

動画中の移動物体追跡に関する研究

大槻 典行* 寺島 圭亮**

The Moving Object Tracking in the Motion Picture.

Noriyuki OHTSUKI Keisuke TERASHIMA

Abstract - This report proposes a method of object tracking in the motion picture. This method contains a new block-matching algorithm. This algorithm dose not require high-performance computers because this algorithm calculates the fact of being similar to few blocks. The proposed method was evaluated with real motion pictures and succeed to track a moving object in the motion picture.

Key words : object tracking, motion picture, moving object, block matching

1. はじめに

近年、パーソナルコンピュータのハードウェアおよびソフトウェアの性能向上には目覚ましいものがあり、中でもハードウェアの処理能力の向上が著しい。パーソナルコンピュータの処理能力の増大に伴い、従来では大型計算機やスーパーコンピュータでなければ処理が不可能であった画像処理や動画画像処理なども手軽に試行することが可能となった。しかし、動画画像から任意の物体の動きを実時間で抽出するような処理など、その対象によっては、まだ処理能力が不十分な場合がある。このような処理に対しては、現在でも処理能力が大きいコンピュータが利用され、手軽に実験等を試行する環境ではない。そこで、我々は、大型計算機や最新型の高処理能力を持つパーソナルコンピュータに比較して処理能力が高くないパーソナルコンピュータを用いて、動画中の移動物体の実時間での追跡を試みた。

本報告では、動画処理を行うには処理能力的に不十分と考えられるパーソナルコンピュータを用いて実時間で移動物体を追跡可能な手法を示すと共に、実験結果からその有効性を検討する。

2. 動き検出

移動する物体は動画画像中で時間毎にその位置が変化する。ここでは、動画画像において、物体の位置座標が時間毎に変化する状態を検出することを動き検出と呼ぶ。

コンピュータ内に取り込まれる動画画像は、図2-1に示すように、連続した複数枚の静止デジタル画像（以下フレームと呼ぶ）で構成される。このフレームの並びの順番は、時間の経過に対応している。動画画像中で移動する物体部分の画像は、隣り合う2つのフレーム中で位置座標のみが異なる部分画像として存在する。つまり、連続する二つのフレームにおいて座標の変化は、時間的な位置変化を示し

ている。したがって、この部分画像の座標変化を検出することが動き検出となる。この動き検出を利用することによって、動画中の物体の移動追跡が可能になる。以下に、動き検出の中の画像の動きの大きさと方向を求める動きベクトル検出の手法[1]について述べる。

2. 1 ブロックマッチング法

前述のように、基本的に動き検出は、連続する二つのフレームにおける部分画像の位置変化を求めることで実現する。動き検出法には、大きくわけてマッチング法と勾配法がある。マッチング法は、精度良く動きを検出できるが処理量が多い。一方、勾配法は処理量がマッチング法に比較して少なく動きを検出できるが、検出の精度が低い。以下にマッチング法による移動検出の原理を示す。

ここでは、マッチング法の中で代表的なブロックマッチング法の動き検出の処理手順を示す。まず、対象となる連続する二つのフレームの一つにおいて、動きを検出したい画像が含まれる小領域をブロックとして記憶する。次に、他方のフレームにおいて、先のブロックと同じ大きさの領域（比較領域のブロック）で画像の類似度を求める。更に、フレーム内の比較領域をずらして同様に類似度を求め

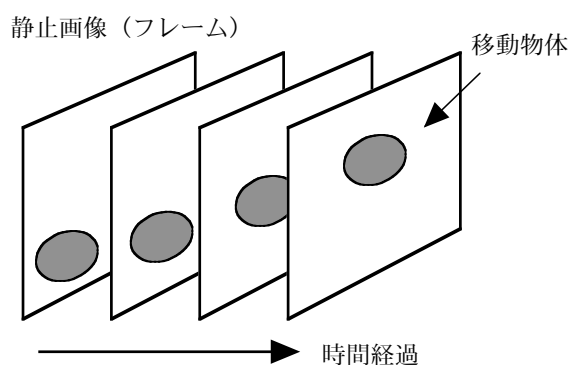


図2-1 複数の静止画像（フレーム）による動画画像の構成

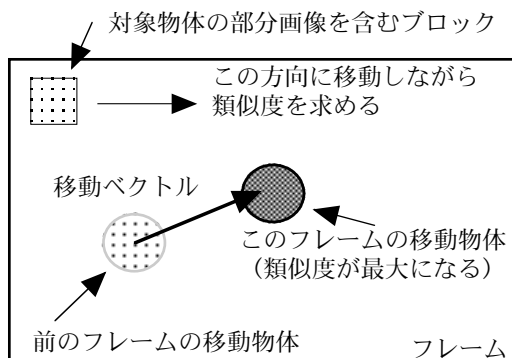


図2-2 動画像におけるブロックマッチング法による移動検出

る。これをフレームの全体の領域に対して行う。この中からもっとも類似度が大きい領域が、動き検出をしたい物体の画像が移動した領域である。この領域の座標と先に記憶したブロックの座標から移動距離と移動方向を求める。(図2-2)

2. 2 ブロックマッチング法の処理速度の改善

ブロックマッチング法は、フレーム全体の領域で対象ブロックと類似度を求めるため、処理量は莫大になる。そこで、処理速度の改善方法がいくつか提案されている[1]。ここでは、二つの手法を示す。一つ目は、全探索法と呼ばれ、移動検出したい物体部分の画像を中心として、その周辺の領域をマッチング領域にするものである。この手法の場合、フレーム全体に対して探索しないので処理量を減少することができる。しかし、探索処理間隔内に周辺領域を越えて、検出対象が移動する場合には、検出を失敗する場合がある。(図2-3)

二つ目は、ツリー探索法と呼ばれ、移動検出したい物体部分の画像を中心として、その周辺をブロックに分け、各ブロックで類似度を求める。この中から最も類似したブロックを選択し、さらにその周辺をブロックに分け、各々類似度を求め最も類似したブロックを選択するものである。求めたブロックの周辺を更にブロック化する際に、中心からの距離を徐々に小さくしていくことにより移動物体部分の画像を求めることができる。この手法は探索する領域が全領域ではなく、ブロック化する回数を制限することで処理量を減らすことができる。しかし、動きの大きい移動物体を検出するにはブロック化の領域間隔を大きく取る必要があり、しかも、ブロック化の回数を制限すると移動物体の部分画像と完全に一致したブロックが求められない場合が生じる。従って、検出精度が低下することになる。(図2-4)

以上のように、移動検出における処理量を減らし処理速度を高める手法が提案されているが、移動検出の精度が低下することは避けられない。検出精度を高くするには、全探索法のように画像全体の領域に対して類似度を求める必要があり、処理量が莫大に

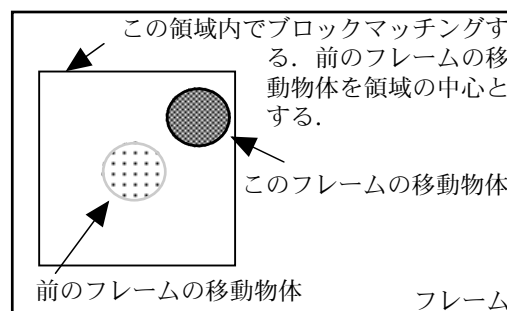


図2-3 周辺領域の全探索法

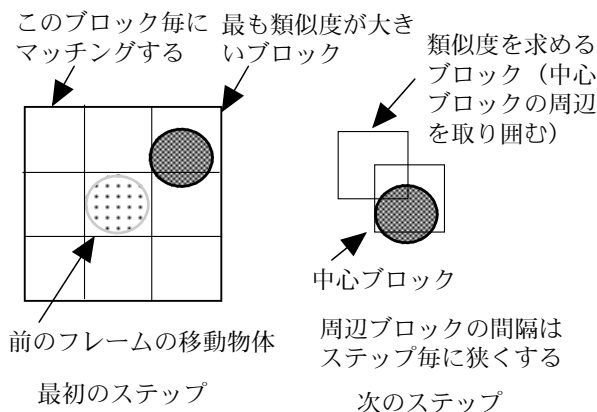


図2-4 ツリー探索法

なる。この処理を実時間でを行うには、高速処理が可能なコンピュータが必要になる。

3. 実時間移動体追跡

本節では、ブロックマッチング法を用いた動き検出手法を改善し、実時間で正確に移動物体の動きを検出する処理量が非常に少ない手法を示す。更に、この手法を用いた移動物体の追跡を行う。

3. 1 マッチング領域の確定

ブロックマッチング法において処理量を減少するには、マッチング領域を小さくすると良い。しかし、小さい領域にすると、動きの速い物体の場合、マッチング領域を逸脱してしまうため動き検出を失敗してしまう。そこで、本手法ではフレーム間差分を用いて、動的にマッチング領域を決定する。ここで決定されるマッチング領域の大きさは、動的で常に移動検出対象となる物体の画像を含んでいる。以下にその手法を示す。

まず、フレーム間差分を求める。フレーム間差分を求めると、図3-1の様に検出対象画像を含む領域とその両端の領域に差分値を持ち、背景部分が差分値0の結果が得られる。これは背景部分の画像は静止しているのでフレーム間の画像の濃度値に差は無く、差分値が0になるのである。しかし、移動する対象物部分の画像は、対象物と背景および対象物と対象物の異なる部分に非零の差分値が得られる。この非零の部分の領域に検出の対象物が存在するので

この領域を矩形領域に拡張してブロックマッチングを行えば良い。しかし、対象物が一様な面を持つ場合、フレーム間差分値は、図3-2の様に対象物の移動方向の前後に非零の領域ができるが、対象物の中心付近では対象物が一様なため差分値が零になる。このような場合は、対象物の前後の非零の差分値部分を全て含む矩形領域を求め、ブロックマッチングを行う。

ここで得られる領域の大きさが小さい場合は、対象物ではなくノイズなどが求められたと考え、次のフレームを使って再び領域の決定を行う。これにより、動画像中に小さなランダムノイズが含まれている場合でも、ノイズの影響を受けることなく移動検出が行える。

3.2 矩形領域内のブロックマッチング

3.1で確定したブロックマッチング領域において、ブロックマッチングを行い移動検出を行う。ブロックマッチングでは、あらかじめ移動検出の対象となる物体部分の画像を記憶しておく必要がある。本手法では、マッチング領域を確定する段階で得られる領域の中心部分の画像を記憶して利用している。この画像は十数フレーム毎に更新することで、対象物部分の画像の変化に追従させている。また、対象物部分の画像を更新しない期間（十数フレーム）を設けている。これは、対象物が短時間静止した場合やフレームから外れてしまった場合に、マッチング領域が決定できないためである。この期間内に再び、対象物が動き始めたり、フレーム内に入ってきた場合はこのブロックを用いて移動検出を行うことで、続けて対象物を追跡可能としている。

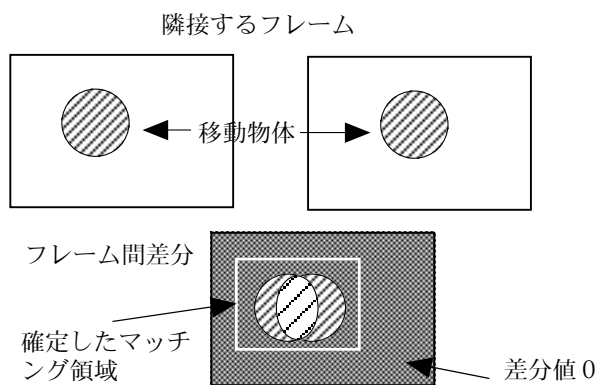


図3-1 フレーム間差分によるマッチング領域の決定

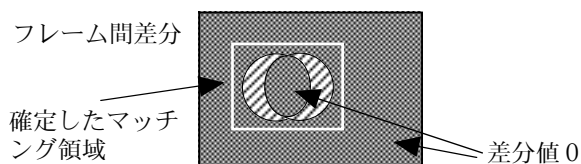


図3-2 差分値が0になる部分も含む矩形領域

3.3 移動物体追跡

本研究の目的である、動画像中の移動物体の追跡は、移動検出のアルゴリズムを基に移動検出を行い、移動ベクトルと物体の判定（認識）で実現する。本手法では、簡単のため平面上で物体の回転を伴わない移動を想定している。従って、移動検出時に移動物体の中心画像を記憶することで、物体の認識も含めて移動検出しているため、移動ベクトルを追って行くことで移動物体の追跡も行っている。

4. 移動物体追跡実験

以下に本報告で示した手法による移動物体追跡を行った結果を示す。

4.1 実験システム

実験に用いたシステムのスペックを以下に示す。

パーソナルコンピュータ

CPU: Celeron 400MHz, Memory: 128MB

Video: ATI RagePro 4MB

Captuer: Bt878 Chip 24bit Color

OS: VineLinux 2.1.5 with Video fo Linux

実験プログラム開発環境

gcc, GTK+, Video for Linux

現在主流のパーソナルコンピュータに比較すると、速度的にも遅い部類に入るものである。パーソナルコンピュータのインターフェーススロットにビデオキャプチャボードを挿入し、ビデオ入力端子にはビデオカメラを接続した。実験システムの構成図を図4-1に示す。

4.2 実験

ここでは、アルゴリズムに従って実験した結果を示す。なお、処理は全て24bitカラーで行った。

4.2.1 予備実験

マッチング領域の確定において、フレーム間差分を求めた結果を図4-2に示す。背景のように静止した部分の差分値が0になっているために、黒く表示され、移動物体がある部分は差分値が0とならず黒以外で表示されているのが判る。また、一様な面を持つ物体の場合は、物体の中心部分が背景と同じように黒色で表示される。

フレーム間差分値からマッチングのための矩形領域を確定した例を図4-3に示す。検出する移動物体

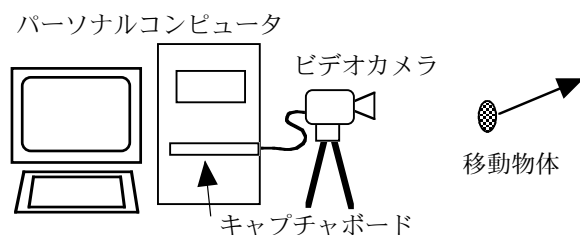


図4-1 実験システムの構成

を全て含んだ矩形領域になっている状態が判る。

4. 2. 2 移動体追跡実験

移動体を追跡した結果を示す。実験条件として、動画像のサイズは160×80画素および320×240画素について行った。追跡ブロックサイズは8×6画素である。画像のサイズによって1秒間当たりの追跡可能なフレーム数が異なる。160×120の画像サイズでは10fps (frame/sec) であるが、320×240のサイズについては2~3fpsまで低下した。追跡対象となる物体は、球形の一様な面を持つ形であり、回転の影響は無い。また、移動方向は平面的であり奥行き方向には移動しないものとする。図4-4に追跡結果を示す。物体を追跡した軌跡は、×印で元画像に書き込むようになっていた。直線的な方向では無くても追跡可能であることが示されている。また、複雑な形をした一様な面を持つ物体についても回転等の状態変化が無い場合は追跡可能であった。

4. 3 考察

予備実験におけるブロックマッチング領域の決定では、フレーム間差分値をそのまま用いた場合背景部分に差分値が0にならない部分が多く見られた。この値は非常に小さなものであるので誤差と考えられる。従って、しきい値を用いて、しきい値に満たない差分値を0としてフレーム間差分値とした。また、矩形領域決定にはフレーム間差分値の和を求め

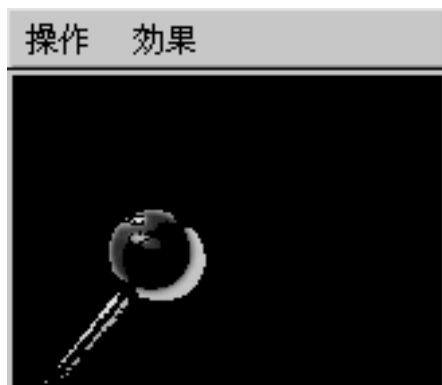


図4-2 フレーム間差分結果



図4-3 マッチング領域確定結果

て、もう一つのしきい値を超えた領域含む矩形領域を求めているが、この結果、差分値の和の小さな値が検出されずノイズ除去の効果が現れている。求められた矩形領域の大きさは、物体の移動に伴い変化している様子が見られ、特に、移動速度との関連が見て取ることができる。このことから領域検出のアルゴリズムが有効に働いていることが判る。

移動体追跡実験において、どの方向から見ても同一で一様な面を持つ物体の場合は十分な精度で追跡した。また、複雑な形をした物体については、追跡に失敗する頻度が多くなった。追跡を失敗する場合は、物体の回転や画像上の物体の一部の輝度が大きく他の部分と異なっている場合である。以上のように、複雑な物体でも精度良く移動追跡を行うには、物体の回転や物体の各部分の輝度を補正する処理が必要である。

5. まとめ

本報告では、十分な処理速度を持っていないと考えられるパーソナルコンピュータを用いて動画中の移動体追跡を実時間で行う手法を提案した。ブロックマッチング法を基本としているが、より高速なコンピュータを用いなければ実現できない処理を、フレーム間差分値を利用したマッチング領域の縮小化によって低速なコンピュータでも実現可能であることを示した。移動物体の形等の条件を限定した場合十分な精度で移動体追跡が可能であることを実験で確認した。移動物体に対する条件を取り除くことでより汎用性のある追跡手法にすることができると考えられる。

本研究では、移動体の移動方向を平面方向に限定しているが、今後、奥行き方向についても正確に追跡可能な手法にすることで、実用的な応用も可能である。

参考文献

[1]八木伸行, "デジタル映像処理", オーム社

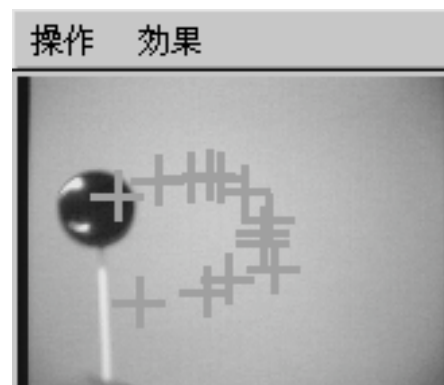


図4-4 移動物体追跡結果 (十字が物体の移動軌跡)