Vol.2013-DPS-154 No.54
Vol.2013-CSEC-60 No.54
2013/3/15

複数のギガビットネットワーク型全方位カメラを用いたシームレスな監視映像システムの研究

今 拓磨[†1]　内田 法彦[†2] 橋本 浩二[†3] 柴田 義孝[†3]

Research on Seamless Surveillance Video Monitoring System using by Multiple Omni-directional Gigabit Network Cameras.

TAKUMA KON[†1]　NORIKI UCHIDA[†2]
KOJI HASHIMOTO[†3] YOSHITAKA SHIBATA[†3]

1. はじめに

今日, 公共施設における凶悪事件や震災や災害など, 様々な用途にカメラを使った監視システムが普及している．特に防犯で使用する監視映像システムの需要が急増している．一方で広域な監視領域を必要とする場所も多くなり，監視システムに要求されてきている．しかしながら現行の監視システムに使用する監視カメラは CCD カメラを利用した単一方向性のカメラを使用しており，広域な監視を行うためには，複数の監視カメラを必要とする．機器のコストの増加につながる．また監視カメラ数の数が増えることによって機器構成が複雑化してしまう要因であると言える．

一方で，カメラ部の回転を行うことができる Pan/Tile/Zoom（PTZ）カメラが出現した．これにより広範囲の監視を可能になった．またカメラ部が動くことによって監視対象を追従しやすいといった特徴も持っている．しかしながら，カメラ部の回転には多少の時間が必要であるため, リアルタイムで追跡などの処理をすることが難しい．また PTZ カメラも瞬間的な撮影範囲は単一方向性の CCD カメラと同等であるため，視野角外の撮影ができず，怪しい人物を見逃す可能性があるといった問題を抱えている．

そして近年，カメラの中心から 360 ° 撮影することができる全方位カメラが出現した．これにより，常時 360 ° の視野で撮影することが可能になった．また PTZ カメラのような回転部を必要としなく小型軽量であるため，非常に安易に設置することができるようになった．しかしながらカメラの中心部分に死角が発生することや，特殊な画像処理が必要といった，問題を抱えている．

†1 岩手県立大学大学院 ソフトウェア情報学研究科
Iwate Prefectural University Graduate School
†2 埼玉工業大学 人間社会学部
Saitama Institute of Technology
†3 岩手県立大学 ソフトウェア情報学部
Iwate Prefectural University

一方で Gigabit Network を用いた Gigabit Ethernet カメラが出現した．これにより Gigabit Network を用いて高品質，高解像度の画像を送受信することが可能となった．

筆者らはこれまで全方位カメラと PTZ カメラを組み合わせた監視映像システム[1]を構築したが，PTZ のカメラ部が回転する遅延が起きるため，リアルタイム性に問題があった．また PTZ カメラの撮影範囲は限定されてしまうため, シームレスな動作が難しいといった問題を抱えていた．

一方で，Gigabit Ethernet カメラに全方位レンズを装着した遠隔テレビ会議システム[2]が構築された．これにより Gigabit Network を利用して遠隔地との映像を送受信することができるようになり，インフォーマルなグループワークが実現した．

そこで本論文では PTZ カメラを使用せず, Gigabit Ethernet カメラに全方位レンズを装着した全方位カメラを使用し，広域な監視を目的とした監視映像システムを提案する．また複数の全方位カメラを設置することでより広域な監視領域を確保し，シームレスな監視を行う．また動体の発見, 認識, 追従処理を行い, 監視システムの有効性を検証する．

2. システム概要

本論文で提案するシステム概要図は図 1 の通りである．

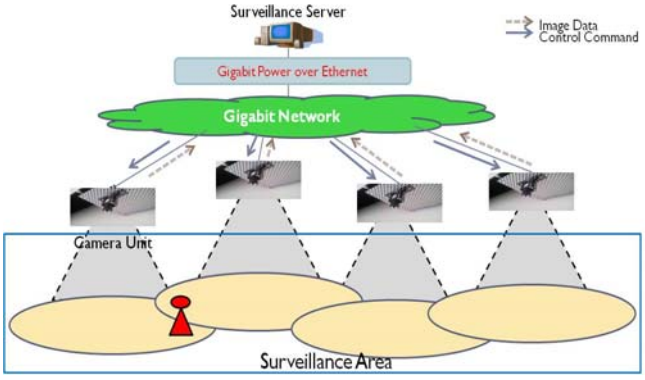

図 1. システム概要図

Vol.2013-DPS-154 No.54
Vol.2013-CSEC-60 No.54
2013/3/15

センサーカメラには小型軽量である Gigabit Ethernet カメラを導入し，Gigabit Ethernet カメラに全方位レンズを装着することで，全方位カメラとして使用する．また監視サーバーでは各センサーカメラから画像を取得し，画像処理を行う．センサーカメラと監視サーバーの間は Gigabit Network を構築することで，画像データの受信や各カメラへのコマンドを送信する．また複数台の Gigabit Ethernet カメラを連携することで連続した監視領域を持ち，シームレスな監視を行うことが可能となる．また各カメラへの電源は Power over Ethernet 機器を用いて供給するため，Ethernet ケーブル 1 本のみで電源の供給と通信路をサポートすることができるようになる．

次にシステムアーキテクチャについて説明する．本システムでは図 2 のように構築する．

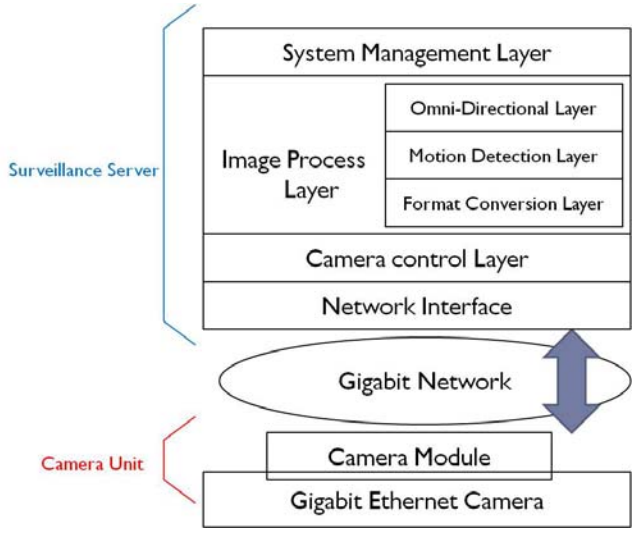

図 2. システムアーキテクチャ

本システムは 3 つの Layer で構築される．System Management Layer ではシステム全般の処理が行われる．Image Process Layer は主に画像処理について行われ，主に 3 つの Sub Layer に分けられる．Format Conversion Layer では画像フォーマットの変換処理が行われる．Motion Detection Layer では動体検出処理や認識，追従処理を行うための画像処理が行われる．Omni-Directional Layer では全方位カメラから取得した 360°の画像（環状画像）をパノラマ画像にするための変換処理を行う．最後に Camera control Layer では各センサーカメラの制御を行い，センサーカメラから画像を取得する．これらの処理を Gigabit Network を介してカメラを制御して行い，システムを処理する．

次に Image Process Layer 内における画像処理の流れについて説明する．本システムは図 3 のように画像処理を行う．

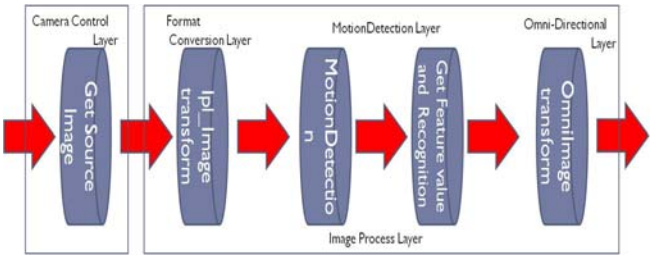

図 3. 画像処理フロー

最初に Camera control Layer 内において各センサーカメラから画像を取得し，抽出処理を行う。また画像データは Image Process Layer 内の Format Conversion Layer に送られる．Format Conversion Layer ではカメラで取得した画像フォーマットの変換処理を行う．その後，Motion Detection によって動体検出処理を行う．動体検出処理は背景差分法を用いて行われる．背景データは一定期間ごとに更新される．Motion Detection によって動体が検出された場合，動体認識，追従処理を行う処理が行われる．ここでは動体画像から特徴値を算出し，それを元に追従処理を行なっていく．特徴値の算出方法はカラーヒストグラムなどのアルゴリズムを用いて行われる．また管理者が画像を確認する際には 360°の環状画像をパノラマ画像へ展開する処理を OmniImage transform 内で行われる．このような流れで本システムは処理を行なっていく．

3. 全方位展開処理

本研究では全方位カメラを利用しており，周辺 360°で撮影された環状画像を取得する．図 4 は実際に取得された環状画像である．

図 4. 環状画像

通常の動体検出処理や認識処理は環状画像で行うが，管理者が画像データを確認する際にはパノラマ展開処理を行い，パノラマ画像を出力するようにしている．パノラマ展開処理は Image Process Layer の Omni-directional Layer 内で行われる処理である．パノラマ展開処理には本研究室で開発された全方位ミドルウェア[3]を使用することで図 5 のようなパノラマ画像を出力する．出力されるパノラマ画像は 2500×250 の解像度で出力される．

Vol.2013-DPS-154 No.54
Vol.2013-CSEC-60 No.54
2013/3/15

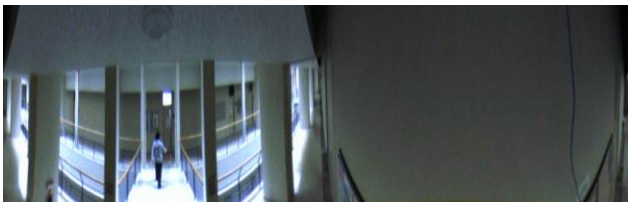

図 5. 出力されたパノラマ画像

4. 画像抽出処理

本研究で使用する環状画像は外角部分の死角が多く，画像処理を行う際に，処理時間が増大する原因を生む．そのため，フレームレートが落ちてしまい，システム全体に影響を及ぼす．そこで環状画像の必要な部分を切り抜く処理を行う．本研究では全方位カメラに合わせて 1080×1080 の画像を抽出し，その画像を用いて動体検出を行う．

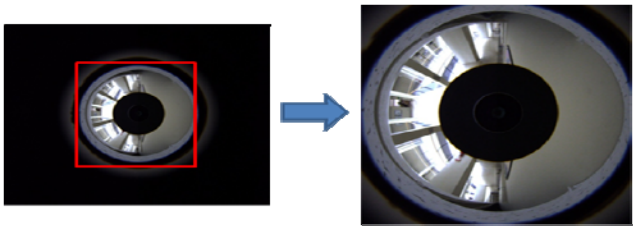

図 6. 死角部分を取り除いた環状画像

5. 動体検出処理

画像処理は最初に動体検出処理から行われる．動体検出処理は Image Process Layer 内の Motion Detection Layer で行われる処理である．動体検出処理は背景差分法を用いて行われる．ここで使用する背景データは一定時間で更新される．背景差分法により差分部分が発生した場合，差分部分の動体領域の推定を行う．その後，ノイズ判定処理を行い，ノイズ以外の部分を動体として判断する．図 6 は抽出された画像データである．

図 7. 抽出された画像データ

１つのセンサーカメラで複数の動体を検出した場合は，それぞれの画像を検知，抽出処理を行う．この処理は毎フレーム行われるので，連続した動体を取得することができる．

動体を検知したのち，抽出した画像データを元に認識処理を行う．抽出した画像データにはそれぞれに識別子を付加する．認識処理は画像データから特徴値を取得し，その特徴値を元に認識処理を行う．特徴値を取得したのち，直前に取得した特徴値との比較処理を行う．比較処理の結果，同じ動体であると判断した場合には，画像情報を更新する．比較処理の結果，違う動体であると判断した場合には新しい識別子を付加し別データとして使用する．図 7 は実際の識別処理の流れ図である．

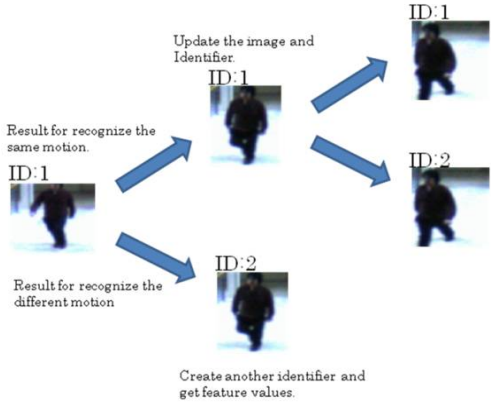

図 8. 認識処理図

認識処理時に使用する特徴値の算出する方法は次の４つを想定している．

1) カラーヒストグラム
2) オプティカルフロー
3) 移動方向取得処理
4) エピポーラ幾何学

カラーヒストグラムは色情報を用いたアルゴリズムであり，２つの画像データの色情報を比較し，その結果を特徴値として取得する．比較時の処理尺度は相関関数を用いて行う．

$$d(H_1,H_2)=\frac{\sum_I(H_1'(I)\cdot H_2'(I))}{\sqrt{\sum_I(H_1'(I)^2)\cdot\sum_I(H_2'(I)^2)}}$$

$$H_k'(I)=\frac{H_k(I)-1}{N\cdot\sum_J H_k(J)}$$

相関関数式

相関関数の結果が+1,0 に近づくほど 2 つの画像の色情報は類似しており，-1.0 に近づくほど 2 つの画像の色情報は相違している．この結果を特徴値として使用する．

オプティカルフローは 2 つの画像の運動ベクトルを計算する手法であり，運動ベクトル値の平均値を特徴値として取得する．Optical Flow を用いる際に，ブロックマッチング法を用いて処理を行う．

$$S(p,q)=\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|I_{t+1}(x+i,y+j)-I_t(x+i+p,y+j+q)\right|$$

ブロックマッチング法

Vol.2013-DPS-154 No.54
Vol.2013-CSEC-60 No.54
2013/3/15

ブロックマッチング法は画像をブロック単位で領域を分割し，ブロック内での差分が最小のブロックを検索する方法である．ブロックマッチング法は明るさの変化に強く，ノイズに強い特性を持っているため，特に照明が不安定な環境下で使用する本システムにおいて，最も適していると考えられる．

図 9. OpticalFlow

移動方向取得処理は動体の移動方向を数フレームかけて取得し，８軸の移動軸によって移動方向を特徴値として取得する処理である．動体の中心座標を計算し，次点の中心座標位置を求めることで移動方向の軸を決定する．

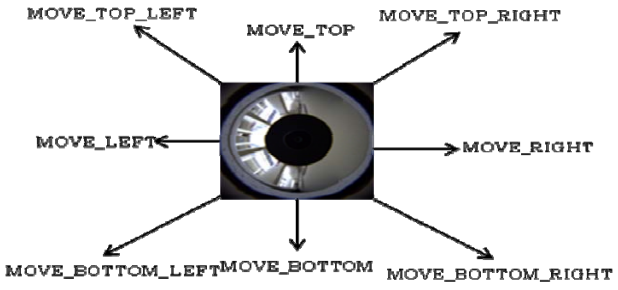

図 10. 8 軸の定義

エピポーラ幾何学は 2 つのカメラからオブジェクトの座標位置を投影行列を用いて計算する手法である．条件として，２つ以上のカメラが動体オブジェクトをとらえている必要があるが，本システムでは全方位カメラを使用しているため，広範囲の監視が可能であり，２つ以上のカメラが動体オブジェクトをとらえていることが多いため，エピポーラ幾何手法を検討している．

今回，カラーヒストグラム，オプティカルフロー，移動方向取得処理を実装し，特徴値として算出している．また認識処理の評価実験を行い，認識，追従できるかどうか，またその有効性の検証を行なっている．

6. 動体追跡処理

本研究では複数の全方位カメラを用いてシームレスな動体の検知，及び認識，追従を目標としている．そのため動体がカメラ間を移動しても追跡できるよう，カメラ間での協調処理を行う必要がある．そこで本研究では次のようにカメラ間の協調処理を行う．

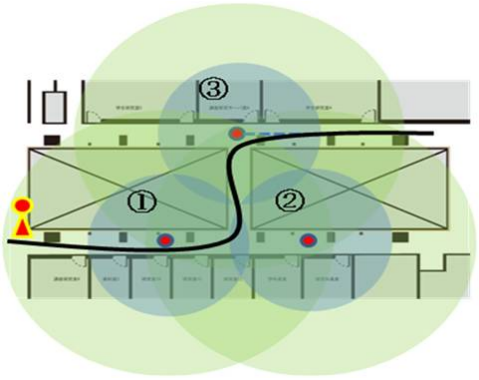

図 11. 動体追跡イメージ図

図 11 は複数のカメラを設置したイメージ図と黒い線を歩く動体のイメージ図である．動体は最初①のカメラの検知範囲に移動すると，①のカメラによって動体検知が行われる．①のカメラで動体を検知したのち，システムは動体の抽出処理を行い，各認識処理を行う．認識処理によって取得された特徴値はシステムで保持され，また各特徴値は画像が取得されるたびに更新されるため，変化が起きても柔軟に対応できるようにしている．

動体が①の検知範囲を抜け，③の動体検知範囲に移動したとする．③のカメラで動体検知を行い，認識処理によって特徴値を算出する．この時に，システムで保持している特徴値との比較処理を行う．例えばカラーヒストグラムでは①のカメラで撮影された最後の画像と，③のカメラで撮影された最初の画像との比較処理を行い，その結果が閾値以上であれば同一人物としてみなす．比較処理の結果，同一人物であると判断した場合には，特徴値を更新し追跡を行う．比較処理の結果が同一人物ではないと判断した場合には新しい情報として取得しシステム内で保持する．この処理が各カメラの動体検知範囲外に移動するまで繰り返すことで連続した同一人物の追跡処理を行うことができる．

7. マップモニタリング処理

検知された動体などについてはマップ上で管理者に通知する．マップ上のカメラは管理者によって追加され，識別される．動体が検知された場合には管理者に通知する．

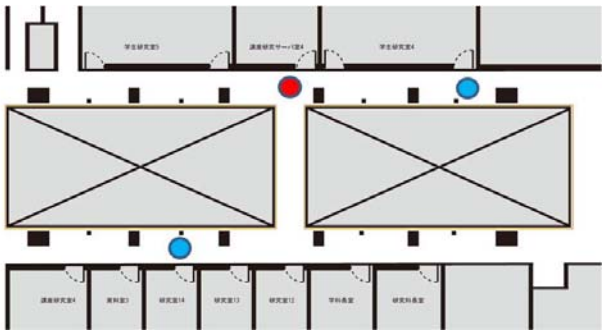

図 12 マップモニタリング図

カメラからの画像情報を元に，将来的には動体の軌跡等をマップ上に表示することで，同一人物の人物をマップ上で監視するようにしたいと考えている．

8. プロトタイプ構成

本研究で使用するプロトタイプ構成は図 13 の通りである．

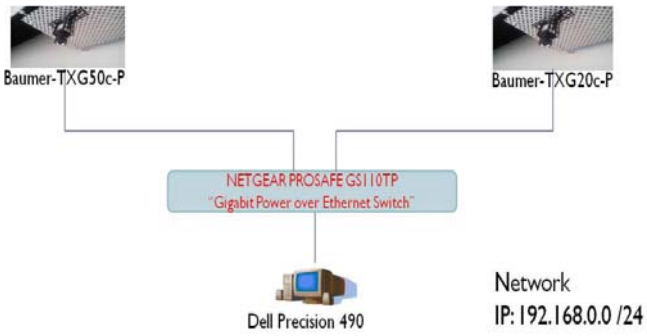

図 13. プロトタイプシステム構成

プロトタイプシステムは 2 台のカメラと 1 つの監視サーバーで構築している．監視カメラに使用するカメラは Baumer 社の TXG20c と TXG50c を使用している．TXG20c は 1624×1232 の解像度で最大フレームレートが 16fps である．TXG50c は 2448×2050 の解像度で最大フレームレートは 15fps である．また監視サーバーには 1Gbps を利用できる Network Interface Card を使用することで Gigabit Network を構築することができる．監視サーバーは Microsoft Windows で動いており，開発言語に Visual C++を使用している．各カメラへの電源供給は Power over Ethernet Switch を用いることで Ethernet ケーブルから電源を供給することができる．

9. 評価実験

提案した手法の有効性を検証するため，プロトタイプシステムを用いて評価実験を行った．評価実験では各認識処理の測定を行い有効性を検証した．また監視サーバーの負荷の測定を行い，動作時のどの程度の負荷がかかるか測定を行った．

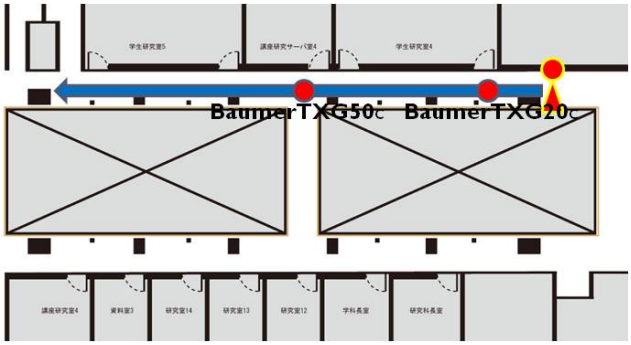

図 14. 動体の移動手順

被験者には図 14 のように，設置したカメラ 2 台を通るように歩いてもらい，各カメラで検知した情報を監視サーバーで取得するように設定を行った．認識処理の実験はカラーヒストグラム，オプティカルフロー，移動方向取得処理の 3 つの特徴値の評価を行い，監視サーバー負荷測定では，各認識手法の処理時間の測定と CPU 使用率を測定した．

i) カラーヒストグラム評価実験結果

カラーヒストグラムは 2 つのカメラで相関関数による比較処理を行い，その結果を取得した．その際に閾値を 0.65 に設定し，閾値以上であれば同一人物とみなした．動体のカメラ間の移動が起きた場合にはその前のカメラで最後に取得した画像と現在のカメラで最初に取得した画像との比較処理を行い，閾値以上であれば同一人物として追跡を行うよう設定した．図 14 は実際にカメラ間の比較時に使用した画像データである．

図 14 カメラ間移動時に使用した画像データ

これらの画像を踏まえ，同一人物の連続した 24 フレームを計算した結果，図 14 のグラフの結果になった．

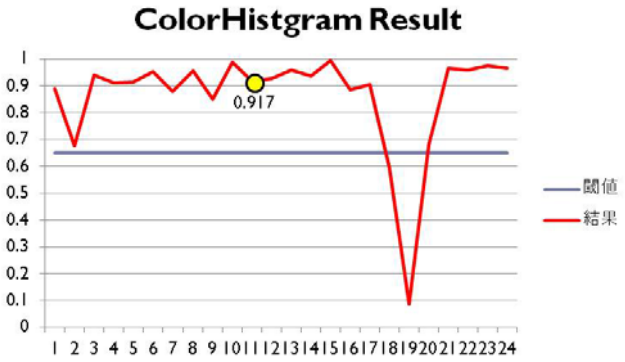

図 15. カラーヒストグラム結果

実験の結果，全 24 フレームのうち，著しく結果が落ちたのは 3 フレームになり，残りは同一人物として判断することができた．また動体のカメラ間の移動が起きた時の比較処理の結果は 0.917 という値になり，閾値以上にであるため同一人物として判断された．これにより，照明が安定している環境下であればカメラ間の移動が起きても同一人物として認識することが可能であることがわかった．しかしながら動体検知処理によって抽出された画像データによっては誤認識を起こす可能性もあることがわかった．著しく結果が悪かった画像データは，図 15 のように頭部のみ写っ

Vol.2013-DPS-154 No.54
Vol.2013-CSEC-60 No.54
2013/3/15

た画像が使用されているため精度が落ちたと考えられる．そのため，画像データの面積を計算するなどして，誤認識を起こす可能性の画像は除外する処理が必要である．また柔軟な変化に対応できるよう，取得する画像のフレームレートが高いほうが精度が高くなることがわかった．

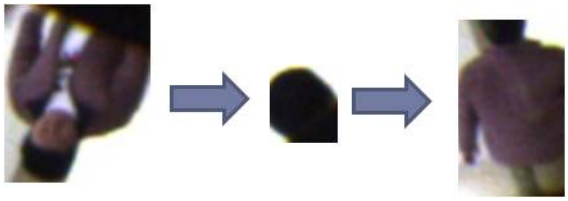

図 15. 誤認識を起こした画像データ

ii)　*オプティカルフロー評価実験結果*

オプティカルフローでは動体検知した環状画像に対してブロックマッチング法による処理を行い，運動量の平均値 X,Y を算出した．人物が一定の速度で歩くと仮定し，運動量平均値のブレが小さいほど同一人物を示す結果になると考えている．図 16，図 17 は各カメラの運動量をグラフに表した図である．

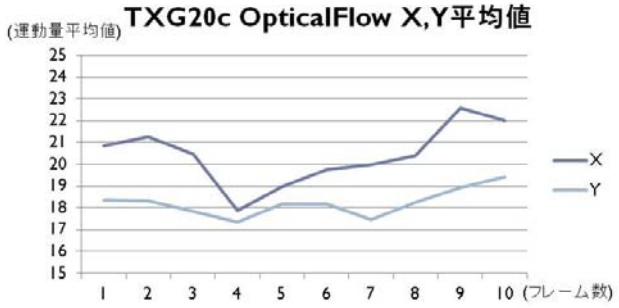

図 16. TXG20c のカメラによる運動量平均値の結果

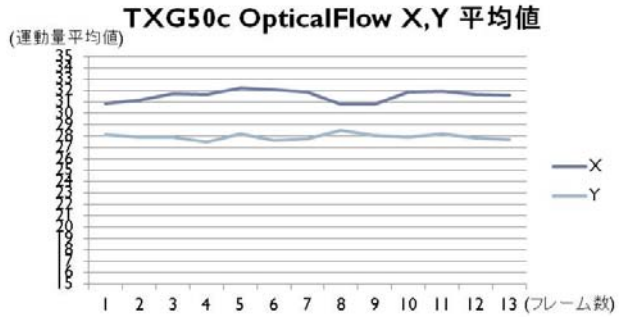

図 17. TXG50c のカメラによる運動量平均値の結果

上記のグラフでは TXG20c のカメラのほうについて運動量の平均値のブレが大きく，同一人物を結びつけることは難しいと考えられる．またこの結果は画像のノイズ情報を含んだ結果であるため，正しい結果であるとは言えないことがわかった．理由としては照明の変化が起きるとその部分に過敏に反応してしまうためである．図 18 は実際にブロックマッチング法を使用した際に誤認識を起こしている画像である．この画像でも分かる通り，特に輝度が高い場所に対して過敏に反応してしまい，その結果が含まれてしまうため正しい結果であるということが言えない．本研究で想定してる，通路や階段等ではこのような変化が常に起こる環境下であるため，本研究の認識処理には対応させることが難しいことがわかった．またノイズはカメラ自身やレンズによって発生しているため，ノイズを除去する処理が必要であることがわかった．

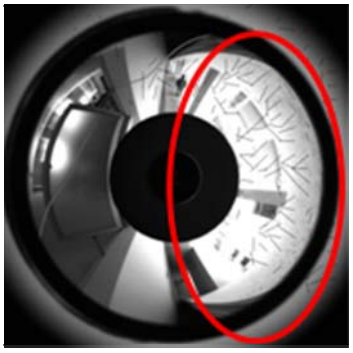

図 18. 誤認識を起こしている画像データ

iii)　移動方向取得処理評価実験結果

移動方向取得処理の評価実験では各カメラで動体を検知後，動体の移動方向を計算し，正しい結果が得られているかどうかを検証した．また動体がカメラ間を移動した時に，比較を行なっても正しい結果を得られるかどうかを検証した．

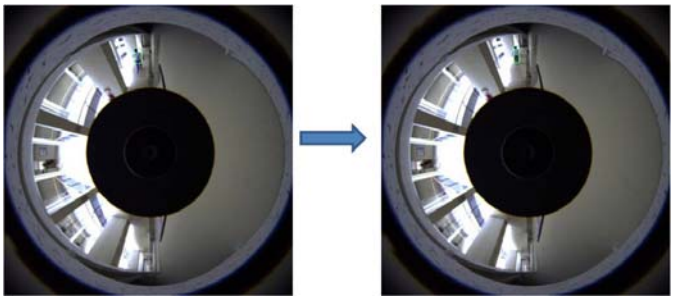

図 19. 移動方向取得処理

実験の結果，下記の通りに移動方向を取得することができた．

実験結果
TXG20c
1 Central = 3 MOVE_BOTTOM_RIGHT
2 Central = 2 MOVE_BOTTOM_LEFT

TXG50c
1 Central = 2 MOVE_BOTTOM_LEFT
2 Central = 2 MOVE_BOTTOM_LEFT
3 Central = 3 MOVE_BOTTOM_RIGHT

この結果から人物が一方の方向に動いていることが特徴値から取得することができた．またカメラ間が移動しても移動方向を比較した際に，同一方向の結果を得られることができるので，同一人物として認識することが可能である

ことがわかった．一方でカメラの設置位置によっては移動軸が大きく変化してしまい，誤認識を起こしてしまうことがわかった．例えば，全方位カメラの内角の死角で人物の移動方向が変わると，システム側はそれを知ることができず，違う動体として判断してしまう．そのためカメラの設置位置が重要であることがわかった．またフレームレートが確保できないと移動時の距離が大きくなってしまい，誤認識を起こしてしまうことがわかった．そのため高いフレームレートを維持し，軸の変化に柔軟に対応することで精度を向上させることができると考えている．

iv) 監視サーバー負荷測定結果

監視サーバーの負荷測定実験ではシステム動作時の認識処理時間の測定と CPU 使用率を測定した. 処理時間の測定には TXG50c を使用するプログラム内にて clock 関数を用いて測定を行った．また CPU 使用率の測定は 10 分間システムを動作させた時の使用率を測定した．図 20, 図 21 はそれぞれ処理時間と CPU 使用率をグラフ化したものである．

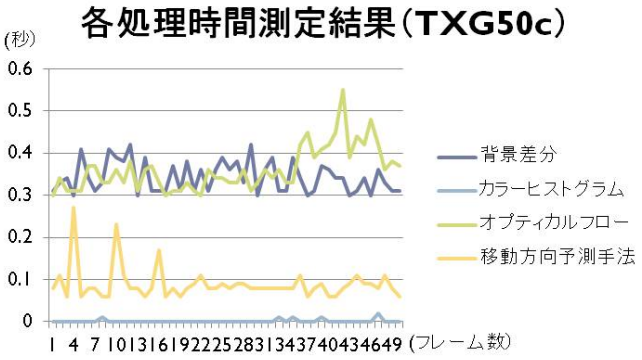

図 20. 各処理時間測定結果

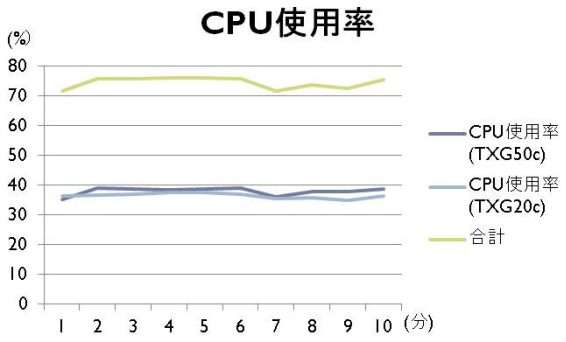

図 21. CPU 使用率測定結果

各処理時間がかかるほど，取得するフレームレートが落ち，CPU 使用率が高いほど監視サーバーに負荷がかかっていることがわかる．処理時間の測定では画像全体に処理を行う背景差分とオプティカルフローが最も大きいことがわかった．また CPU 使用率は各カメラ合計で約 74%程度の負荷がかかっていることがわかった．この結果から監視サーバーには高負荷がかかっており，カメラの台数を更に増やすためには負荷分散処理を行う必要があることがわかった．また処理時間が長くなると取得するフレームレートが落ちることから認識率にも影響が及ぶことがわかった．このことから負荷分散処理の検討が必要であることがわかった．

10. まとめと今後の課題

本研究では複数の全方位カメラを用いたシームレスな監視映像システムを提案し，有効性を検証した．提案した手法によって広範囲の監視が可能になり，設置位置等の物理的制限がなくシームレスな監視が可能となる．また評価実験において，カメラ間の認識，追跡処理を検証し，カメラ間の追跡処理が可能であることがわかった．

今後の課題として，環境に適応した認識処理を検討していく必要がある．今回，認識処理に使用した各手法は照明の変化が起こると結果が落ちる可能性があり，誤認識を起こす可能性がある．そこでその時点での環境に適した認識処理を模索し，認識率を高めていく必要がある．また監視サーバーの負荷が高い問題を解消するため負荷分散処理を行う必要があると考えている．

参考文献

[1] 佐藤洋介, 米田裕也, 橋本浩二, 柴田義孝, “ 全方位カメラと制御カメラを利用した遠隔カメラワークシステム ” 情報処理学 研究報告マルチメディア通信と分散処理（DPS 研究会）, No. 130, p429-434, 2007 年 3 月

[2] 大葛広和，佐藤洋介，米田裕也，橋本浩二，柴田義孝:“Gigabit Ethernet カメラを利用した超高精細全方位映像システム ”,情報処理学会第 71 回全国大会 pp283-284

[3] 米田裕也, 橋本浩二, 柴田義孝: 高解像度全方位映像の利用と通信のためのミドルウェアの開発, 情報処理学会第 68 回全国大会, pp581-582(2006)

[4] Ross Cutler, Yong Rui, Anoop Gupta, JJ Cadiz, “Distributed Meetings: A Meeting Capture and Broadcasting System”, ACM Multimedia ’02, pp.1-6, December 2002

[5] Mohamed F. Abdelkader, Rama Chellappa, Qinfen Zheng, LipChen Alex Chan, “Integrated Motion Detection and Tracking for Visual Surveillance.”, ICVS 2006: 28

[6] Syed Sohaib Ali, M. F. Zafar, Moeen Tayyab, “Moving human detection and recognition in videos using adaptive method and support vector machine”, FIT 2009: 37

[7] Welcome OpenCV wiki, http://opencv.willowgarage.com/wiki/

[8] Baumer- Passion for Sensors, http://www.baumer.com/en.htm