

# 低照度映像向け高速動体検知アルゴリズムのための GPU ベース実験システムの開発

三浦康之<sup>†</sup> 成重涼太<sup>†</sup> 根本智文<sup>†</sup> 藤島秀一郎<sup>†</sup> 周瑋傑<sup>†</sup> 杉岡純平<sup>†</sup>

侵入者や不審者の監視・記録を行う監視カメラは、施設内はもとより、近年は市街地などに設置されて監視業務を行っている。また、近年のビデオカメラの価格低下により、個人でも監視カメラを所有し自宅などに設置することも可能になっている。そのような監視カメラの中には、動体検知機能を備えたものも存在する。我々は、さまざまな動体認識アルゴリズムのうち、時空間特徴量に着目し、実時間で大量の映像の識別が可能となる高速な動体認識のためのアルゴリズムの検討を行っている。これまでに、探索時間を大きく短縮するアルゴリズムの提案や、通常のビデオカメラで撮影された 1lux 以下の低照度映像を対象とした動体検知の研究を行ってきた。

これまで、DSP 評価ボードを使用して実験を行っていたが、装置の低価格化や高性能化を目指して GPGPU を用いた実験システムの開発を進めている。本稿では、現在研究中の動体検知アルゴリズムの紹介を行うとともに、GPGPU を用いた実験システムの開発状況について報告する。

## The Development of the GPU based Experiment System for the High-Speed Moving-Object Detection for Low Illumination Video Image

YASUYUKI MIURA<sup>†</sup> RYOTA NARISHIGE<sup>†</sup> TOMOFUMI NEMOTO<sup>†</sup>  
SHUICHIRO FUJISHIMA<sup>†</sup> WEIJIE ZHOU<sup>†</sup> JUNPEI SUGIOKA<sup>†</sup>

A security camera is installed in facilities and a city, and they are watching intruders and strangers. In recent days, it is possible that individual person possesses a security camera and installs in a home by price decline of video camera. Several types of such cameras have the mechanism of moving-object-detection which can recognize moving objects. Among the various moving-object-detection algorithms, we focused "spatio-temporal motion features". We have researched the high-speed moving-object-detection algorithm which can discern the large amount of video images in real-time environment. Until now, we have researched the algorithm with low processing time, and researched the method of image correction for moving-object detection in the low illumination video image of "below 1 lux" obtained by normal type video camera.

Currently, we have researched by using DSP Evaluation Board. Now, we are developing the evaluation system by GPGPU for cost reduction and performance improvement. In this paper, we introduce the moving-object detection algorithm researched by us, and we report the status of development of evaluation system by GPGPU.

### 1. はじめに

侵入者や不審者の監視・記録を行う監視カメラは、施設内だけではなく市街地などに設置されて監視業務を行っている。近年、カメラの価格低下により個人でも監視カメラを所有し自宅などに設置することも可能になっている。そのような監視カメラの中には、動いている物体を認識する動体検知機能を備えたものも存在する。動体検知の機能により、監視作業を円滑に実施することが可能となる。

映像の中から、人物などの動く物体を認識する方法としては、以下のようなものが知られている。

- ・ フレーム間差分に基づく手法<sup>1)2)</sup>

動画像の前後のフレームとの差分を求め、閾値を上回った場合に動体と認識する単純な手法である<sup>1)</sup>。一般向けに提供されているソフトウェアの中には、フレーム間差分とメディアンフィルタの組み合わせをベースとした手法により、ほぼ正確に動体を認識するものが存在す

る<sup>2)</sup>。この手法は、アルゴリズムが単純で、実際に動いている物体をほぼ間違いなく認識することができるが、光の加減などにより動体以外を認識する場合がある他、樹木やの水面のゆらぎの類を誤認識する欠点がある。また、対象物が静止した場合、認識することができないといった欠点がある。

- ・ 背景差分法に基づく手法<sup>3)4)5)6)</sup>

予め用意された背景画像と対象画像を比較して、動体を認識する手法。背景画像を用意するために、背景のみの画像を予め撮影する他に、過去の複数のフレームの中から、統計的な手法を用いて背景部分のみを抽出した画像を用意するなどの手法がある。このような手法は、背景画像が正確であれば高い精度で動体を認識することが可能となるが、照度の変化など環境の変化に弱いという弱点がある。

- ・ 画像の色に着目した認識法<sup>7)</sup>

<sup>†</sup> 湘南工科大学  
Shonan Institute of Technology

画像の色情報に着目し、特定の色の領域を動体と見做す、あるいは同じ色の物体を追跡する手法。背景差分同様、環境の変化に弱いという弱点があるが、照度の変化を推定して、色情報を同定する手法などが提案されている<sup>7)</sup>。この手法は、しばしば他の手法と組み合わせて使用されることがある<sup>8)</sup>。

特徴量を用いる方法<sup>9)12)13)</sup>

特徴量を用いる手法としては、HOG<sup>10)</sup>や Haar-Like<sup>11)</sup>をベースとした手法がよく知られている。うち、HOG は人物領域の識別に、Haar-Like は顔画像の認識に適している。これらは、映像中の人物の認識や追跡に広く用いられている。これら特徴量の多くは、特定の性質を持った対象物を認識するためにしばしば用いられる。

また、オプティカル・フローの抽出と類似したアルゴリズムに基づき、時空間方向の画素情報に基づいた複数の時空間特徴量を求めて判別を行う手法が存在する<sup>12)</sup><sup>13)</sup>。この手法は、動体と背景の揺らぎ（ノイズや水面のまたたき、植物などの背景の揺れ等）を弁別することに適している。

符号化画像の復号に伴う符号化情報を用いる手法<sup>8)14)15)</sup>

MPEG や H.264-AVC などの符号に含まれる情報を利用するもので、符号中の直流成分や動きベクトルなどが使用される。このような手法は、追加的な計算量を要することなく、精度の高い認識が可能となるため、高速処理に適しているが、復号に伴う特徴量の抽出には動画像符号化に関する高度な知識が必要となる。

上記のうちいくつかの手法は、しばしば組み合わせて用いられる。また、ブースティング<sup>16)17)</sup>などの学習アルゴリズムを組み合わせることにより、認識の精度を高めることが可能となる。

我々は、上記のうち、対象物に依存しにくく、かつ非動体との判別精度の高い時空間特徴量に着目し、実時間で大量の映像の識別が可能となる高速な動体認識のためのアルゴリズムの検討を行っている。これまでに、以下のようなことを明らかにした。

- 過去に提案された手法に基づいて、枝切りを行うことにより、探索時間を大きく短縮することが可能となった<sup>18)</sup>。
- 輝度補正やノイズ除去フィルタと組み合わせることによって、極端な低照度環境下における動体検知を可能とする枠組みを提案した<sup>19)</sup>。

上記のアルゴリズムに関して、実映像を用いた実験を行うため、DSP 評価ボードを用いた映像処理装置の開発を行った。また現在、OpenCV を用いて GPGPU による映像処理を行う実験装置の開発を進めている。

本稿では、現在研究中の動体検知アルゴリズムの紹介を行うとともに、GPGPU を用いた実験システムの開発状況に

ついて報告する。

## 2. 時空間特徴量に基づく動体認識

### 2.1 提案手法

時空間特徴量に基づく動体認識は、「F1:空間平均した動きの強さ」「F2:時間平均した動きの強さ」「F3:時間的な動きの一様性」の三種類の特徴量に基づき、動体と背景のゆらぎを識別するものである<sup>12)13)</sup>。これら三つの特徴量のうち、F1 は、長期的な時間的情報を必要とせず、二枚のフレームの関係のみにより計算可能な特徴量であることから、比較的高速に処理が可能である。一方、木の葉の揺れや水面のゆらぎ、自動車のヘッドライトや環境光の変化など、相当程度の背景の揺らぎの弁別が可能である。そこで我々は、F1 に着目し、認識の精度を大きく損なうことなく動体と背景の揺らぎを弁別する方法を用いた。

我々の手法では、フレームを  $N \times N$  ピクセルのブロックに分割し、ブロック毎に動体領域か否かを判定することにより、計算量を大幅に削減する。我々の手法では、以下の手順により動体の判別を行う。

(1) 着目フレームと直前のフレームのフレーム間差分により、動体候補となる画素を抽出する。

着目フレーム内の  $(x, y)$  における画素値を、 $I(x, y)$ 、直前のフレームの  $(x, y)$  における画素値を、 $I_p(x, y)$  とする。全画素に対して、フレーム間差分値

$$I_d(x, y) = |I(x, y) - I_p(x, y)| \quad (1)$$

を求め、以下のような閾値処理により、フレーム間差分後の画像を二値化する。

$$I_{th}(x, y) = \begin{cases} 1 & (I_d(x, y) \geq th1) \\ 0 & (I_d(x, y) < th1) \end{cases} \quad (2)$$

ここで、 $th1$  は二値化のための閾値である。

(2) ブロック内の全画素の多数決により、当該ブロックが動体候補か否かを判別する。

はじめに、 $(x, y)$  から  $(x + N - 1, y + N - 1)$  までの  $N \times N$  ピクセルのブロック内において、以下の値  $s$  を計算する

$$s = \sum_{i=x}^{x+N-1} \sum_{j=y}^{y+N-1} I_{th}(i, j) \quad (3)$$

上記の  $s$  が  $N^2/2$  以上となった場合、当該ブロックを動体候補ブロックとする。ここまで処理は、背景差分画像に対するメディアンフィルタ、および二値化の処理を施すことと等価であるため、対象フレームのノイズの影響が除去される。また、(3)式の処理はメディアンフィルタに比べて計算量が大幅に少ないため、高速な処理が可能となる。

(3) 動体候補ブロックに対する判別を行う。

(2)において動体候補となったブロックに対して、前述の F1 に類似した特徴量を用いて、動体が背景の揺らぎ

かの判別を行う。当該ブロックにおいて、ブロックマッチング法によりオプティカル・フローを求める。求められたオプティカル・フローを $(v_x, v_y)$ とする。当該ブロックにおけるマッチング誤差 $err(u, v)$ を、

$$err(u, v) = \sum_{i=x}^{x+N-1} \sum_{j=y}^{y+N-1} |I_p(i, j) - I(i+u, j+v)| \quad (4)$$

としたとき、 $err_0 = err(0,0) - err(v_x, v_y)$ が $th2$ 以上となった場合、当該ブロックの画素は特定方向に水平移動しているものと見做されるため、動体と判定される。 $th2$ を下回った場合、当該ブロックは背景の揺らぎを有するものと判定される。

図1に、本稿の手法による動体識別の結果を示す。図1の赤い四角が動体領域のブロック、白い四角が背景の揺らぎとなる。



図1 本稿の手法による動体識別の結果  
Figure 1 The Result of Moving Object Recognition

## 2.2 計算時間の削減

前述の手法は、条件によってはさらなる計算量の削減が可能となる。具体的には、以下の二つのアプローチにより計算量を削減できる。

### 提案手法1：ブロックマッチング法の計算量削減

2.1 節(3)におけるブロックマッチングにおいて、計算途中で求められる誤差の値と $err(0,0)$ を比較し、比較結果が $th2$ 以上となった場合、当該ブロックは動体と判断することができる。そのため、その時点で計算を中断して、次のブロックの探索に移行することができる。また、誤差の計算途中において誤差の値が一定以上となることが判明して、 $err(0,0)$ との比較値が $th2$ 以下となることが明らかになった場合、誤差の計算そのものを中断しても判定結果に影響がないことが分かる。この場合、正確なオプティカル・フローを求めることができないが、動体と背景の揺らぎの弁別には影響がない。

### 提案手法2：ブロックマッチング法の計算量削減

動体と判定されたブロックが一定数（本稿の実験では5ブロックとした）を超えた場合、当該フレーム内に動体があるものみなして、処理を中断する。この場合、動体の正確な位置を特定することができないが、フレーム

内における動体の有無を判定することは可能である。

640×480ピクセルの動画像に対して、二つの提案手法の実行時間を測定した。実験に使用した画像を図2に、実験結果を表1に示す。表1に示すように、二つの提案手法によって大幅に計算量を削減することが可能となる。



図2 実験映像

Figure 2 Experimental Video Images

表1 提案手法による実行時間

Table 1 Processing Time of Proposed Method

画像	結果	実行時間(秒)	
		従来法	提案手法
樹木 A	背景	80.176	2.203
樹木 B	背景	47.131	1.765
通行人	動体	35.427	0.348
人混み	動体	48.229	0.047

## 3. 低照度映像を対象としたフレーム間差分処理

個人が所有しているカメラは、性能によっては夜間を撮影するのに適していないものがある。そのようなものでは、対象を捉えることが困難な場合があるため、動体検知は非常に困難になる。そこで我々は、通常のビデオカメラで撮影された1lux以下の低照度映像を対象とし、フレーム間差分法による動体検知のための画像補正に関する検討を行う。夜間に撮影された映像などの低照度映像に対する動体検知を行う際、前処理として輝度補正とノイズ除去を行うことによって認識の精度を高めることが可能となり、照度の変化に対して頑強な動体検知を行うことができる。そこで、ガンマ補正とノイズ除去フィルタを組み合わせた前処理において、適切なパラメータを設定する方法を検討する。特に今回は、ガンマ補正のためのガンマ値について、検知の

精度を高めるための設定法を提案し、性能を評価する。

### 3.1 提案手法

動体検知に関する研究として、背景差分に基づく方法が多く提案されているものの、実時間性を要するアプリケーションの多くはフレーム間差分に基づく方法が用いられている。本稿では、照度の変化に頑強な後者の方法を仮定する。

本手法は、フレーム間差分の前段階の処理となる。従って、2.1節における(1)のフレーム間差分に代わって、下記の処理を行うことになる。

- 1) 対象となる低照度画像をガンマ補正により補正し、輝度を上げる。
- 2) ガンマ補正後の画像に対してフィルタリングを行い、ノイズを除去する。
- 3) フレーム間差分処理を行う。
- 4) 閾値処理により動体候補を抽出する。

以下、各段階における処理の詳細を記述する。

#### 3.1.1 ガンマ補正

ガンマ補正は、一般には入出力機器の特性に応じて画像の明るさを調整するための手段として用いられているが、低照度画像の補正に用いることも可能である。補正前の輝度値を  $x$ 、補正後の輝度値を  $y$  とすると、ガンマ補正の式は下式に示される通りになる。

$$y = x^{\gamma_r} \quad (5)$$

ただし、 $\gamma_r = 1/\gamma$ であり、 $\gamma$ はガンマ補正のガンマ値である。

動体部分の輝度の中央値を  $x_1$ 、非動体部分の輝度の中央値を  $x_2$  とする。また、ガンマ補正後のそれぞれの輝度値を、それぞれ  $y_1 = x_1^{\gamma_r}$ 、 $y_2 = x_2^{\gamma_r}$  とする。本稿の手法では、それぞれの中央値の差が最大になるようにガンマ値を設定する。したがって、

$$f(\gamma_r) = y_1 - y_2 = x_1^{\gamma_r} - x_2^{\gamma_r} \quad (6)$$

としたとき、

$$f'(\gamma_r) = \frac{df(\gamma_r)}{d\gamma_r} = 0 \quad (7)$$

となるような  $\gamma_r$  を求めれば良い。

$$f'(\gamma_r) = \ln x_1 \cdot x_1^{\gamma_r} - \ln x_2 \cdot x_2^{\gamma_r} = 0 \quad (8)$$

より

$$\ln x_1 \cdot x_1^{\gamma_r} = \ln x_2 \cdot x_2^{\gamma_r} \quad (9)$$

$$\frac{\ln x_1}{\ln x_2} = \frac{x_2^{\gamma_r}}{x_1^{\gamma_r}} = \left(\frac{x_2}{x_1}\right)^{\gamma_r} \quad (10)$$

となる。よって求める  $\gamma_r$  は、

$$\begin{aligned} \gamma_{r\text{MAX}} &= \log_{x_2/x_1} \frac{\ln x_1}{\ln x_2} \\ &= \frac{\ln(-\ln x_1) - \ln(-\ln x_2)}{\ln x_2 - \ln x_1} \end{aligned} \quad (11)$$

となる。

対象画像の動体と非動体の領域が明確でない場合、両者

の輝度値の中央値が同一であると見做して  $\gamma_r$  を求めることも可能である。その場合、

$$\begin{aligned} \lim_{x_1 \rightarrow x_2} \gamma_{r\text{MAX}} &= \lim_{x \rightarrow x_2} \frac{(\ln(\ln x) - \ln(\ln x_2))'}{(\ln x_2 - \ln x)'} \\ &= \lim_{x \rightarrow x_2} \frac{\frac{1}{\ln x} \cdot \frac{1}{x}}{-\frac{1}{x}} = \lim_{x \rightarrow x_2} \left(-\frac{1}{\ln x}\right) \\ &= -\frac{1}{\ln x_2} \end{aligned} \quad (12)$$

として、 $\gamma_{r\text{MAX}}$  を求める。

#### 3.1.2 ノイズ除去

ノイズ除去のためのフィルタとしては、メディアンフィルタ、ガウシアンフィルタ、移動平均フィルタなどが考えられるが、メディアンフィルタでは 1lux 以下の低照度画像に対するガンマ補正に伴って強調されるノイズには対応できないことや、移動平均フィルタに比べて周波数特性の見通しが立てやすいことから、今回はガウシアンフィルタを用いてノイズの除去を行った。

今回の手法では、 $3 \times 3$  ないし  $5 \times 5$  のガウシアンフィルタを繰り返し通すことにより、ノイズの強度を低減する。繰り返しの回数を目安として、動体付近における 2 画素分を周期とする (1 画素単位で発生する) 細かいノイズの強度を、ガンマ補正前の水準に戻すことを考える。当該周期におけるガウシアンフィルタの周波数特性  $F_2$  は、 $3 \times 3$  のフィルタにおいて約  $F_2 = 0.456$ 、 $5 \times 5$  のフィルタにおいて約  $F_2 = 0.253$  となる。そこで、(11)式により  $\gamma_{r\text{MAX}}$  を求めた場合は  $a = y_1/x_1 = x_1^{\gamma_r - 1}$  より、(12)式により求めた場合には  $a = y_2/x_2 = x_2^{\gamma_r - 1}$  より画素値の増幅率  $a$  を求め、 $F_2^n$  と  $1/a$  が最も接近する整数  $n$  を、フィルタリングの繰り返し回数としてフィルタリングを実行する。

#### 3.1.3 動体候補の決定

最初の 2 枚のフレームにおけるフレーム間差分においては、(12)式に基づいて  $\gamma_{r\text{MAX}}$  を決定し、1)~4)の処理に基づいて動体候補を決定する。このときに決定した動体候補領域に基づき、以降のフレームの差分処理を行う。動体候補領域と非動体候補領域における画素の中央値をそれぞれ  $x_1$ 、 $x_2$  とし、(11)式に基づいて  $\gamma_{r\text{MAX}}$  を決定し、1)~4)の処理を行う。以降同様の処理を繰り返す。

### 3.2 実験結果

3 節の方法に基づいて、2 枚の静止画像を用いて動体候補の検出を行った。実験に用いる画像を図 3 に示す。図 3(a)(b)は、照明のある状態の画像で、図 3(b)左中央にある物体を動体候補として検出する。図 3(c)は、図 3(b)と同じ対象を、証明のない状態で撮影したものである。本実験では、図 3(c)のような画像を対象に実験を行う。

図 3(c)の画像に補正を行った結果を図 4 に示す。図 4(a)は、ガンマ補正後の結果、図 4(b)は、ガンマ補正とノイズ除去を行った結果である。今回のケースでは、(12)式によ

って求められる  $\gamma_{r\text{MAX}}$  は、 $1/4.87 \approx 0.205$  となる。したがって、 $a = 47.23$  (倍) となり、 $5 \times 5$  のフィルタの場合  $F^n$  と  $1/a$  が最も接近する整数  $n$  は、 $n = 3$  となる。図 4(a) から分かるように、ガンマ補正後の画像はノイズが激しいため、フレーム間差分に適さないため、図 4(b) のように、ノイズ除去フィルタを通すことにより、フレーム間差分が可能となる。実験の結果、閾値が  $25/256$  付近のフレーム間差分において、動体候補の検知が可能であると判明した。

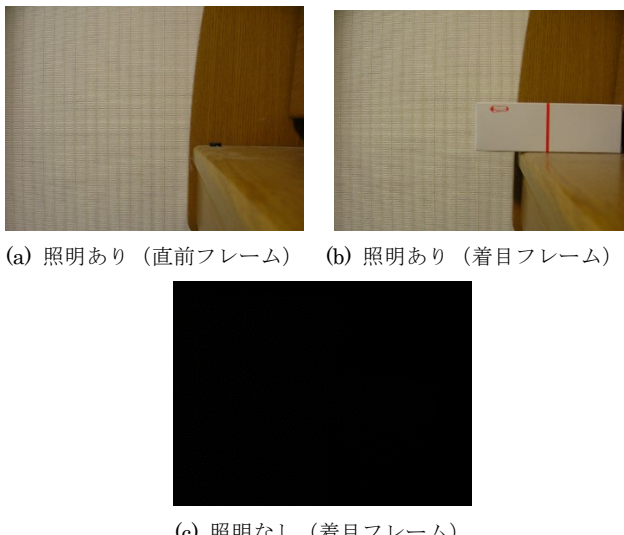


図 3 実験に用いる画像  
Figure 3 Experimental Images

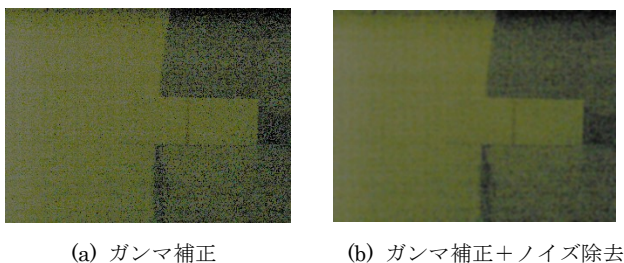


図 4 処理後の画像  
Figure 4 Images after Processing

### 3.3 提案手法の改善

ガウシアンフィルタの代わりに移動平均フィルタを用いる方法を検討している。移動平均フィルタは、フィルタの構造が単純なため、ガウシアンフィルタに比べてアルゴリズムの改善の余地が大きい。

今回は、全画素に対するフィルタリング処理を行う代わりに、縦方向のピクセルの平均値を求めて結果を保存し、それらを横方向に平均するという方法を用いた。 $5 \times 5$  の移動平均フィルタに対する双方のアルゴリズムを図 5 に示す。図のアルゴリズム `moving_average()` は、フレーム全体の移動平均フィルタを求めるアルゴリズムである。図 5(b) 中の `sum_v(&I[x][y-2], &I[x][y+2])` は、画素  $(x, y - 2)$  から画素

$(x, y + 2)$  の縦に並んだ 5 画素の合計を求める関数、`sum_x(&Imf_y[x-2][y], &Imf_y[x+2][y])` は、配列変数の要素 `Imf_y[x-2][y]` から `Imf_y[x+2][y]` の 5 つの要素の合計を求める関数である。図 5(a) の通常実装が、すべての画素において  $5 \times 5$  の移動平均フィルタを求めているのに対して、図 5(b) は、上半分のループでにおいて縦方向の合計を求めた後に、下半分のループにおいて、それらの合計を求めている。 $640 \times 480$  ピクセルの画像に対して簡単な実験を行って 500 回のフィルタリングの平均を求めた結果、図 5(a) と図 5(b) のアルゴリズムの間で、平均で 2.51 倍の実行速度の短縮が実現した。

```

moving_average(){
    for(y=0; y<HEIGHT; y++){
        for(x=0; x<WIDTH; x++){
            Imf[x][y] = 0;
            for(i=x-2; i>=x+2; i++){
                for(j=y-2; j>=y+2; j++){
                    Imf[x][y] += I[i][j];
                }
            }
            Imf[x][y] /= 25;
        }
    }
}

```

(a) 通常の実装

```

moving_average(){
    for(y=0; y<HEIGHT; y++){
        for(x=0; x<WIDTH; x++){
            Imf_v[x][y] = sum_v(&I[x][y-2], &I[x][y+2]);
        }
    }
    for(y=0; y<HEIGHT; y++){
        for(x=0; x<WIDTH; x++){
            Imf[x][y]
                = sum_x(&Imf_y[x-2][y], &Imf_y[x+2][y]);
            Imf[x][y] /= 25;
        }
    }
}

```

(b) 高速化された実装

図 5  $5 \times 5$  の移動平均フィルタ  
Figure 5  $5 \times 5$  Moving-Average Filter

## 4. DSP 評価ボードによる実験装置の構築

2 節および 3 節のような実験を行うため、動画像を実時間で処理できる装置の構築を行った。構築したシステムの模式図を図 6 に示す。動画像処理には、DSP の一種である DM365 の評価ボードを使用し、コンポーネント端子の入出力により動画の取り込みおよび出力を行っている。評価ボードは、Montavista Linux で動作する。評価ボードへのコンソール入出力は、RS-232C ポートを経由して PC により行う。

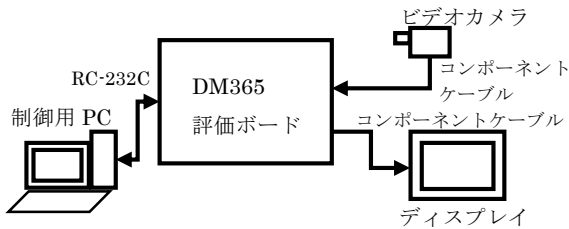


図 6 DM365 評価ボードによる実験装置  
Figure 6 Experimental System by using DM365

上記の装置に対して、2.1 節および 3.1.1 節の処理を実装した。その結果、720×480 ピクセルの動画に対して 16×16 ピクセルのブロック単位による、約 30 frames/s の実時間処理が可能であることが確認された。しかしながら、ノイズ除去フィルタを複数回実行する 3.1.2 節のような処理は、実現が困難であることが問題となっていた。

## 5. GPGPU による実験装置の実現

現在、GPGPU を用いた処理を行う実験システムの開発を進めている。動画の取り込み、および GPU の制御を、OpenCV を介して行い、USB カメラからの映像をリアルタイムに処理する。処理結果は、PC 内部に蓄積される一方、ディスプレイに表示される。

表 2 に開発環境を示す。OS は Windows 8.1 を使用する。今回使用した GPU は、NVIDIA の Quadro K620 である。Quadro K620 は、384 コアを有し、2GB の DDR3 SDRAM を搭載している。メモリインターフェイスは 128bit であり、29GB/s のメモリバンド幅を有している。OpenCV のバージョンは、OpenCV 3.0 であり、CUDA Toolkit 7.0 を使用している。

表 2 開発環境

Table 2 Development Environment

OS	Windows 8.1
CPU	Intel Core i7-5820K
GPU	NVIDIA Quadro K620
開発ツール	Visual Studio 2010
OpenCV	OpenCV 3.0
CUDA	CUDA Toolkit 7.0

上記のような装置に対して、まずは簡単な行列計算を行い、CPU における実行時間と GPU における実行時間の比較を行った。比較に用いた行列計算は、入力行列の各要素に対して、各々を指数とする自然対数の底  $e$  のべき乗となる行列を求める計算で、`cv::exp()`メソッドにより処理が行われる計算である。比較の結果を表 3 に示す。表に示すように、CPU と GPU の間に、最大で約 70 倍の速度差が生じることが明らかになった。

表 3 行列計算の実行時間

Table 3 Execution Time of Matrix Computation

行列のサイズ	実行時間 (秒)		CPU/GPU
	CPU	GPU	
128×128	0.031	0.016	1.94
512×512	0.54	0.016	34.19
2048×2048	8.94	0.14	63.85
8192×8192	142.62	2.06	69.13

図 7 に、システムの画像処理の流れを示す。

①においてフレームの取り込みを行って②においてグレイスケール化し、③の `cudaMemcpy` 関数によりフレームの画素をデバイスメモリにコピーする。④のカーネル関数 `image_proc` において動体認識に必要な任意の処理を実行する。その後、⑤においてホストメモリに画素をコピーする。上記を、フレームごとに繰り返す。

```
main(){
    cv::Mat src0;        //入力フレーム
    cv::Mat src;        //グレイスケール化後のフレーム
    cv::Mat dst;        // GPU 処理後のフレーム

    //cuda のためのメモリ割り当て
    cudaMalloc((void**)&gpu_srcdata, cols*rows*3);
    cudaMalloc((void**)&gpu_dstdata, cols*rows*3);

    while(1){
        // ①フレームの取り込み
        capture >> src0;

        // ②グレイスケールにする
        cvtColor(src0,src,cv::COLOR_RGB2GRAY);

        // GPU での処理
        cudaMemcpy(gpu_srcdata, src.data, cols*rows*3,
            cudaMemcpyHostToDevice); //③ホスト→デバイス
        .....
        // ④GPU 処理の起動
        d_image_proc <<<<grid, block >>>>(gpu_srcdata, gpu_dstdata);
        cudaMemcpy(dst.data, gpu_dstdata, cols*src*3,
            cudaMemcpyDeviceToHost); //⑤デバイス→ホスト

    }

    cudaFree(gpu_srcdata);
    cudaFree(gpu_dstdata);
}
```

図 7 画像処理の流れ

Figure 7 The Data Flow of Image Processing

## 6. まとめ

本稿では、現在研究中の動体検知アルゴリズムについて、時空間特徴量に基づいて高速処理を可能にした手法、およびガンマ補正とノイズ除去フィルタの組み合わせにより低照度映像に対する動体認識を可能にした方法を提案し、検討を行った。これらの結果から、以下のことが明らかになった。

- 時空間特徴量に基づいて高速処理を可能にした手法を実装した結果、大幅な高速化を実現し、実時間で処理可能となった。
- ガンマ補正とガウシアンフィルタの組み合わせにより、1lux 以下の環境における低照度映像において動体候補の検知が可能であることが明らかになった。

また、これらの手法に関する検証を行うため、GPGPUを用いた映像処理システムの開発を進めている。開発中のシステムは、OpenCV 3.0 により動画の取り込み、およびGPUの制御を行うもので、CPUによる処理に比べて大幅な性能向上が期待される。今後は、時空間特徴量に基づいた手法の実装、特に低照度映像に対応したガンマ補正とノイズ除去フィルタの実装を進める。また将来的には、ネットワークを介した監視カメラ網に対する侵入者認識を可能とするため、動画の圧縮に伴う認識精度に対する影響を評価する予定としている。

## 参考文献

- 1) 絹川亨, 石村理知, 簡便でノイズ耐性の高いフレーム間差分による動体検知法 -抽出画素の時空間でのつながり情報を用いたランダム変動背景の除去-, 電子情報通信学会技術研究報告(CS), 107(379), pp.147-152, 2007.12.
- 2) 動体検知ソフト Msako : <http://msako.jp/>
- 3) R. Yumiba, et.al., Moving Object Detection with Background Model based on Spatio-Temporal Texture, 2011 IEEE Workshop on Applications of Computer Vision (WACV), pp.352 - 359, 2011.
- 4) C. Stauffer and W.E.L. Grimson, Adaptive Background Mixture Models for Real-time Tracking, Proc. of 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), Vol.II, pp.246-252, 1999.
- 5) Desa, S.M, and Salih, Q.A., Image subtraction for real time moving object extraction, Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV'04), pp. 41-45, 2004.
- 6) 篠崎真太郎, 中島克人, 実時間物体追跡に適した動的背景推定法と背景差分法, 知能と情報 (日本知能情報ファジー学会誌), Vol.24, No.2, pp.637-647, 2012.
- 7) 林裕司, 藤吉弘亘, 岩堀祐之, 照度変化を考慮した Mean-Shift によるカラートラッキング, 電子情報通信学会論文誌 D, Vol.J90-D, No.8, pp.2039-2048, 2007.
- 8) 動き補償と色情報を組み合わせた MPEG 映像からの人物領域抽出, 情報処理学会研究報告, AVM-36, pp.31-36, 2002.
- 9) 三井相和, 山内悠嗣, 藤吉弘亘, Joint 特徴量を用いた 2 段階 Boosting による物体検出, 電子情報通信学会論文誌, Vol.J92-D, No.9, pp. 1591-1601, 2009.
- 10) Dalal, N, Triggs, B, Histograms of Oriented Gradients for Human Detection, IEEE CVPR, pp.886-893, 2005.
- 11) Viola, P and Jones, M, Rapid object detection using a boosted

cascade of simple features, CVPR, Vol.1, pp.511-518, 2001.

- 12) T.Haga, K.Sumi, and Y.Yagi, Human detection in outdoor scene using spatio-temporal motion analysis, Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Vol.4, pp.331-334, 2004.
- 13) 羽下哲司, 八木康史, 時空間動き特徴に着目した屋外侵入者監視技術に関する研究, 情報処理学会研究報告, CVIM-154, pp.259-274, 2006.
- 14) Changick Kim and Jenq-Neng Hwang, Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No.2, pp.122-128, 2002.
- 15) Dizan Vasquez and Thierry Fraichard, A Novel Self Organizing Network to Perform Fast Moving Object Extraction from Video Streams, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.4857 - 4862, 2006.
- 16) Y. Freund and R. E. Schapire, A decisiontheoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, No.1, Vol.55, pp. 119-139, 1997.
- 17) R. E. Schapire, Y. Singer, Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, No.37, pp.297-336, 1999.
- 18) 邢斌, 三浦康之, 渡辺重佳, 実時間性を考慮した監視カメラ映像のソフトウェアベースによる動き検出, 第 74 回情報処理学会全国大会, 1R-1, 2012.
- 19) Yasuyuki Miura, Yuta Fujii, The Examination of the Image Correction of the Moving-Object Detection for Low Illumination Video Image, IEEE International Conference on Consumer Electronics - Taiwan (IEEE 2015 ICCE-TW), pp.33-34, 2015.