスク2)

表 4.7 達成時間の平均、分散、標準偏差(タスク2)

	3D	2D
平均	248.344 秒	329.586 秒
分散	27864.36385	36386.3805
標準偏差	166.9262228	190.7521442

表 4.7 の平均値からタスク 2 においては、本システムの方が良いパフォーマンスを発揮したことになる。両システムの平均値で約 80 秒という大きな開きはあるものの、平均の値に対して標準偏差があまりにも大きいので平均値そのものはほとんど信用できないことになる。そこで両システムの頻度分布を考える。図 4.25 に両システムの達成時間のヒストグラムを示す。

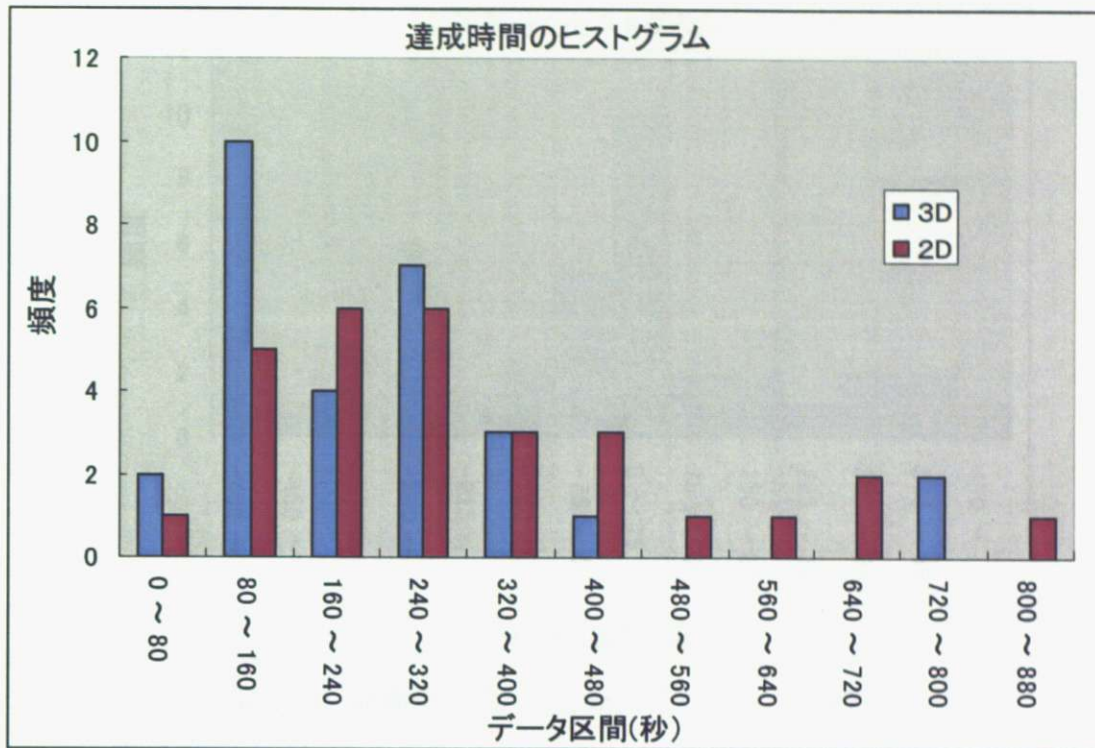


図 4.25 達成時間のヒストグラム(タスク2)

この図から 3D と 2D では多少ピークがずれていることが読み取れる。3D ではピークが 80~160 付近にあるのに対し、2D では 160~240 付近にある。このことからタスク 2 では本システムの方が高いパフォーマンスをだしたと言える。このことからある程度タスク 2 の狙いであった以下の、

- ・ 再生回数の多い動画を視覚的に目立つように表示させた有効性
- ・ 全体俯瞰エリアを設け多くの情報を一度に表示させたことの有効性

において本システムは従来システムに比べ優位であったことは言えそうであるが、いったいどれほど有効だったのかまではわからない。そこで以下、タスク 2 における本システムの定量的評価を試みる。

まず、図 4.26 に 2D から 3D の達成時間を引いた、差分値のヒストグラムを示す。ただし、パフォーマンスが向上した部分（プラス値）を濃い色で、パフォーマンスが低下した部分（マイナス値）を薄い色で表している。

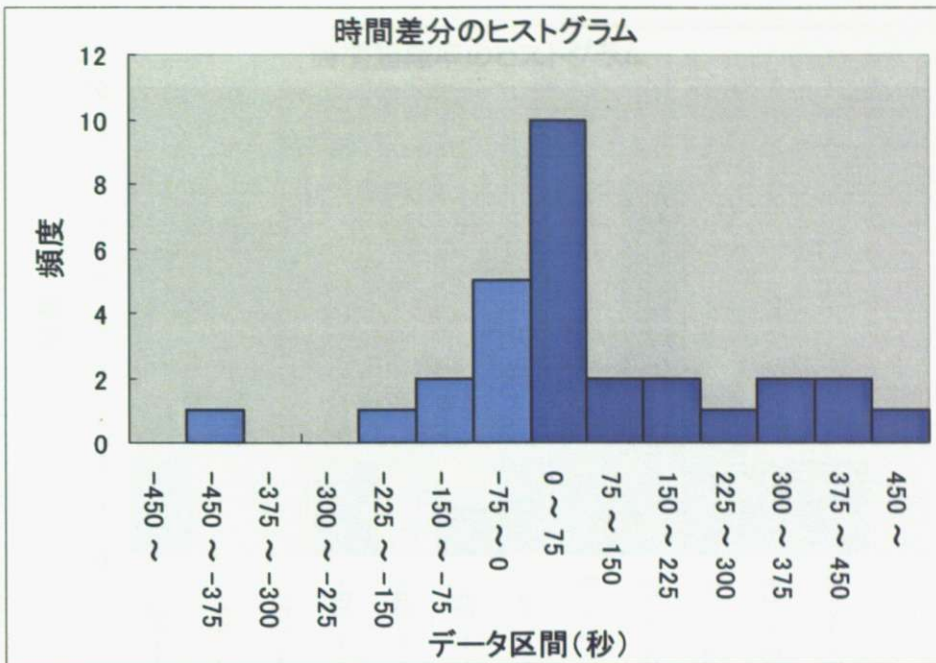


図 4.26 時間差分(2D-3D)のヒストグラム(タスク2)

図 4.26 から、値としてはプラスのものが多いことが読み取れる。このことからタスクを達成するまでの時間は本システムで短くなったユーザーが多いことを示す。計算された平均値で言えば、約 80 秒短縮された。しかし、最も高いピークは 0~75 付近に存在し、両システムの差分の値は小さいものが多い。このことから、単純にタスク達成時間という観点で本システムの有効性を示すのは無理がある。

そこで、タスク 1 の時と同様、時間短縮率を考えてみる。そのグラフを図 4.27 に示す。

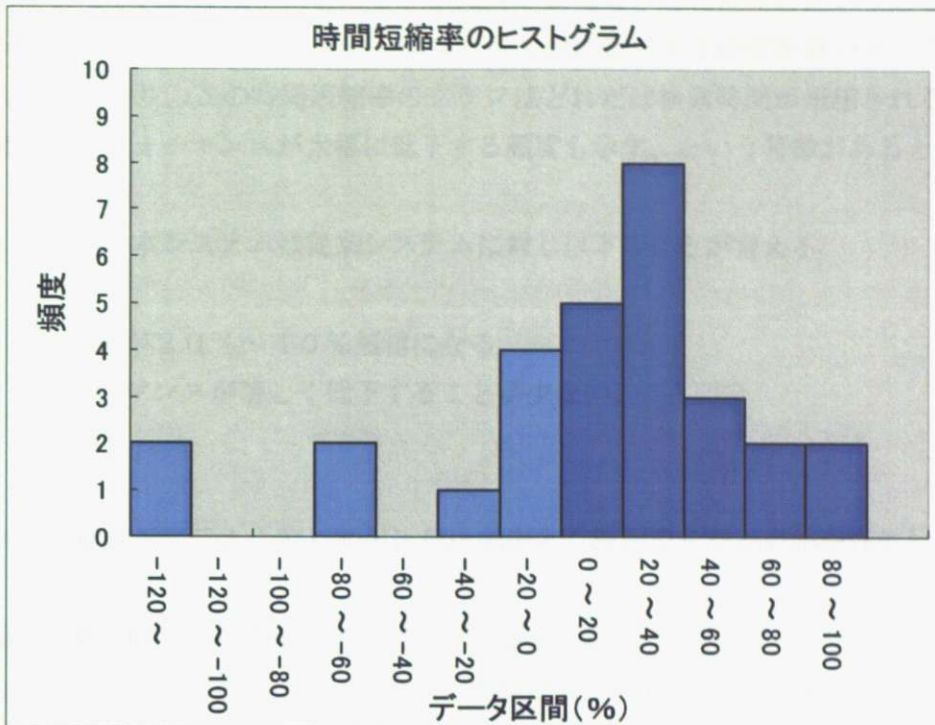


図 4.27 時間短縮率のヒストグラム(タスク2)

図 4.27 より、ピークが 20%~40% 付近にあり、本システムは従来システムに比べ作業時間が 20%~40% 短縮することが推測される。しかし、時間短縮率においてその平均をとってみるとほんの 6% 程度であった。このずれはいかにして生まれたのだろうか。

その原因として考えられるのは、時間短縮率の求め方そのものである。時間短縮効率のとり得る値の範囲は上限 100% であるにも関わらず下限はマイナス無限大まで取れる。

$$-\infty < \text{時間短縮率} < 100 \quad (\%)$$

特に、2D に比べ 3D での検索時間が大幅に伸びた場合に、大きなマイナス値が生じる。逆に 2D に比べ 3D での検索時間が大幅に減少した場合でも、最高 100% しかとることができない。つまり、この評価手法はデータにより有利不利が生じると言える。特に本タスクのように両データに大きな差が生まれる場合はなおさらである。実際、図 4.27 の一番左で頻度 2 を表しているデータは -130% と -277% である。これが平均値に影響し、ヒストグラムにおけるピークとずれが生じたものと考えられる。そこでこれらの値をはずれ値とし、本システムでは従来システムに比べ 20%~40% 検索時間が短縮されたとするのが妥当だろう。

また逆に言えば、このはずれ値の生じる頻度が無視できない程高いということは、従来システムに対し本システムのパフォーマンスが大幅に低下する確率が高いということを示すと言える。つまり、この時間短縮率のグラフはどれだけ検索時間が短縮されるのかを示すと同時に、パフォーマンスが大幅に低下する頻度も示す、という特徴があると言える。

以上のことから本システムは従来システムに対し以下のことが言える。

- ・ 検索時間が20%~40%短縮になる。
- ・ パフォーマンスが著しく低下することが少ない。

では次に、従来システムに対してどれだけ効率よく検索できたかのかを評価したい。

この場合、検索開始から検索終了までの全仕事量に対し、1秒当たりどれだけの仕事をこなすことができたのかに注目する。例えば、被験者Aが全仕事量1の検索に10秒かかったとすると、被験者Aの1秒間で行う仕事量は1/10となる。そこで従来システムに対して、単位時間あたりどれだけ多く仕事のできたのかを、作業効率と呼び以下の式で定義する。

$$\text{作業効率} [\%] = - \frac{\frac{1}{2D} - \frac{1}{3D}}{\frac{1}{2D}} \times 100$$

$2D$ [秒] : 従来システムでの検索時間

$3D$ [秒] : 本システムでの検索時間

例えば、 $3D=10$ [秒] $2D=20$ [秒]だった場合、

$$\text{作業効率} [\%] = - \frac{\frac{1}{20} - \frac{1}{10}}{\frac{1}{20}} \times 100 = 100 [\%] \quad \text{と計算され、100\%の作業効}$$

率UPとすることができる。符号が時間短縮率と反対になるのは、時間は小さい値ほど高評価となるが、仕事量は大きい値ほど高評価となるためである。

しかし、この作業効率はとり得る値の範囲が上限プラス無限大なのに対し下限が-1と

なり、時間短縮率の場合と逆のことが起きる。

$$-100 < \text{作業効率} < +\infty (\%)$$

特に2D に対して3D での検索時間が大幅に短縮された場合、大きなプラス値をとることになる。これは時間短縮率の時とは逆で、作業効率のヒストグラムは本システムの作業効率がどれだけ向上したかを示すと同時に、パフォーマンスが大きく向上する頻度も示す、という特徴を保持するということである。

このようなことを踏まえた上で作業効率のグラフを図 4.28 に示す。

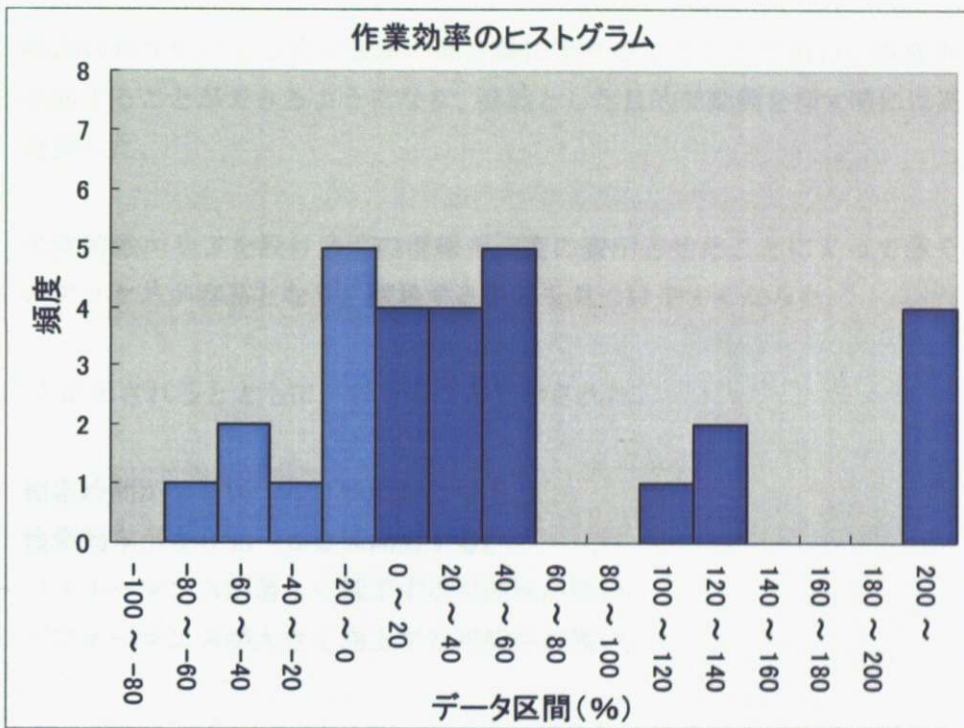


図 4.28 作業効率のヒストグラム(タスク2)

図 4.28 から、20~40 付近でピークがあることが読み取れる。しかし、全体の平均値を計算してみると、82%作業効率が上がったという結果が得られた。これはグラフ右端の他に比べ非常に大きな値が影響したと考えられる。時間短縮率においてこの大きな値は全体に比べ頻度が少なかったためはずれ値とみなしたが、図 4.28 ではピークとほぼ同程度の頻度でこのはずれ値が生じているため、無視することはできない。よって、ピークがひとつ右にあると考え、本システムは従来システムに比べ、40%~60%検索効率が向上したとするのが妥当だろう。

また、高い頻度でこのような値が生じるということは、従来システムに対して本システムの作業効率が大幅に向上する確率が高いことを示すものである。

以上のことから本システムは従来システムに対し以下のことが言える。

- ・ 検索効率が40%～60%向上する。
- ・ パフォーマンスが大きく向上する可能性が高い。

これら考察を全て踏まえ、タスク2での狙いであった以下の2点

- ・ 動画の再生回数を3次元表示の高さ軸に割り当てることで面白い動画を視覚的に判断することができるようになり、漠然とした目的で動画を探す時には高い効果を発揮した。
- ・ 全体俯瞰エリアを設け多くの情報を一度に表示させたことによって多くの動画へのアクセスが容易となり、満足する動画を見つけやすくなった。

ということが示されるとともに、以下のことも示された。

- ・ 検索時間が20%～40%短縮になる。
- ・ 検索効率が40%～60%向上する。
- ・ パフォーマンスが著しく低下する可能性が低い。
- ・ パフォーマンスが大きく向上する可能性が高い。

特に上記のリストにおいて、「パフォーマンスが著しく低下する可能性が低い」ということが示されたのは重要なことである。いくら全体としてパフォーマンスが向上していても、著しくパフォーマンスが低下する可能性が高ければそれは良いシステムであるとは言えない。面白い動画を発見するという偶発性の高いタスクにおいて、多少の時間の差異がユーザーに与える印象を決定付けるとは考えにくい。しかし、著しくパフォーマンスが低下する（検索時間が長くなる）、ということは少なくともユーザーに与える印象が良いとは言えない。特にWeb上のインターフェースにおいては誰もが簡単に扱える必要があり、ある一定のパフォーマンスを常に発揮できるというのはかなり大きな評価ポイントである。

タスク1と2における考察を合わせ、以下のことが言える。

「本システムは、視覚的に情報の特徴をとらえることができるため新しく有意な情報を発見するのに有効であるのと同時に、詳細な情報を確認するという作業においても従来の2D表示に劣らない性能を保持する。」

自由記述アンケートにおける考察

自由記述アンケートを元に考察を行う（発話分析）。ユーザーの生の声を分析することでタスク1とタスク2で明らかになったことを裏付ける、もしくは、予想できなかったような長所や課題を知ることが目的である。

最初に記述式アンケート結果の内容を簡易化した表を以下に示す。（複数回答可）

表 4.8 自由記述アンケート結果(良いと感じた点)

良いと感じた点	人数(29人中)
面白い動画が一目でわかる	20
全体を見渡せる	7
アニメーションで動画が移動する	5
余計な情報が少ない	4
画面スクロールの手間が省ける	4
戻るボタンを押す回数が少ない	4
1画面で多くの情報が見られる	3
かっこいい、おしゃれ	3
文字が読みやすい	3
奥行きが新鮮	2
おもしろい	2
検索が楽しい	1
別ウィンドウで動画が再生されて便利	1
暇な時に使いたくなる	1

表 4.9 自由記述アンケート結果(良くないと感じた点)

良くないと感じた点	人数(29人中)
上下ボタンを何度も押さなければいけない	6

端までいくと戻るのが大変	5
1つの動画に対する情報量が少ない	3
表示される情報が多すぎる	3
表示された画像サイズが小さい	3
動画をうまく動かせなかった	3
全体俯瞰エリアの動画詳細が確認できない	2
詳細情報を表示できる数が少ない	2
再生回数だけでなく他の特徴も知りたい	2
システムの反応が少し遅い	2
飛び出た動画が面白くなかった	1
トータルヒット数がわからない	1
クリックではなく、キー操作のほうがよかった	1
詳細情報確認エリアに動画が隠れてしまう	1
クリックが速いと画像がずれる	1
画像の重なりが大きすぎる	1
上下ボタンの距離が遠い	1
検索画面から新しい検索ができない	1
慣れが必要	1
一覧表示のほうが見やすい	1
詳細情報確認エリア内の情報に差異がない	1
特になし	2

良いと感じた点では、「面白い動画が一目でわかる」という点が29人中20人を占め、圧倒的に多い。これは漠然と何か面白いものを探せというタスク2の結果にも現れており、定量データ、発話データの両方から同じことが言えるため、再生回数で動画を配置する高さを変えた効果が非常に大きかったと断言できる。しかし、「飛び出た動画が面白くなかった」という意見も一人あった。これは動画を配置する高さを再生回数そのもので決めているのではなく、検索結果中の割合で決めているためであると考えられる。もしくは、個人の感性と大衆の感性とのずれとも考えられる。

次に多いのは「全体を見渡せる」という点である。これは「1画面で多くの情報が見られる」というコメントとほぼ同意で、両者を合わせると10人の人がそれらを良い点としてあげている。タスク2では検索結果が多く返ってくるので必然的に情報に目を通す量も増える。そうなると、全体情報の中での位置把握、過去の情報・新しい情報へのアクセスの容易さが重要となり、コメントの中にもその結果が現れたものと考えられる。

上記の2点はタスク2の遂行時間にも現れており、本システムが有効であることを裏付けるものとなった。

また、記述式アンケートでユーザーの生の声を聞いたことによりタスクでは得られなかった意外なことも知ることができた。それは、本システムがユーザーの感性をくすぐるものであったということである。これは、「かっこいい・おしゃれ」(3人)、「おもしろい」(2人)、「奥行きが新鮮」(2人)、「検索が楽しい」(1人)というような機能には関係ないコメントをした人が多いことから伺える。つまりこのことからユーザーの主観的満足を高く評価することができる。

次に良くないと感じた点に関する考察をまとめる。良くないと感じた点として圧倒的に多いのがボタンインターフェースに関するコメントである。「上下ボタンを何度も押さなければならぬ」(6人)、「端までいくと戻るのが大変」(5人)、「動画をうまく動かさなかった」(3人)、「クリックではなくキー操作の方がよかった」(1人)、「上下ボタンの距離が遠い」(1人)、「慣れが必要」(1人)というように29人中17人が良くなかった点として操作性に関する問題をあげている。

タスク1では主に操作性に焦点を絞り、調査を行った。得られた結果から、上下ボタンインターフェースの操作性・詳細情報の確認のしやすさは従来システムと同程度であると考察したが、ユーザーは潜在的にはこの操作を快く思っていなかった可能性がある。上に羅列したコメントは操作しづらい、覚えにくい、というよりはむしろ操作が面倒くさいというコメントであり、操作における仕事量の軽減が必要であることがわかる。使う機会の少ないシステムなら問題ないが、頻繁に使うシステムにおいては操作にかかる仕事量が多いとユーザーは不快を感じるだろう。操作のわかりやすさ・覚えやすさに加え、仕事量の少なさという観点は今後の課題である。

次に多いコメントが詳細情報に関するものである。「1つの動画に対する情報量が少ない」(3人)、「全体俯瞰エリアの動画詳細が確認できない」(2人)、「詳細情報を表示できる数が少ない」(2人)と全体に占める割合は多い。本システムでは詳細情報として、動画タイトル・詳細説明・再生時間を表示させたわけだが、現行のYouTubeサイトではこの他に再生回数・投稿日時・投稿ユーザー名・カテゴリー・タグという要素も文字情報として扱っている。ユーザーは最終的に動画に関する情報を文字情報として確認したいと思っており、その段階において3次元表示は無意味となる。そこで文字情報の扱い方が3次元表示の課題であるとも言え、そのデザインの良し悪しがシステムの出来を左右するといっても過言ではない。

3次元表示において文字情報がオーバーラップを起こすとその視認性は極端に落ちてしまうため、扱える文字情報の多さは2D表示に勝てない。動画検索においては扱う文字情報はさほど多くないため、対策として詳細情報確認エリアをもう少し大きくする、全体俯瞰エリアにある詳細情報はマウスオーバーで表示させる、ということが考えられる。しかし、確認すべき文字情報の量が多いようなシステムの場合、ディスプレイ内で全体俯瞰エリアと詳細情報確認エリアの2つの役割を分担させることは難しく、できたとしてもフォントが小さい、行間が狭いという理由から視認性が落ちることは間違いない。このような場合は、全体俯瞰と詳細情報の確認を別ステップとして完全に分離するやり方を採用するのが賢明だろう。

また、「画像の重なりが大きすぎる」とコメントしたのはほんの1人であり、アニメーションの設定が効果的だったことが伺える。本システムにおいて、奥深くに配置されたサムネイルは前後で重なりがかなり大きく、ほぼ認識できない程度である。それにも関わらずユーザーに対する印象が悪くないのは動画を動かすことができるからだろう。ある程度近くまで来ると認識できるようになり、ユーザーは操作の中でそのことを学習したと言える。

以上、タスク1・タスク2・自由記述アンケートで行った考察をもとに Nielsen のユーザビリティの観点から本システムを評価する。

Nielsen のユーザビリティの定義は以下のとおりである。

- 学びやすさ(Learnability)
- 効率の良さ(Efficiency)
- 記憶しやすさ(Memorability)
- 間違いにくさ(Errors)
- 主観的な満足度(Satisfaction)

以下、簡潔にそれぞれの項目について本システムを評価する。

学びやすさ

すぐに、そして簡単に使用することが可能かどうか

ユーザーテスト実行の手順として、操作説明を行う前にまずユーザーに自由に使ってもらった。この作業中に数人のユーザーは本システムの機能を発見し、自由に使いこなすこ

とができた。また、操作説明を行った場合でも、説明直後から操作に問題はなくスムーズにタスクに移行できた。このことから、「学びやすさ」においての評価は高い。

効率の良さ

学習後は高い生産性を創出可能かどうか

効率の良さとは、学習後のユーザーが決められた作業をいかに速くきちんと終わらせるかを示す指標である。タスク2で評価した項目に関しては、本システムの「効率の良さ」は高いと評価できるが、現行のYouTubeサイトは多様な機能を提供しており、本システムはそれらを全て実現できていない。多機能性を実現すれば、同時に「学びやすさ」「記憶のしやすさ」の評価を維持するのは難しく、トレードオフの関係にある。ある特定の作業で本システムは効率が良いが、記述式アンケートにおいても機能性に関するものが多くみられており、全体的にみれば従来システムに比べ「効率の良さ」は低下すると言える。

記憶のしやすさ

簡単に使い方を記憶することが可能かどうか

タスク遂行中にユーザーが操作について再び質問を行うということにはなかった。また、タスク1からタスク2への移行時に数分のタスク説明を行ったのであるが、それでもユーザーが操作方法を忘れるということにはなかった。これより、「記憶のしやすさ」においての評価は高い。

間違いにくさ

間違いを起こしにくく、また起こしても簡単に回復可能かどうか

上下ボタンの操作を誤るユーザーが数名あった。また、間違いが発生した時には正しい操作を行うことができたが、時間がたつと再び同じ間違いを起こすことがあった。このことから、「間違いにくさ」は「学びやすさ」、「記憶のしやすさ」に比べ評価は低い。

主観的満足度

ユーザーが満足できるよう楽しく利用することが可能かどうか

自由記述アンケートにおいて、楽しい、おもしろい、かっこいい、新鮮といった感性的なコメントをしたユーザーの割合が高いこと、タスクにおいて検索時間が飛躍的に伸びる頻度が高かったこと、逆に検索時間が極端に伸びる頻度が少なかったこと、などを踏まえると、「主観的満足度」の評価は高い。

以上、本システムの評価を通して考察したことをまとめる。

タスク1では、本システムの

- ・ 上下ボタンインターフェースの操作性
- ・ 詳細情報の確認のしやすさ

が従来システムと同程度であることを考察した。

タスク2では、本システムの

- ・ 3次元の高さ方向に軸をとり、再生回数の多い動画を視覚的に目立つように表示させたことの有効性
- ・ 全体俯瞰エリアを設け多くの情報を一度に表示させたことの有効性

が従来システムに比べ高いパフォーマンスを示すことを考察した。

自由記述アンケートでは、本システムにおける以下の課題を考察した

- ・ 操作における仕事量の多さ → 入力インターフェースの改善
- ・ 表示される文字情報(詳細情報)の少なさ

第5章 結論

本研究では、Web API を用いてインターネット上のマルチメディア情報に付属する XML 形式のメタデータを WEB3D で可視化するシステムを開発・評価した。その際、第3章では音楽、第4章では動画を扱った。様々な属性値をもつ多次元のマルチメディア情報に関しては、3次元空間での可視化の方がより多様な属性を表現できると考え、そのメタデータである XML 情報の可視化に3次元表示がどのように有効であるかを調査・考察した。特に YouTube の API を用いて動画情報を可視化したシステムではユーザテストを通じ、以下のようなことを示した。

- ・ 検索対象が明確である場合においては2Dでの文字情報の確認が必須であり、その点において従来システムと同程度の性能を保つことを示した。
- ・ 検索対象が曖昧である場合においては確認すべき対象が膨大であるためデータの特徴を素早く把握できることが重要であり、その点において従来システムに対して優位であることを示した。

それにより複雑な XML 情報を効率よく表示させるためには、以下のようなことが重要であることが明らかとなった。

- ・ 扱う文字情報の量や表示領域に留意し、3次元表示と2次元表示を共存させること
- ・ 3次元表示でデータの特徴を俯瞰的・視覚的に示すこと

以上のことを考慮し、3次元表示のインターフェースとしての有効性を最大限に引き伸ばすことで、今後多様化し続ける Web 上の情報に対して3次元表示が十分応用可能であることを示したと言える。

今後、WEB3D がその有効性を発揮し発展していくためには、第4章で挙げた課題である、文字情報の扱い方・インターフェースのデザインに加え、開発の難しさを克服するこ

とが必要である。本研究は、API と連動したシステムにおいて WEB3D を用いて実装を行った最初の例であるが、サーバ上でリアルタイムに MTX ファイル(WEB3D 用ファイル)を書き出すためのコーディングには非常に頭を悩まし時間を費やした。2D 座標から 3D 座標へレイアウトの幅が拡張したことでアニメーションなどの動的な表現を行うためには数学的な発想が必要であり、一般開発者にとっては大きな負荷となる。

このようにまだまだ課題の残る WEB3D ではあるが、大きな可能性のある分野であり、これからの発展に期待したい。

参考文献

- 1) 増井俊之. 情報の科学と技術, 情報科学技術協会. 54 卷 11 号. 558~567. 2004.
- 2) Tim O'Reilly. What is WEB2.0, Design Patterns and Business Models for the Next Generation of Software. 2005.
- 3) Google Maps, <http://maps.google.com>
- 4) Amazon, <http://www.amazon.com>
- 5) Simple API, <http://www.simpleapi.net>
- 6) Google Maps×Hotpepper,
<http://www.ussy.info/weblog/2007/09/google-mapshotpepper.html>
- 7) J. D. Mackinlay, G. G. Robertson, S. K. Card, The Perspective Wall: Detail and Context Smoothly Integrated; Proceedings of the ACM Conference on Human Factors in Computing Systems. 173-179. 1991.
- 8) J. L. Steffen, S. G. Eick, E. E. Sumner Jr, Seesoft - A Tool for Visualizing Line-oriented Software Statistics; IEEE Transactions on Software Engineering. 957-968. 1992.
- 9) G. G. Robertson, J. D. Mackinlay, S. K. Card, Cone Trees: Animated 3D Visualizations of Hierarchical Information; Proceedings of the ACM Conference on Human Factors in Computing Systems. 189-194. 1991.
- 10) Brian Johnson, Ben Shneiderman, Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures; Proceedings of the 2nd International IEEE Visualization Conference. 284-291. 1991.
- 11) John Lamping, Ramana Rao, Peter Pirolli, A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies; Proceedings of the ACM Conference on Human Factors in Computing Systems. 1995
- 12) mixiGraph, <http://www.fmp.jp/~sugimoto/mixiGraph/>
- 13) Ramana Rao, Stuart K. Card, The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular

Information: Proceedings of the ACM Conference on Human Factors in Computing Systems. 318-322. 1994.

- 14) J.Cuigini, C.Piatko, S.Laskowski, Design of 3D visualization of search result; NIST. 1999.
- 15) G.Robertson, M.Czerwinski, K.larson, D.Robbins, D.Thiel, M.van Dantzich, Data Mountain: Using Spatial Memory for Document Management; Proceedings of the 11th annual ACM symposium on User interface software and technology. 153-162. 1998
- 16) SUBARU「Forester」3 D ビジュアルシミュレーター,
<http://www.subaru.jp/forester/forester/exterior360/index3d.html>
- 17) 日興コーディアル証券 「イージー 3 D 証券ビューワー」,
http://www.nikko.co.jp/eztrade/report_news/ez3d/index.html
- 18) iTunes, <http://www.apple.com/jp/itunes/>
- 19) OpenOffice, <http://www.openoffice.org/>
- 20) Google Docs & Spreadsheets, http://www.google.com/google-d-s/hpp/hpp_co_jp.html
- 21) Viewpoint, <http://www.viewpoint.com/technologies/enliven.shtml>
- 22) J. J. Gibson. 生態学的視覚論. サイエンス社. 1986
- 23) YouTube, <http://www.youtube.com>
- 24) YouTube Wikipedia, <http://ja.wikipedia.org/wiki/YouTube>
- 25) 池谷祐二. 進化しすぎた脳. 朝日出版社. 2004
- 26) J.Nielsen. Usability Engineering. Academic Press. 1993
- 27) ISO 9241. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11:Guidance on Usability. 1998

謝辞

本研究は様々な方々の支援や助言によって成り立ちました。論文を書き終えた今、感謝の気持ちでいっぱいです。

主に指導をしていただいた岡本先生には、最後まで暖かく見守っていただきました。お忙しいところ週に1度、欠かさず研究室会を開いていただき、研究に対してコンスタントにモチベーションを維持することができました。モノを作ることが好きな性分のため、研究の目的や意義を見失い突っ走ってしまう自分を鋭い指摘で軌道修正していただきました。そういった一貫した視野での物事の見方は社会人になってからも忘れずに持ち続けたいと思います。また、最終的にこのような研究ができて本当に幸せでした。全く新しいことに挑戦することができ、気持ちの高ぶりを抑えることができませんでした。本当にこのような機会を与えていただき感謝申し上げます。2年間本当にありがとうございました。

染矢准教授には、研究における指摘もさることながら研究を快適に行うための環境も提供していただきました。共有ハードディスクの整備や被験者への祝儀の手配等も面倒くさがらずに実行していただき、何事も遠慮することなく相談できたことは自分の研究成果を実りあるものにできた大きな要因だと思っています。研究室旅行や飲み会なども染矢先生のおかげで楽しむことができました。2年間大変お世話になりました。

岩田研究室の岩田先生には1年の頃、何をしてもよいかわからない自分と積極的に議論していただき、本当に感謝しています。先生の幅広い知識や研究に対するストイックな姿勢に自分の甘さが認識されると共に非常に刺激的でした。

岡本研のリーダーである篠原さんを始め、飯田さん、桑原さん、ネジェットさん、ジョンさん、ウディンさん、岩丸君、吉田君、倉君とは楽しい時間を過ごすことができ、研究生活を有意義にすることができました。また、1つ上の先輩である坪内さん、朝倉さん、川崎さんには自分の研究を発展させる上での非常に有用な成果を残していただき感謝して

おります。

また、同期の浅野君、王子君、野田君、大原君とは研究仲間である以上に友達として確かな信頼関係を築くことができました。卒業後もお互いの活躍を見守りたいと思っています。2年間お世話になりました。そしてこれからもよろしく。

最後に、東京での生活を経済的にも精神的にも支援してくれた家族のみんなに感謝の意を示し、以上をもって謝辞とさせていただきます。皆様、誠にありがとうございました。

平成20年2月7日

藤井 公輔