

## 動体検知と顔検知を組み合わせた全方位動画 データベースシステム

渡邊 俊哉<sup>†</sup> 渋谷 進<sup>†</sup>

概要：監視システムでは、過去の膨大なサイズの動画から必要な動画のみを検索する必要がある。本研究では、監視目的を想定して、全方位カメラによる動体検知、顔検知の結果を、動画とともに保存、検索できる動画データベースシステムの構築を行う。全方位カメラからの入力画像に対して動体検知と顔検知を行い、その結果を日時とともにデータベースに保存する。これにより、動体検知と顔検知の結果を条件検索することができる。この設計に従いシステムを構築し、評価を行った。その結果、条件検索を行うことによって、全データ検索よりも検索時間を短縮することができた。

## Omni-Directional Movie Database System with Motion and Face Detections

Toshiya Watanabe<sup>†</sup> and Susumu Shibusawa<sup>†</sup>

Abstract : In the security monitoring system, it is necessary to retrieve only aimed movies from huge-size past movies. In this study, we construct a movie database system with motion and face detections for omni-directional camera movies. This system conducts motion detection and face detection for omni-directional movies, and preserves the results with the date, which enables clients to do the conditional retrieval. According to this design, we constructed a database system and evaluated it. As the results, the conditional retrieval time is shorter than the retrieval time for the total database.

### 1. はじめに

近年、防犯や防災、施設の監視などの用途で監視システムが用いられることが多くなってきており、いくつかの研究が行われている<sup>1),2)</sup>。監視システムでは広い範囲を監視す

る必要があるが、通常は定点カメラが用いられており、一台当たりの撮影範囲が狭い。景観配信システムや、監視システム<sup>3)</sup>、ウォークスルーシステム、知能ロボット<sup>4)</sup>などに、周囲 360° を一度に撮影できる全方位カメラが用いられている。全方位カメラは、監視システムのような周囲の映像を撮影するのに必要なカメラの台数を抑えることができ、またカメラの撮影方向を制御する必要がない。

他方で、コンピュータのディスク容量が大幅に増加し、大量の動画を扱うデータベースシステムについての研究<sup>5),6),7),8)</sup>が増えてきた。監視システムでは過去に撮影された動画を後から検索する必要があるため、あらかじめ検索に必要なメタデータをデータにつけたデータベースを構築する必要がある。監視システムでは、そのメタデータに顔検知や動体検知を用いることで、人が映っている動画や、動きのある動画を検索することができる。

本研究では、監視に必要な動画の検索を容易にできるようにすることを目的とした、動体検知と顔検知を組み合わせた全方位カメラ動画データベースシステムを構築する。構築するシステムはクライアントサーバシステムであり、全方位カメラ、サーバ PC、クライアント PC から成り立っている。このシステムでは、サーバにおいて全方位カメラからキャプチャした映像に対して顔検知と動体検知を行い、その結果を動画撮影した日付時刻とともにデータベースに保存する。この結果、データベースには撮影時の時刻情報と顔検知、動体検知の結果、ファイル名がメタデータとして保存される。クライアントはデータベースに接続し検索する。その中から見たい動画をサーバに要求、サーバはクライアントにその動画を配信する。

今回作成したシステムを評価するため、評価実験を行った。評価実験では顔検知の性能と、大量にデータを保存したときのデータベースの検索時間を測定する実験を行った。今回実装した顔検知機能による顔の検知率は、顔の角度によっては検知できる距離が減少した。また、一日分の動画のデータサイズは 87.16GB であり、データベースの全データ検索時間は、データ数が増えるごとに線形的に増加した。しかしながら、検索条件を用いた検索では検索されたデータ数自体が減少するため、結果的に検索時間が短縮された。

本報告の構成は以下の通りである。第 2 節で関連研究、第 3 節でシステムの設計、第 4 節でシステムの実装、第 5 節で実験について述べ、最後に第 6 節でまとめを述べる。

<sup>†</sup> 茨城大学 工学部

## 2. 関連研究

監視システムについては、以下のような研究が行われている。小林らは、パーティクルフィルタを用いて視野を共有する複数カメラでの三次元空間における人物の実時間追跡を行っている<sup>9)</sup>。パーティクルフィルタとは、状態量とゆら度をもつ多数の仮数群により離散的な確率密度として追跡対象を表現し、動きや観測のノイズに対して頑健な追跡を実現する。この手法に Haar-Like 特徴<sup>10)</sup>を用いた AdaBoost 学習<sup>11)</sup>による検出器を応用することにより、高精度に人物の頭部追跡を行っている。鈴木らは、カメラ、GPS、レーザレーダなどから得られる人物動線データ群から逸脱行動人物検出及び行動パターンの分類を行っている。HMM(Hidden Markov Model)を用いて人物動線データをモデリングし、パターン分類を行っている。

横矢らは、複数の全方位カメラを用いて移動物体の位置推定及び追跡が行えるシステムを研究している<sup>12)</sup>。カメラをサーバ側、監視者側 PC をクライアント側としたクライアントサーバシステムであり、ネットワークを利用してクライアント側で画像、物体検出情報の一極集中遠隔監視が行える。

NeoFace<sup>13)</sup>という製品の中で自然な言葉を入力することによって、多数の監視カメラによって撮影された大量の映像から、服や顔の情報により特定の人物を高精度に発見できる人物検索技術が開発されている。従来の技術では、服装をもとにした人物検索では服の色として代表的な色を入力するか、あるいはその服の画像を入力する必要があったが、NeoFace では入力された自然な言葉を画像の特徴データに変換し検索を行うことができる。

## 3. システム設計

### 3.1 動体検知の手法

今回実装する背景画像の時間的な変化に考慮した動的背景更新の手法は、参考文献<sup>12)</sup>に用いられている。この手法では、背景領域画素の輝度  $I$  を以下のようにモデル化する。

$$I = \bar{I} + \sigma \sin(2\pi\omega t) + k\zeta \quad (1)$$

上の式は入力画像中の各背景画素で成立し、 $\bar{I}$ は輝度値の時間平均、 $\sigma$ は輝度の振幅、 $\omega$ は輝度の周波数、 $t$ は時間、 $k(-1 \leq k \leq 1)$ は係数、 $\zeta$ はカメラのみに依存するノイズの最大値をそれぞれ表す。このモデルにおいて、 $\bar{I}$ が  $I - \sigma - \zeta \leq \bar{I} \leq I + \sigma + \zeta$  の場合に、その画素は背景領域に存在する画素であると判断する。

背景と判断された領域では輝度平均値  $\bar{I}$  と振幅  $\sigma$  を以下の式を用いて更新する。

$$\bar{I} = \frac{n-1}{n} \times \bar{I} + \frac{1}{n} \times I \quad (2)$$

$$\sigma = \frac{n-1}{n} \times \sigma + \frac{1}{n} \times \sqrt{2 \times (I - \bar{I})^2} \quad (3)$$

ここで  $\bar{I}$ 、 $\sigma$  はそれぞれ次フレームの  $\bar{I}$ 、 $\sigma$ 、 $n$  は更新速度パラメータである。この式は、ゆっくりとした背景の変化に対応するために用いられる。一方、物体領域と判定された領域では、輝度平均値は元の値を保持し、振幅  $\sigma$  のみを以下の式を用いて更新する。

$$\bar{I} = \bar{I} \quad (4)$$

$$\sigma = \frac{m-1}{m} \times \sigma + \frac{1}{m} \times \sqrt{2 \times (I - \bar{I})^2} \quad (5)$$

ここで、 $m(m \geq n)$  は物体領域更新速度パラメータである。この式は、撮影範囲内に静止物がおかれた場合や急激な背景の変化に対応するために用いられる。

### 3.2 システム構成

図 3.1 にシステムの構成を示す。システムはサーバ、クライアント、全方位カメラによって構成される。まず、サーバ上で動画撮影プログラムで動画とその動画情報をデータベースに保存する。クライアントはサーバ上にあるデータベースに検索を要求し、検索結果を表示する。検索結果からクライアントは見たい動画をサーバに要求し、配信された動画をサーバから受信する。

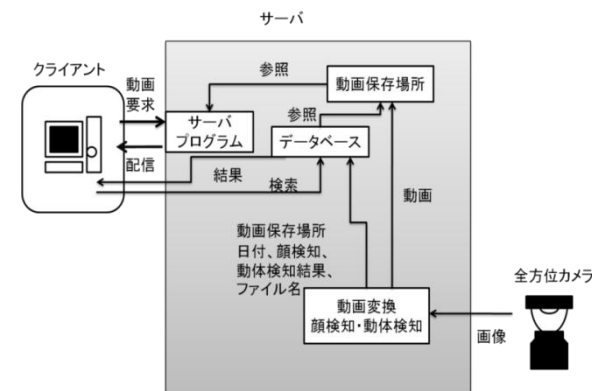


図 3.1 システム構成図

### 3.3 各プログラムの設計

#### 1) 動画撮影プログラム

動画撮影を行うとともにキャプチャした画像を利用して、顔検知処理や、動体検知の

処理を行う。動画として保存する際にファイル名、顔検知および動体検知の結果、日付とともにデータベースに登録する。図 3.2 にその処理を示す。

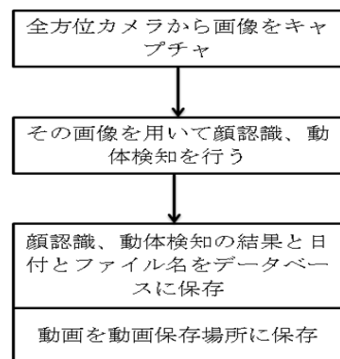


図 3.2 動画撮影プログラムの処理

## 2) サーバ・クライアントプログラム

サーバプログラムはクライアントからの要求を受け、動画の配信を行う。このプログラムではデータベースのデータを扱わない。クライアントは直接データベースに対して検索や結果の送信などを行う。

クライアントプログラムはサーバに接続し、検索要求を出し応答を受け取ることでよりデータベース内容を取得する。検索要求はサーバ上のデータベースに送信し、結果を受信する。動画の要求をサーバプログラムに送信し、動画の受信を行う。動画は円画像かパノラマ展開画像かを選ぶことができるようにする。

## 3.4 顔検知の手法

一般的なオブジェクト検出の流れを図 3.3 に示す。顔検知もこの手法を用いて行うが、今回はあらかじめ用意されている顔検出器を用いて顔検知を行う。そのため、学習フェーズの処理は行っていない。

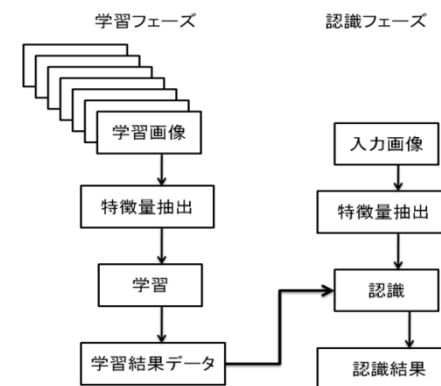


図 3.3 オブジェクト検出の流れ

## 3.5 データベース

データベースの動画データのフォーマットは、年,月,日,曜日,分,顔検知結果,動体検知結果,ファイル名となる。また、動画、動画データは 1 分ごとに保存する。検索は各項目間で and 検索とするが、顔検知と動体検知間のみ or 検索もできるようにする。

## 4. システム実装

開発環境は表 4.1 のとおりである。本システムの各プログラムの開発環境として Microsoft Visual C++ 2005, データベース環境に MySQL<sup>14)</sup>, 画像処理に OpenCV<sup>15)</sup>を利用した。

表 4.1 実装環境

	サーバ PC	クライアント PC
OS	Windows Vista Home Premium	Windows XP Home Edition Service Pack2
CPU	Intel Core 2 Quad CPU Q9450 @2.66GHz 2.67GHz	Intel Pentium4 3.00GHz
メモリ	4.00GB	512MB

## 4.1 サーバ

MySQL を用いてデータベースを作成する。あらかじめ、データ保存用のテーブルを作成しておく。作成する列の要素は年、月、日、曜日、時、分、秒、顔認識の結果、動体検知の結果、ファイル名とする。

### 4.1.1 動画撮影プログラム

入力部、検出部、保存部にわけてまとめる。

#### (1) 入力部

OpenCV の関数から直接全方位カメラに接続し、画像を取得する。取得した画像は、IplImage 型となる。画像は 15fps で取得する。

#### (2) 検出部

読み込んだ画像を用いて顔検知と動体検知の処理を行う。

##### ・顔検知処理

3.4 節で述べた方法で顔検知処理を行う。誤検知が多いことから、誤検知を防ぐ処理をする。検知した場合にその座標を控えておき、連続 3 フレームでその 10 ピクセル以内を検知した場合に顔を検知する、という処理にする。

##### ・動体検知処理

3.1 節で述べた動体検知処理を行う。ノイズが多いのでノイズを動体と検知しないように、平滑化した上で一定の閾値以上の面積をもった動体のみを検知するようにする。

#### (3) 保存部

MySQL のテーブルに接続し、あらかじめ保存しておいた動画の撮影開始時間と、検出した結果、ファイル名を保存すると同時に、動画を動画保存用のフォルダに保存する。MySQL のテーブルへの保存処理は SQL 文の INSERT 文を用いて行う。動画保存用のフォルダは、サーバプログラムの存在するフォルダとする。動画は OpenCV で読み込めるように Microsoft video1 により圧縮、また 15fps で保存する。

### 4.1.2 サーバ・クライアントプログラム

サーバプログラムが要求があった動画をクライアントに配信する。クライアントの要求は動画のファイル名であり、send 関数により送信される。動画の配信はソケット通信を用いて動画の画像データを一枚一枚 send 関数で送る処理を行う。クライアントプログラムはサーバから配信された画像を recv 関数で受信し、必要によりパノラマ展開して表

示する。

## 4.2 ユーザインタフェース

作成したクライアントプログラムのインタフェース部分を以下の図 4.1 に示す。ユーザが受信した動画像の例を図 4.2 に示す。

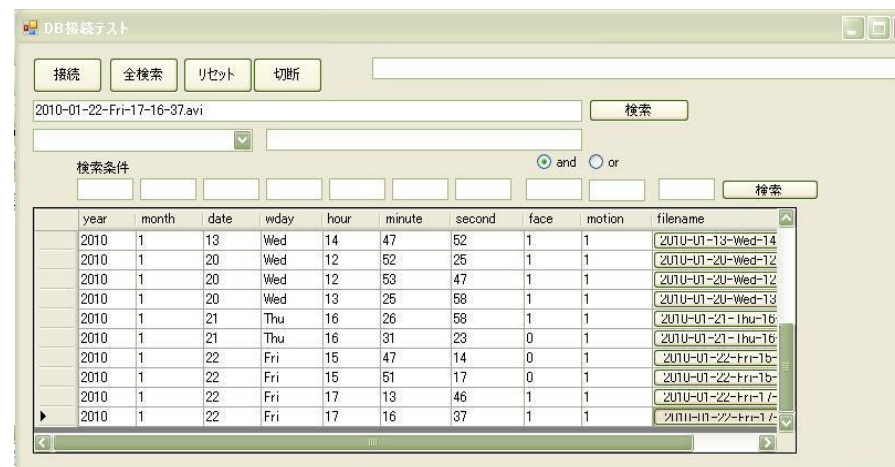


図 4.1 クライアントプログラム UI 部分



図 4.2 ユーザが受信した動画像の例

クライアントプログラムの機能は以下の通りである。

表 4.2 クライアントプログラムの機能

UI 操作	機能
接続ボタン	サーバプログラムに接続する
全検索ボタン	全データを検索する
クリアボタン	表示されているデータをクリア
切断ボタン	サーバプログラムから切断する
検索ボタン	検索条件入力部分に入力された条件で検索
検索条件入力部分	検索条件を各項目ごとに入力する
データ表示領域	検索されたデータを表示する
パノラマ展開チェックボックス	チェックを入れることにより配信された動画をパノラマ展開する

検索機能は、ユーザが各テキストボックス内に検索したい時刻などを入力することにより検索を行うことができるようにする。検索できる項目は年、月、日、曜日、時、分、秒、顔検知の結果、動体検知の結果、ファイル名である。テキストボックス内に入力された文字列は、sql 文に変換し MySQL に問い合わせ、結果を取得する。検索条件は and 検索であり、顔認識の結果と動体検知の結果のみ or 検索もできるようにする。これを表 4.2 に示す。

表 4.3 and, or での検索条件

年、月、日、曜日、時、分、秒、 ファイル名	顔検知	動体検知	顔検知と動体検知の間
and	and	and	and, or

## 5. 実験

本システムの評価をするため、評価実験を行った。評価実験では、顔検知機能の性能評価とデータベースに一定量のデータを保存したときの検索時間の測定を行う。

### 5.1 顔検知機能の性能評価

顔検知機能の性能についての実験を行う。手順を以下に示す。

- 1) 1 分間顔を映している状態で顔の検知回数を測定
- 2) 顔は正面と上下左右それぞれの方向に向けて測定する。顔の角度は 15° 単位で検知率が 0 になるまで傾けていく。
- 3) 距離は 20cm から 10cm 単位で 80cm まで測定する。  
 顔の検知率=顔の検知回数/全フレーム数で計算した。

結果と考察を以下に示す。図 5.1 に顔の各方向、各距離での検知率の図を示す。まず、正面では検知率がカメラからの距離 40cm から 50cm の間で大きく下がり、60cm を超えると検知ができなくなる。また、距離 20cm から 40cm の間はほぼ同一の検知率と考えられる。

図 5.1(a)に顔を上方に向けてときの角度と距離での検知率を表す。上方では 45° 以上の顔の角度では検知しなかった。他のどの顔の向きよりも検知率が高く、より大きい角度でも検知できた。これは上方に顔の角度を増やしても検知に使っている目と鼻近くの領域が隠れないためだと考えられる。

図 5.1(b)に顔を下向きにしたときの角度と距離での検知率を示す。顔が上向きするときとは違い、検知率が大幅に落ちている。これは、髪で顔が隠れて検知がしにくくなったためと考えられる。撮影される人の髪型によってはもう少し検知率が上がる可能性がある。

図 5.1(c), 図 5.1(d)に顔の向きが左、右方向のときの検知率を示す。左と右で比べた場合、検知率は通常同じような値になると考えられるが、実験結果では各角度での検知率は左向きのほうが高い傾向があった。この理由としては、左向きと右向きとで顔への光の当たり方が変わったためであると考えられる。

全体として、どの顔の向きでも検知できる限界の距離はカメラから 60cm 離れた程度の距離であり、顔の角度を大きくすると検知できる距離が減っていく傾向があった。これは距離を離すこと、顔の向きを変えることの両方で顔を認識できる部分の画素が減少するためだと考えられる。

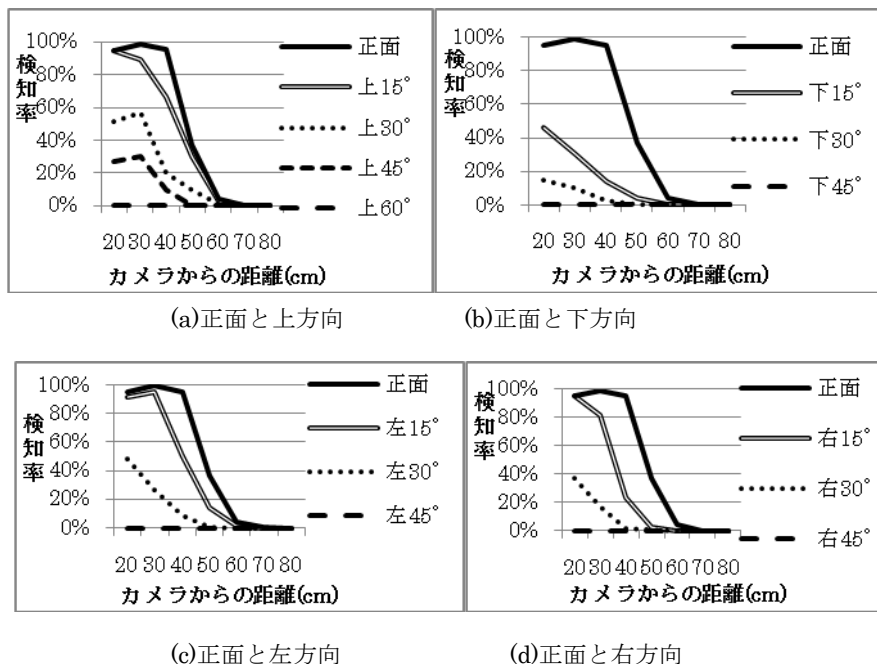


図 5.1 顔の各方向での検知率

## 5.2 動画データベースとしての性能評価

動画撮影プログラムを用いて以下の条件で 24 時間のデータの蓄積を行った。

- ・場所：研究室内
- ・24 時間連続で撮影

このとき、データ数は 1440 個、総動画データサイズは 87.16GB となった。動画保存の際に負荷がかかることがあり、撮影時間が数分に 1 回ずれることがあった。このことで動画の保存ミスは起きておらず、正常に再生できた。

以下の 2 項目の性能評価を行う。

- 1) クライアントプログラムでデータベースの全データ検索時間を測定する。データ数は 2 時間分のデータである 120 個から 120 個毎に 1440 個までの 12 パターンで行う。また、検索時間はそれぞれ 10 回測定し、平均をとる。

- 2) クライアントプログラムでデータベースの条件検索を行い、検索されたデータ数と検索時間を測定する。条件検索の条件は、顔検知、動体検知の項目を利用する。以下の表 5.1 に測定する検索条件の項目の組み合わせを示す。この評価は、全データが 1488 個のときに行った。

表 5.1 検索条件組み合わせ

検索条件 項目	なし		and 検索		or 検索	
	顔検知	動体検知	顔検知	動体検知	顔検知	動体検知
各要素 の値	1	—	1	1	1	1
	0	—	1	0	1	0
	—	1	0	1	0	1
	—	0	0	0	0	0

以下に結果と考察を示す。

- 1) 図 5.2 にデータベースの各データ数での全データ検索時間を示す。図より、検索時間は線形に近い形で増えているのが分かる。
- 2) 各検索条件でのデータ数と検索時間の相関を図 5.3 に示す。図 5.2 と比べてみると、ほぼ同じ線が描かれており、検索時間はデータ数によって線形的に増えている。また、各データ数での検索時間はほぼ等しい。このことから、検索条件を付けた検索でも検索時間は検索にヒットするデータ数によってほぼ一定である。実際には検索条件での検索では検索にヒットするデータ数自体が全データ検索に比べると少なくなるため、検索時間は大幅に短縮される。顔検知の項目が 1 の場合のみの条件で検索した場合の検索時間は 0.2 秒であり、全データ検索の 1.94 秒よりは大幅に短縮されている。

また、全データ検索でのデータ数、検索時間の関係から 1 データあたりにかかる検索時間は 0.0013 秒と計算できるが、実際には 0.20 秒かかっている。これは、データベースに接続するまでの時間の影響で検索時間が伸びていることが原因として考えられる。



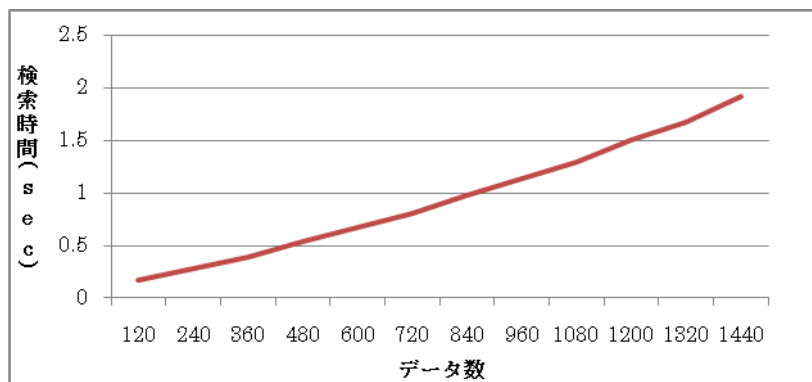


図 5.2 データ数と検索時間の相関

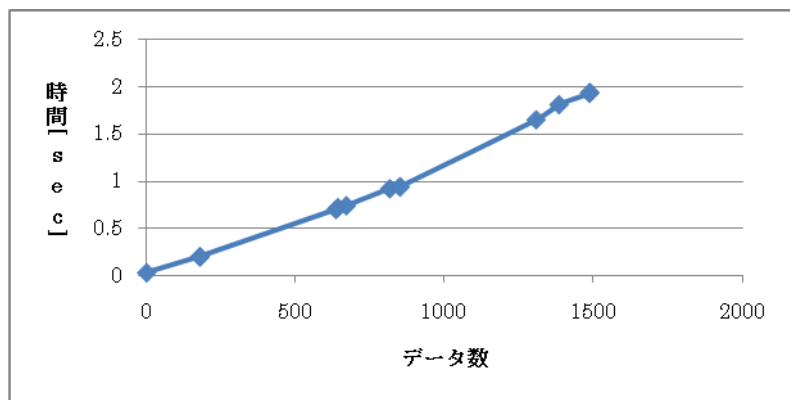


図 5.3 検索条件での検索データ数と検索時間の相関

## 6. おわりに

本研究では、動体検知と顔検知を組み合わせた全方位カメラ動画データベースシステムの構築を行った。また、顔検知の機能とデータベースについて性能評価を行った。顔の検知率では、正面以外の向きの顔の検知率が低く、検知する距離も短かった。データベースの評価では、データ数を少しずつ増加させていき、データベースの検索時間の測

定をした結果、データベースの検索時間がデータ数によってほぼ線形的に増加していくことがわかった。また、検索条件を使った検索でも、データベースの検索時間は検索されたデータ数に対してほぼ一定だった。しかし、検索条件を使った検索では、検索されたデータ数が減るため、結果として検索時間が減少する。

今後の課題として、人の顔の検知率をあげるために、顔検知のプログラムを改良する必要がある。また、肌色領域の検知や追跡を行い、さらにそれをデータベースの要素として保存することで人が動画に映っているということをより明確にすることが考えられる。これにより、監視目的の検索においてより有用になる。データベースについては、インデックス検索を利用することで更に検索時間を短くすることができると考えられる。また、動画撮影プログラムにおいては、動体検知の際に動体の数が多くなりすぎると処理が重くなり、結果として動画のフレーム数が少なくなってしまうことがあった。これについては処理の負荷を減らす工夫が必要だろう。また、複数の全方位カメラを利用できるようにし、さらに監視範囲を広げることも課題となる。

## 参考文献

- 1) 鈴木直彦, 平澤宏祐, 田中健一, 他: 人物動線データ群における逸脱行動人物検出及びパターン分類, 電子情報通信学会論文誌, Vol.J91-D, No.6, pp.1550-1560 (2008).
- 2) 川端聡, 日浦慎作, 佐藤宏介: 複数の未校正カメラを用いた三次元侵入検知システム, 電子情報通信学会論文誌, Vol.J91-D, No.1, pp.110-119 (2008).
- 3) 佐藤洋介, 米田裕也, 橋本浩二, 柴田義孝: 全方位カメラを用いた監視追跡システム, 情報処理学会研究報告, Vol.2007, No.58, pp13-18 (2007).
- 4) 近藤一晃, 八木康史, 谷内田正彦: 知能作業ロボットのための非等方性全方位視覚, 電子情報通信学会, D-II, Vol.J88-D-II, No.8, pp.1521-1529 (2005).
- 5) 何書勉, 横田裕介, 石黒浩, 上林弥彦: 全方位センサによる分散映像データベースのアーキテクチャとプロトコル設計, 第13回データ工学ワークショップ, B5-4 (2002).
- 6) 白井宏一, 椎名広光: XML データベース上における動画データベースへの文検索システム, 第18回データ工学ワークショップ, A7-8 (2007).
- 7) 三好茂樹, 内野権次, 石原保志, 他: インターネットを介した専門用語手話画像データベースの活用 筑波技術短期大学テクノレポート, Vol8, No.2, pp.17-20 (2001)
- 8) 渡辺陽介, 秋山亮, 大喜恒甫, 北側博之: 大規模映像ストリームのための映像情報統合基盤システムの提案, 日本データベース学 Letters, Vol.6, No.4, pp.13-16 (2008).
- 9) 小林貴訓, 杉村大輔, 平澤宏祐, 他: パーティクルフィルタとカスケード型識別器の統合による人物三次元追跡, 電子情報通信学会論文誌, Vol.J90-D, No.8, pp.2049-2059 (2007).
- 10) Lienhart, R. and Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection, *Proc. the*

2002 *IEEE International Conference on Image Processing*, Vol.1, pp.900-903 (2002).

11) Viola, P. and Jones, M.J.: Rapid Object Detection Using a Boosted Cascade of Simple Features, *Proc. the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.511-518 (2001).

12) 森田真司, 山澤一誠, 寺沢征彦, 横矢直和: 全方位画像センサを用いたネットワーク対応型遠隔監視システム, 電子情報通信学会, D-II, Vol.J88-D-II, No.5, pp.864-875 (2005).

13) NeoFace, <http://www.nec.co.jp/soft/neoface/product/neoface.html>

14) MySQL, <http://www.mysql.com/>

15) OpenCV, <http://opencv.jp/>