

ABMとMSCによる実時間移動物体検出の検討

山本 文香[†], 岩井 儀雄[†], 谷内田 正彦[†]

[†]大阪大学大学院基礎工学研究科システム創成専攻

〒560-8531 大阪府豊中市待兼山町1-3

yamamoto@yachi-lab.sys.es.osaka-u.ac.jp, iwai@sys.es.osaka-u.ac.jp,

yachida@sys.es.osaka-u.ac.jp

本稿では、高精度な移動物体検出を実時間で実現するために、ハードウェアを用いて検出処理を行うことを提案する。ABM(Adaptive Background Model: 明るさ可変背景モデル)を用いて背景成分を推定、更新し、MSC(Margined Sign Correlation: マージン付き符号相関)を利用することで移動物体検出を行う。また、ハードウェアとして、プログラム可能なLSIであるFPGAを用いる。FPGAを用いて実装を行うためにアルゴリズムの検討を行い、実験により最適な実装方法を検証した。

Examination of Real-Time Object Detection by Adaptive Background Model and Margined Sign Correlation

Ayaka YAMAMOTO[†], Yoshio IWAI[†], Masahiko YACHIDA[†]

[†]Graduate School of Engineering Science, Osaka University

1-3 Machikaneyama, Toyonaka, Osaka 560-8531 Japan

yamamoto@yachi-lab.sys.es.osaka-u.ac.jp, iwai@sys.es.osaka-u.ac.jp,

yachida@sys.es.osaka-u.ac.jp

In this paper, we propose a fast and efficient method that detects moving objects. To achieve efficient performance, we estimate background components adaptively using ABM(Adaptive Background Model), and a novel method for detecting objects using MSC (Margined Sign Correlation). We use programmable LSI called FPGA, to achieve fast performance by optimizing the algorithm. We show experimental results using real images to verify the effectiveness of the proposed system.

1 はじめに

画像中から移動物体を検出する手法は、従来から数多く提案されており、様々な分野において応用されている。その例として、防犯カメラ等のセキュリティシステムへの応用が挙げられる。防犯カメラ等の映像を人が長時間監視することは、監視者への負担や人件費などコストの面で問題がある。画像中から移動物体を検出し、監視システムの自動化を図れば、監視者の負担の軽減やコストの削減が期待できる。

このようなシステムへの応用を実現するためには、システムの性格上、移動物体の検出手法は高精度かつ実時間処理を行えることが望まれる。これに対して、従来から多くの手法が提案されている。なかでも、伊藤らは明るさ可変背景モデルを用いた手法[1]を提案しており、屋外環境において照明変動に対し

てロバストな検出を行い、かつ、移動物体とその影を正確に分離している。検出には、色情報の明るさをパラメータ化することで、様々な明るさの背景画像を高精度に再現できる明るさ可変背景を用いており、この背景画像はカルマンフィルタを用いて随時更新される。この手法において扱うデータは色情報のみであるため、背景と移動物体の色が非常に近い領域においては、移動物体であっても背景と認識されていた。また、背景画像の更新では、ゲイン行列がランク不足を起こし、短時間で推定値が発散する問題があった。

吉村らは、対象画素とその周辺画素は何らかの相関をもつという仮定の下、移動物体の検出に周辺増分符号関数[2]の拡張であるマージン付き符号関数の利用を提案した[3]。伊藤らの手法に空間情報としてマージン付き符号相関を用いた識別を組み込み、よ

り高精度な移動物体の検出を実現した。推定値が発散する問題は、背景成分の推定を前時刻の背景成分を用いて逐次的に行うことで回避した。吉村らの手法は、より高精度な移動物体領域の検出を実現したが、処理がソフトウェアで行われているため検出に時間がかかり、実時間で処理が必要となるシステムへの応用は困難であった。

そこで、我々は移動物体の検出を実時間で実現するために、ハードウェアを用いて検出処理の実装を行うことを提案する。吉村らの手法をFPGAを用いて実装することで、移動物体領域の検出を高精度かつ実時間で行うことができると考えられる。

2 移動物体の検出

屋外環境において移動物体の検出を安定に行うためには、明るさの変動に対してロバストであることと、移動物体とその影を正確に分離することが求められる。伊藤らは色情報に基づいた明るさ可変背景モデルを用いて検出を行い、吉村らは伊藤らの手法に空間情報としてマージン付き符号相関を用いた識別を組み込み、高精度な移動物体の検出を実現した。本章では、この二つの検出方法について詳しく説明する。

2.1 明るさ可変背景モデル (ABM)

様々な明るさの背景を数学的にモデル化した明るさ可変背景モデル (Adaptive Background Model: 以下 ABM) について説明する。

ABM を説明するにあたり、雲の影響によって急激な明るさの変化が起きている場合について考える。なお、屋外環境においては太陽光以外の光源は無視できると仮定する。このとき、屋外環境における観測光は、太陽直射光成分 (以下 直射光) と天空光成分により構成される。ここで、天空光成分とは太陽光が空気中の粒子等に衝突し、散乱した後に地上に届く光である。短期的には天空光および直射光の色度の変化は小さいことから、これを無視すると、時刻 t における画素 $\mathbf{x} = (x, y)^T$ において観測される対象の色 $\mathbf{Y}(\mathbf{x}, t) = (\mathbf{Y}_R(\mathbf{x}, t) \ \mathbf{Y}_G(\mathbf{x}, t) \ \mathbf{Y}_B(\mathbf{x}, t))^T$ は、天空光成分 $\mathbf{X}e(\mathbf{x}, t) = (\mathbf{X}e_R(\mathbf{x}, t) \ \mathbf{X}e_G(\mathbf{x}, t) \ \mathbf{X}e_B(\mathbf{x}, t))^T$ と直射光成分 $\mathbf{X}d(\mathbf{x}, t) = (\mathbf{X}d_R(\mathbf{x}, t) \ \mathbf{X}d_G(\mathbf{x}, t) \ \mathbf{X}d_B(\mathbf{x}, t))^T$ からなると考えられる。ここで A^T は、ベクトル A の転置を表す。雲による分光吸収はごくわずかであ

るから、直射光が雲を通過するときだけ明るさが変化すると考えれば、雲の影響により明るさが変化しているときに観測される対象の色 $\mathbf{Y}(\mathbf{x}, t)$ は式 (1) でモデル化できる。

$$\mathbf{Y}(\mathbf{x}, t) = \mathbf{X}e(\mathbf{x}, t) + k(\mathbf{x}, t)\mathbf{X}d(\mathbf{x}, t) \quad (1)$$

式 (1) において、 $k(\mathbf{x}, t)$ ($0 \leq k \leq 1$) は直射光成分の混合率を表す。このモデルに従えば、明るさ可変背景は天空光成分から成る背景画像と直射光成分から成る背景画像を使って表現できる。これらの画像を背景成分画像とする。この背景成分画像を使えば、直射光成分の混合率 $k(\mathbf{x}, t)$ を明るさパラメータとして利用することで、様々な明るさの背景画像を高精度に再現することが可能となる。明るさが急激に変化しても明るさパラメータ $k(\mathbf{x}, t)$ を調整するだけで、様々な明るさにロバストに対応することができる。

この手法では、移動物体の影となって直射光が遮られている部分についても、雲によって直射光が遮られる部分と同様であると考えられる。従って、明るさの変動に対して各ピクセルにおいて独立に処理を行えば、移動物体の影も除去することができる。

2.2 マージン付き符号相関 (MSC)

ABM を用いた識別手法 [1] では、色情報のみを用いて背景と移動物体とを分離する。そのため、RGB 空間上で背景と移動物体のテクスチャの色が近接している場合は移動物体をうまく分離することができず、しばしば物体領域を背景として認識してしまう。そこで物体の検出精度の向上を図るため、マージン付き符号相関 (Margined Sign Correlation: 以下 MSC) を用いる。MSC は周辺増分符号相関 (Peripheral Increment Sign Correlation: 以下 PISC) [2] の応用であり、以下の式で定義する。

$$MSC_m(f(\mathbf{x}), g(\mathbf{x})) = \frac{(\text{sgn}_m(f(\mathbf{x})), \text{sgn}_m(g(\mathbf{x})))}{\|\text{sgn}_m(f(\mathbf{x}))\| \cdot \|\text{sgn}_m(g(\mathbf{x}))\|} \quad (2)$$

$$\|\text{sgn}_m(f(\mathbf{x}))\| = \sqrt{(\text{sgn}_m(f(\mathbf{x})), \text{sgn}_m(f(\mathbf{x})))} \quad (3)$$

$$(\text{sgn}_m(f(\mathbf{x})), \text{sgn}_m(g(\mathbf{x})))$$

$$= \sum_{\mathbf{x}^* \in N(\mathbf{x})} \text{sgn}_m(f(\mathbf{x}^*) - f(\mathbf{x})) \text{sgn}_m(g(\mathbf{x}^*) - g(\mathbf{x})) \quad (4)$$

$$\text{sgn}_m(x) = \begin{cases} +1 & |m| \leq x \\ 0 & -|m| \leq x < |m| \\ -1 & x < -|m| \end{cases} \quad (5)$$

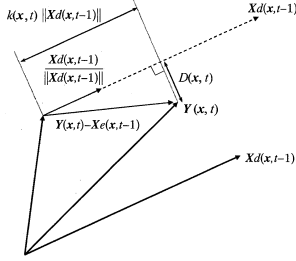


図 1: 明るさパラメータの推定

ただし, (\cdot) はベクトルの内積を表す。また, \mathbf{x}^* は対象画素 \mathbf{x} の近傍 $N(\mathbf{x})$ に含まれる点で, m はノイズを考慮したマージンを表す。 $m = 0$ の場合, $MSC_0(f(\mathbf{x}), g(\mathbf{x}))$ は PISC と等価になる。すなわち, 本論文で提案する MSC は, 観測ノイズを軽減するためにノイズマージンを組み込んだ PISC の拡張である。また式 (2) から分かるように, $MSC_m(f(\mathbf{x}), g(\mathbf{x}))$ は $\|\text{sgn}_m(f(\mathbf{x}))\| = 0$ や $\|\text{sgn}_m(g(\mathbf{x}))\| = 0$ のときには定義できない。このとき, 画像 f または g が近傍 $N(\mathbf{x})$ 内で常にノイズの影響を受けることを表し, $MSC_m(f(\mathbf{x}), g(\mathbf{x}))$ の相関値は信頼性が低くなる。

2.3 背景と移動物体の識別

時刻 t の入力画像 $\mathbf{Y}(\mathbf{x}, t)$ と前時刻 $t-1$ の天空光成分 $\mathbf{X}e(\mathbf{x}, t-1)$, 直射光成分 $\mathbf{X}d(\mathbf{x}, t-1)$ を用いて時刻 t の背景成分を推定し, 入力画像中の画素 \mathbf{x} が移動物体かの識別を行う。識別には色情報を用いた識別と空間情報を用いた識別の二つの手法を用いる。

2.3.1 背景成分の推定

時刻 t における明るさパラメータ $k(\mathbf{x}, t)$ を推定し, 式 (1) により時刻 t での背景を構成する。時刻 t の入力画像 $\mathbf{Y}(\mathbf{x}, t)$ と前時刻 $t-1$ の背景成分 $\mathbf{X}e(\mathbf{x}, t-1)$, $\mathbf{X}d(\mathbf{x}, t-1)$ を用いて, 式 (6) により明るさパラメータ $k(\mathbf{x}, t)$ を算出する。

$$k(\mathbf{x}, t) = \frac{((\mathbf{Y}(\mathbf{x}, t) - \mathbf{X}e(\mathbf{x}, t-1)), \mathbf{X}d(\mathbf{x}, t-1))}{\|\mathbf{X}d(\mathbf{x}, t-1)\|^2} \quad (6)$$

ただし, (\cdot) はベクトルの内積を表す。ここで, 短期的には天空光成分及び直射光成分の色度の変化は小さいため, $\mathbf{X}e(\mathbf{x}, t) \approx \mathbf{X}e(\mathbf{x}, t-1)$, $\mathbf{X}d(\mathbf{x}, t) \approx \mathbf{X}d(\mathbf{x}, t-1)$ と仮定している。式 (6) の幾何学的関係を図 1 に示す。

2.3.2 色情報を利用した識別

観測される対象画像の色 $\mathbf{Y}(\mathbf{x}, t)$ と前時刻の天空光成分 $\mathbf{X}e(\mathbf{x}, t-1)$ と直射光成分 $\mathbf{X}d(\mathbf{x}, t-1)$ との色成分の相関を計算することで影除去と移動物体検出を行う。色差ベクトルのユークリッドノルムを $D(\mathbf{x}, t)$ として, 式 (7) により算出する。

$$D(\mathbf{x}, t) = \|\mathbf{Y}(\mathbf{x}, t) - (\mathbf{X}e(\mathbf{x}, t-1) + k(\mathbf{x}, t)\mathbf{X}d(\mathbf{x}, t-1))\| \quad (7)$$

ここで, 前時刻 $t-1$ の背景成分を $\mathbf{X}e(\mathbf{x}, t) \approx \mathbf{X}e(\mathbf{x}, t-1)$, $\mathbf{X}d(\mathbf{x}, t) \approx \mathbf{X}d(\mathbf{x}, t-1)$ と仮定している。求めたユークリッドノルム $D(\mathbf{x}, t)$ を用いて, 式 (8) により各画素について背景か移動物体か最終的な識別を行なう。

$$\begin{cases} \text{back.} & D(\mathbf{x}, t) < D_{th} \\ \text{object} & \text{otherwise} \end{cases} \quad (8)$$

ここで, D_{th} は対象画素が背景か移動物体かを決定する閾値である。ノルムが閾値 D_{th} より小さい場合は, RGB 空間上における明るさ可変背景と入力画素が非常に近接していると判断できるので, 対象画素を背景と識別する。逆にノルムが閾値 D_{th} より大きい場合には非背景, つまり移動物体と識別する。

2.3.3 空間情報を利用した識別

色情報による識別は, 背景の色と移動物体の色が似ている場合においては, 移動物体であろうと背景と識別してしまう。これを改善するために, 色情報以外にも空間情報を用いて識別を行う。具体的にはテクスチャの違いを利用して, 式 (9) により識別を行なう。

$$\begin{cases} \text{back.} & MSC_m(\mathbf{Y}(\mathbf{x}, t) - \mathbf{X}e(\mathbf{x}, t), \mathbf{X}d(\mathbf{x}, t)) \leq E_{th} \\ \text{object} & \text{otherwise} \end{cases} \quad (9)$$

ただし, $\mathbf{Y}_i, \mathbf{X}e_i, \mathbf{X}d_i$ ($i=R, G, B$) はそれぞれの RGB 成分の値を示す。ここでも $\mathbf{X}e(\mathbf{x}, t) \approx \mathbf{X}e(\mathbf{x}, t-1)$, $\mathbf{X}d(\mathbf{x}, t) \approx \mathbf{X}d(\mathbf{x}, t-1)$ ($i=R, G, B$) の仮定を用いる。

2.3.4 背景と移動物体の識別

色情報を利用した識別と空間情報を利用した識別の二つの結果に基づいて, 最終的に入力画素が背景か移動物体かの識別を行う。吉村らの手法では, 二つの識別結果を組み合わせた単純な識別を行って

表 1: 背景と移動物体の識別

texture difference	chroma difference	result
background	background	background
background	object	object
object	background	object
object	object	object

る。色情報を利用した識別と空間情報を利用した識別の結果が、両方とも背景と識別された場合にのみ、その画素を背景と決定する。逆にどちらか一方でも移動物体と識別された場合には、背景と識別したもう一方の識別結果を無視して、その画素を移動物体と決定する。最終的な識別方法を表にしたものを表 1 に示す。

3 ハードウェア実装

ABM と MSC による移動物体領域の検出処理をハードウェアを用いて実装することで、処理が高速化され実時間システムへの応用が期待できる。本章では、2章で説明した ABM と MSC を用いた移動物体検出処理のハードウェアへの実装方法について述べる。

3.1 システムの構成

我々はハードウェアとして FPGA を用いることを提案する。FPGA を用いて検出処理を実装することで、精度を低下させずにファイル処理を高速に行い、ソフトウェアで実装していた際に問題となっていた処理時間を短縮する。ハードウェア記述言語には VHDL (Very high speed integrated circuit Hardware Discription Language) を使用した。FPGA 内では、画像が入力されてから結果を出力するまでの全ての処理が行われる。FPGA 内の処理の流れを図 2 に示す。

まず、入力された画像データから時刻 t における明るさパラメータ $k(\mathbf{x}, t)$ を求める。明るさパラメータ $k(\mathbf{x}, t)$ が求まると、各ピクセルについてそれぞれ背景か非背景かの識別を行う。この識別には色情報および空間情報を用いる。二つの識別結果から最終的に背景と識別された画素は白色で出力し、移動物体と識別された画素は入力された画素の色をそのまま出力する。このとき背景と識別された画素については、カルマンフィルタを利用して天空光成分 $\mathbf{X}e(\mathbf{x}, t-1)$

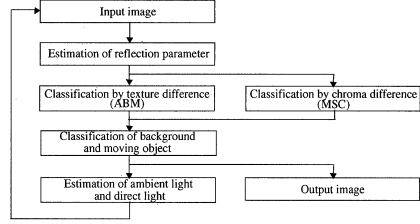


図 2: FPGA 内の処理の流れ

の更新を行う。天空光成分の観測値 $\mathbf{Y}e(\mathbf{x}, t)$ を次式により推定する。

$$\mathbf{Y}e(\mathbf{x}, t) = \mathbf{Y}(\mathbf{x}, t) - k(\mathbf{x}, t)\mathbf{X}d(\mathbf{x}, t-1) \quad (10)$$

これにカルマンフィルタを適用することで時刻 t における天空光成分の状態 $\mathbf{X}e(\mathbf{x}, t)$ を推定する。このときの天空光成分の観測方程式を式 (11) に、状態方程式を式 (12) に示す。

$$\mathbf{X}e(\mathbf{x}, t) = \mathbf{Y}e(\mathbf{x}, t) + \varepsilon_{O_e} \quad (11)$$

$$\mathbf{X}e(\mathbf{x}, t) = \mathbf{X}e(\mathbf{x}, t-1) + \varepsilon_{S_e} \quad (12)$$

ただし、 $\varepsilon_{O_e} \sim N(0, \Sigma_{O_e})$ は観測ノイズ、 $\varepsilon_{S_e} \sim N(0, \Sigma_{S_e})$ は状態ノイズを表す。天空光成分 $\mathbf{X}e(\mathbf{x}, t)$ の更新を行った後、同様に直射光成分 $\mathbf{X}d(\mathbf{x}, t-1)$ についても、直射光成分を十分に含んでいる場合に限り、更新を行う。影領域においては直射光成分がほとんど含まれていないので、式 (13) を満たす場合にのみ更新を行う。

$$k(\mathbf{x}, t) \geq K_{th} \quad (13)$$

ここで K_{th} は直射光成分が最小含有率となる際の $k(\mathbf{x}, t)$ の値である。直射光成分の観測値 $\mathbf{Y}d(\mathbf{x}, t)$ は次式により推定する。

$$\mathbf{Y}d(\mathbf{x}, t) = \frac{\mathbf{Y}(\mathbf{x}, t) - \mathbf{X}e(\mathbf{x}, t)}{k(\mathbf{x}, t)} \quad (14)$$

これにカルマンフィルタを適用することで時刻 t における直射光成分の状態 $\mathbf{X}d(\mathbf{x}, t)$ を推定する。このときの直射光成分の観測方程式を式 (15) に、状態方程式を式 (16) に示す。

$$\mathbf{X}d(\mathbf{x}, t) = \mathbf{Y}d(\mathbf{x}, t) + \varepsilon_{O_d} \quad (15)$$

$$\mathbf{X}d(\mathbf{x}, t) = \mathbf{X}d(\mathbf{x}, t-1) + \varepsilon_{S_d} \quad (16)$$

ただし、 $\varepsilon_{O_d} \sim N(0, \Sigma_{O_d})$ は観測ノイズ、 $\varepsilon_{S_d} \sim N(0, \Sigma_{S_d})$ は状態ノイズを表す。以上の処理を全ての画素 \mathbf{x} について行う。

吉村らの手法ではソフトウェアを用いて実装を行っていたため、これらの処理を逐次行っていた。提案システムでは、ハードウェアを用いて実装を行うことにより、色情報および空間情報による識別処理、及び最終的な識別結果の出力と背景成分の更新処理をそれぞれ並列に行うことで処理を高速化している。

3.2 アルゴリズムの検討

ハードウェアを用いることの最大のメリットは、処理を高速化できることであるが、その一方で、実装可能な処理に限られるというデメリットもある。そこで、移動物体の検出処理をハードウェアを用いて実装するにあたり、ハードウェア特有の制約下で高精度な検出処理を行うアルゴリズムの検討を行った。

3.2.1 変数

本システムでは、処理の高速化を図るために固定小数点方式を用いる。固定小数点方式を用いる場合、整数部と小数部にそれぞれ何ビットを割り当てるかを予め決めておく必要がある。特に小数部を何ビット扱うかが問題となり、小数部に割り当てるビット数を増やせば検出精度は向上するが、扱う信号数が増え計算に時間がかかってしまう。そこで、ソフトウェアを用いて小数部に割り当てるビット数と検出精度の関係を検証した。

3.2.2 色情報を利用した識別

色情報を利用した識別を行う際、吉村らの手法では対象画素 x だけではなくその近傍画素 x^* の情報も用いていた。対象画素と近傍画素の相関度を示す重みパラメータ $w(x^*, t-1)$ を用いることで、式(17)により各画素について背景か移動物体かの識別を行なっている。

$$\begin{cases} \text{back.} & \sum_{x^* \in \{N(x) \cup x\}} w(x^*, t-1) D(x^*, t) < D_{th} \\ \text{object} & \text{otherwise} \end{cases} \quad (17)$$

ただし、 D_{th} は対象画素が背景か移動物体かを決定する閾値である。ここで、ユークリッドノルム $D(x^*, t)$ を求めるには平方根を計算する必要がある。しかし、ハードウェアでは平方根の計算に多数のクロックを

必要とし、高速な計算をするためには大量のハードウェア資源を必要とする。そのため、式(17)を用いた識別処理を容易には実装できない。そこで、ハードウェアで簡便に計算できるように、式(18)の形の識別手法を提案する。

$$\begin{cases} \text{back.} & D(x, t)^2 < D_{th}^* \\ \text{object} & \text{otherwise} \end{cases} \quad (18)$$

ただし、 D_{th}^* は新たに設定した閾値であり、式(17)中の D_{th} とは異なる。式(18)ではユークリッドノルムの二乗の値により識別を行うことで、平方根を用いず積和計算のみを用いて識別を行うことができる。また、式(18)では近傍画素の情報を用いず対象画素の情報のみを用いて識別を行う。これは、空間情報を利用した識別を別に行っており、色情報を利用した識別においても空間情報を扱うことが冗長であるという考えに基づく。また、近傍画素の情報を考慮する際に用いる重みパラメータ $w(x^*, t-1)$ は、背景成分を更新する度に更新する必要があったが、式(18)では、重みパラメータの更新処理が省略できるという点、近傍画素のデータを必要としないという点で処理を大幅に高速化することができる上に、ハードウェア資源も節約できる。

3.2.3 空間情報を利用した識別

空間情報を利用した識別処理は、ソフトウェア上では多くの時間を必要としていた。これはMSCの算出に用いる隣接画素への符号の設定が逐次行われていたためである。そのため、識別にかかる時間は窓サイズに依存する。そこで、ハードウェア上では符号の設定を窓内の全画素で並列に行い、識別の高速化を図る。そのためには、予め窓サイズを決めておく必要がある。吉村らの実験により、最適な窓サイズはシーンに依存するが、今回用いたシーンにおいては 7×7 から 9×9 程度が最適であることが分かっている[3]。また、窓サイズが大きいほど移動物体領域と背景領域の検出率は向上し、その反面、影領域の除去率が落ちることが実証されている。しかし、FPGAで扱える入力信号数に制限があり、窓サイズが 5×5 以下でなければ識別に用いる全画素を同時に入力できない。ここで、窓サイズを 5×5 に設定すると、最適な窓サイズである 9×9 の場合と比較して移動物体と背景の検出精度が低下することが考えられる。し

かし、本システムでは高速に識別を行うことを優先し、 5×5 の窓内の全画素に対して並列に符号を設定することにする。

4 実験

アルゴリズムを変更したことによる検出精度への影響をソフトウェアを用いて検証した。実験には 720×486 [pixel]、8bit カラー動画像を用いる。この動画像の撮影日時は、2005年5月8日の正午頃であり、その日の天気は晴れ時々曇りで時刻により明るさが大きく変動している。ABMによる識別において用いられる閾値 D_{th} と D_{th}^* は、それぞれ10、 10^2 と設定した。MSCによる識別で用いる閾値 E_{th} は0.4とし、マージン m はSDの値を用いた。SDは標準偏差の最大値を表し、これは背景成分の更新に用いたカルマンフィルタの共分散の平方根である。またRGB成分はそれぞれ独立であると仮定したので、共分散行列 Σ_{O_e} と Σ_{O_d} はそれぞれ対角成分を 10^2 とする対角行列とした。また同様に、共分散行列 Σ_{S_e} と Σ_{S_d} についてもそれぞれ対角成分を 1^2 とする対角行列とした。これらの値は経験的に設定した。

4.1 実験結果

図3に実際に提案システムを用いて処理を行ったときの、2960フレーム目の入力画像を示す。また、図4にそれぞれ小数部に割り当てるビット数を3通りに変化させて処理した場合の検出結果を示し、そのときの移動物体領域と背景領域の検出率、及び影領域の除去率を表2に示す。ここで、小数部に割り当てるビット数は、10進数で小数第1位までを考慮した4ビット、小数第2位までを考慮した7ビット、第3位までを考慮した10ビットの3通りを用いた。図5は、色情報を利用した識別で式(18)を用いた手法と式(17)を用いた従来の検出手法の検出結果を示す。また、表3にそれぞれの移動物体領域と背景領域の検出率、及び影領域の除去率を示す。図6には、空間情報を利用した識別で、符号を設定する窓サイズを実装可能な 5×5 に設定した場合と最適な窓サイズ 9×9 に設定した場合の検出結果を示す。また、そのときの移動物体領域と背景領域の検出率、及び影領域の除去率を表4に示す。最後に、提案手法と従来手法の検出結果を比較して図7に示す。また、それぞれの移動物体領域と背景領域の検出率、及び影領域の除去率を表5に示す。これらの図において、白

く表示された領域は背景を、水色で表示された領域はサチュレーションを起こしている領域である。

4.2 考察

表2より、小数部に割り当てるビット数を増やすことによって、検出率の向上が見てとれる。ビット数を4ビットから7ビットに増やした場合には、移動物体領域と背景領域の検出率、影領域の除去率の全てにおいて大幅な検出精度の向上が見られた。しかし、ビット数を7ビットから10ビットに増やした場合には、検出精度の低下が見られる。これはノイズの影響によると考えられる。以上のことから、FPGAで実装する際、変数の小数部に割り当てるビット数は7ビットに設定することにした。

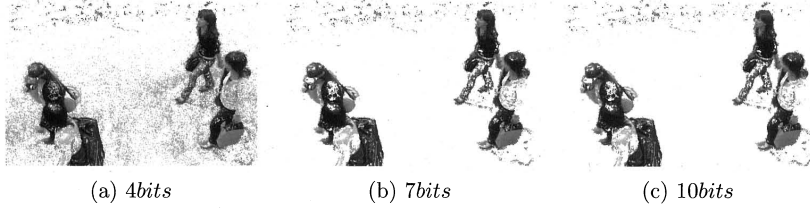
空間情報を利用した識別では、表4より、窓サイズを9から5に変更することで従来手法と比較して移動物体領域の検出精度の低下が見られたが、その低下率は約1.8%であり、処理の高速化を優先するのに十分小さいと考えられる。さらに、影領域の除去率は13%向上した。以上のことから、ハードウェア上で実装を行う際に窓サイズを5に設定する。

色情報を利用した識別では、図5より、背景領域において、画像上部の草木の領域が、誤検出されにくくなっていることが分かる。これは、従来手法では草木の揺れによって、入力画像と背景成分との間のユークリッドノルムが非常に大きくなり、識別結果へ影響していると考えられるが、MSCを用いた識別は広範囲の空間情報を扱い、また窓内の全画素で重みが等しいため、草木の細かい揺れに対して柔軟な識別が行えるためである。そのため、提案手法ではMSCを用いた識別のみで空間情報を用いることで、草木の揺れによる誤検出を減らすことができたと考えられる。また、移動物体領域の検出率の向上も見られた。図5より、画像右上部の女性の膝部分や左上部の女性の帽子部分などの、直射光成分が多く白っぽくなっている部分で検出精度が向上していることが分かる。これについても同様に、MSCを用いた識別のみで空間情報を用いることで、誤検出された画素のユークリッドノルムの影響を周辺に伝播させていないためであると考えられる。以上のことから、式(18)のように単純化した識別処理で高精度な識別を行うことができることを確認した。

以上のことから、従来手法と比較して、提案手法による検出は高精度に行えていることが分かる。よっ



図 3: 入力画像 (2960frame)



(a) 4bits

(b) 7bits

(c) 10bits

図 4: ビット数に対する検出結果

表 2: ビット数に対する検出結果

		object region		background region		removal region	
		detection pixels	detection rate(%)	detection pixels	detection rate(%)	detection pixels	removal rate(%)
number of decimal bits	4	54864	82.44	155168	64.54	4999	53.84
	7	59190	88.94	220625	91.76	7179	77.32
	10	58580	88.03	220214	91.59	7161	77.13



(a) 従来アルゴリズム

(b) 提案アルゴリズム

図 5: 提案手法を用いた ABM の検出結果

表 3: 提案手法を用いた ABM の検出結果

		object region		background region		removal region	
		detection pixels	detection rate(%)	detection pixels	detection rate(%)	detection pixels	removal rate(%)
Previous method		57745	86.77	230772	95.98	7373	79.41
Proposed method		59188	88.94	235195	97.82	7185	77.39



(a) 9

(b) 5

図 6: 窓サイズを変更した場合の検出結果

表 4: 窓サイズを変更した場合の検出結果

		object region		background region		removal region	
		detection pixels	detection rate(%)	detection pixels	detection rate(%)	detection pixels	removal rate(%)
window size	9	59188	88.94	235195	97.82	7185	77.39
	5	57960	87.10	199744	83.08	5404	90.52

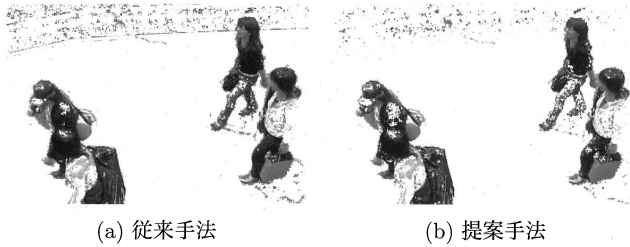


図 7: 従来手法と提案手法の比較

表 5: 従来手法と提案手法の比較

	object region		background region		removal region	
	detection pixels	detection rate(%)	detection pixels	detection rate(%)	detection pixels	removal rate(%)
Previous method	58809	88.37	188199	78.28	7913	85.23
Proposed method	57960	87.10	199744	83.08	8404	90.52

て、ハードウェアを用いて検出処理を実装しても高精度な検出を行えると考えられる。

- [3] 吉村浩典, 岩井儀雄, 谷内田正彦, “屋外環境における明るさ可変背景と MSC を用いた移動物体検出”, 電子情報通信学会論文誌. D Vol. J90-D No.8 pp.1987-1997, 2007.

5 おわりに

我々は、ハードウェアを用いることで高精度な移動物体の検出手法を高速化することを提案した。移動物体の検出手法には、ABM と MSC を用いることで屋外環境において高精度な検出を行うことができる吉村らの手法を用いた。この吉村らの手法を FPGA を用いて実装するにあたり、ハードウェア特有の制約下で高精度な検出を行うアルゴリズムの検討を行った。そして実際に屋外環境において撮影された動画画像を用いて実験を行った。これにより従来手法と提案手法との検出精度を比較評価し、提案手法を用いても精度を低下させずに検出処理を実装できることを確認した。

今後は、実際に FPGA を用いて検出処理を行い、ソフトウェアを用いた場合と比較して処理の速度や精度といった性能の評価を行う。

参考文献

- [1] 伊藤竜之介, 岩井儀雄, 谷内田正彦, “照明変動と影の影響を抑えた移動物体の検出”, 情報学研報. No.127-17 pp.119-126, 2001.
- [2] S.Kaneko, Y.Satoh, S.Igarashi: “Roust object detection image sequence using peripheral increment sign correlation”, Proc. of the 5th Japan-France Congress on Mechatronics, pp.287-292, 2001.