

修士論文

題目

空間特徴を用いた混合ガウス分布モデル
による背景モデルの構築

指導教員

近藤 利夫 教授

2017 年度

三重大学大学院 工学研究科 情報工学専攻
コンピュータアーキテクチャ研究室

ZHENG KAN (415M5E1)

内容梗概

背景モデルを用いて監視ビデオから移動物体を検出する手法が多数提案されている。これらの手法は、平均法または学習率を用いる移動平均法などで既に観測されたフレームから生成する背景画像と現在のフレームの差分をとることで移動物体を簡単かつ高速に抽出できるものの、屋外のシーンは植物の枝や葉の揺れ、水の波打ちなど背景中で常時変化しており、その変化部分に属するピクセルを誤抽出する問題がある。この問題を解決するため、混合ガウス分布モデル(GMM)を利用する統計的な手法が提案されている。一般的に混合ガウス分布は単一のピクセル値に対して時間領域での変化だけでモデリングするので、モデルが画像のノイズや急激に変化している背景ピクセルに対応できるとは限らず、それが背景モデルの安定的な構築の障害となっている。そこで、本論文では、入力画像の GMM と複数の局所的な空間的特徴の GMM のそれぞれから構築する複数の背景モデルを総合的に判定することで背景変化への追従性を高める手法を提案した。この提案手法では、最後の移動物体と背景の判定において、それぞれのモデルによる判定結果が同じなら、その判定結果で移動物体に属するピクセルか否かを確定する。一方、判定結果が異なる場合は、一定の範囲内で画素値が類似している確定ピクセルを探し出し、それらの過半数が移動物体に属する場合に、移動物体中のピクセルとして確定する。

また、この提案手法により得た背景モデルを評価用データセットの解である Ground Truth と比較し Precision Rate と Recall Rate を算出して F-measure 手法で評価した。その結果、提案手法により従来法の 89.82%が 98.07%まで改善されることが明らかになった。

Abstract

Many methods for detecting a moving object from surveillance video using a background model have been proposed. Mixed Gaussian distribution is widely used to construct background models, but it cannot cope well with scenes due to the frequent background changes. In this paper, we proposed a method with high followability to background change by using spatial information of pixels as feature, constructing a background model with mixed Gaussian distribution at multilevel. Finally, when judging a moving object and a background, if the outputs among the models are equal to one another, the same output becomes the result directly. For a case when the outputs are different to one another, the output with larger number of determined pixels whose pixel value is similar within a certain range becomes the result.

The results of the proposed method were compared with the Ground Truth of the data set and evaluated by the F - measure method, and it was confirmed that the precision rate and the recall rate of the proposed method exceed the conventional methods such as Gaussian Mixture Model (GMM) [3] and Local Binary Pattern (LBP) [4].

In addition, as the local spatial features, it was shown that the effects of the average pooling and the maximum pooling are larger than the minimum pooling, and especially the average value is the most effective since it is increased conventional method from 89.82% to 98.07%.

目次

1	まえがき	1
2	固定カメラから移動物体の検出	3
2.1	確率モデルにより移動物体の検出	3
2.2	混合ガウス分布モデルとそれによる背景モデル構築	3
2.2.1	混合ガウス分布モデル	3
2.2.2	モデルの更新	4
2.2.3	モデルによる背景/移動物体の判別	6
3	関連研究	7
3.1	他の背景モデルにより移動物体の検出	7
3.2	局所的な特徴量を考慮した従来法	7
3.3	従来法の問題点	8
3.3.1	単一の座標位置のピクセル値分布を生成するに對しての手法の問題点	8
3.3.2	隣接領域を考慮した手法の問題点	8
4	提案手法	10
4.1	改善の對象	10
4.2	基本的なアイデア	10
4.2.1	フィルタにより画像処理	10
4.2.2	プーリングにより画像処理	11
4.3	背景/移動物体の判定	12
5	性能評価	16

5.1	評価環境.....	16
5.2	予備試験.....	17
5.3	評価結果.....	19
5.3.1	探索範囲について.....	22
5.3.2	モデルの数について.....	24
6	あとがき.....	25
	お謝辞.....	25
	参考文献.....	27

図目次

図 1	混合ガウス分布によるモデルの構築.....	4
図 2	前景, 背景に生じる誤判定.....	9
図 3	移動平均フィルタによる画像処理.....	11
図 4	プーリング処理.....	12
図 5	提案手法の流れ.....	13
図 6	プーリング手法の比較.....	18
図 7	探索範囲の変化より結果への影響.....	23

表目次

表 1	判定アルゴリズム.....	14
表 2	評価指標の比較.....	20
表 3	処理結果との比較.....	21
表 4	5つのモデルの性能評価.....	24

1 まえがき

コンピュータ技術の進歩によりカメラで撮った画像から不審侵入物体などを容易に検出できるようになったことから移動物体の検知システムは防犯や監視等の用途に広く利用されている。

移動物体検知の要の技術は、背景との差分により移動物体抽出する技術であり、事前に取得した背景画像と比較することでビデオストリーム内の移動物体を検出する[10]。この手法では、一般に移動物体より背景が圧倒的に多くのピクセルを占める。このため、平均法または学習率を用いる移動平均法などで既に観測されたフレームから生成する背景画像と現在のフレームの差分をとることで移動物体を簡単かつ高速に抽出できる。しかし、自然のシーンは植物の枝や葉の揺れ、水の波打ちなどの現象があるため、背景中のピクセルは常に変化している[1]。このため、平均法などのフレーム差分法では頻繁に変化する背景に対応できないことが少なくない。この問題を解決するため、統計的な手法を使って背景モデルを構成する手法が提案されている[10]。この手法では入力画像の中で、モデルに適合する部分を背景とし、そうでなければ移動物体とする。背景モデルを生成する過程は背景モデリングと呼ばれる。

統計的な手法を利用する背景モデリングの一つである C.R. Wren が提案したシングルガウス分布によるピクセルの変化への対応は、シングルガウス分布が単峰(unimodal)であるため、多様な背景ピクセルの変化を捉えられない弱点がある。この問題の解決に向け、Stauffer らはピクセルの変化にロバストな混合ガウス分布を用いる背景モデル構築手法(Gaussian Mixture Model: GMM)を提案した[3]。この手法では各々のピクセルの時間領域内の変化を複数のガウ

ス分布でモデリングすることによってピクセルのポリモーフィズムを近似的に表現できる. 他にも, 画素値をコードブック(Codebook)に格納したり, 画素値ではなくて Local Binary Pattern(LBP)を利用して画像のテクスチャをバイナリパターンに変換して局所的な特徴量としたりするモデリング手法が提案されている[4].

しかし, GMM, Codebook などのピクセル値でモデリングする手法は屋外の急激な変化, センサのノイズなどに影響を受けやすい一方, LBP 法のように空間特徴を利用する手法も抽出領域のエッジや輪郭など重要な情報を失いやすい弱点がある.

ピクセルまたテクスチャの単一のモデリングでは, これまで以上の大幅な性能向上は望めないとの観点から, 本論文では, ピクセルの時間領域における変化だけをモデリングではなく, 空間的な特徴の両方も利用してモデリングする手法を新たに提案する. 以下では, 2 節で文献[3]が提出した混合ガウス分布による背景モデルの構築と関連研究を紹介する. 3 節で従来法の不足を説明する. 4 節で提案手法について述べる. 5 節では, 提案手法の性能を評価, 分析する. 最後の 6 節では本稿をまとめる.

2 固定カメラから移動物体の検出

2.1 確率モデルにより移動物体の検出

一定の時間内には、一つのピクセルでは、異なる移動物体より背景の出現頻度の方が高い。この性質を利用し、与えられたピクセルの値を記録して一定の時間を経たら、出現頻度が高いピクセル値が背景の色とし、頻度が低いピクセル値を移動物体とすることによって分別することが可能である[3]。また、背景の多様性に対し、確率モデルでは異なる確率により背景色の変化を表現できる。

このような特性に基づき、文献[5][6]がそれぞれ混合ガウス分布モデル(GMM), Code-Book を提出した。

2.2 混合ガウス分布モデルとそれによる背景モデル構築

2.2.1 混合ガウス分布モデル

図 1 に示すように、混合ガウス分布は、画像の個々のピクセル値で重み付けした K 個のガウス分布を重畳すること得られる分布であり、ピクセルの多様性をモデル化する手法である。

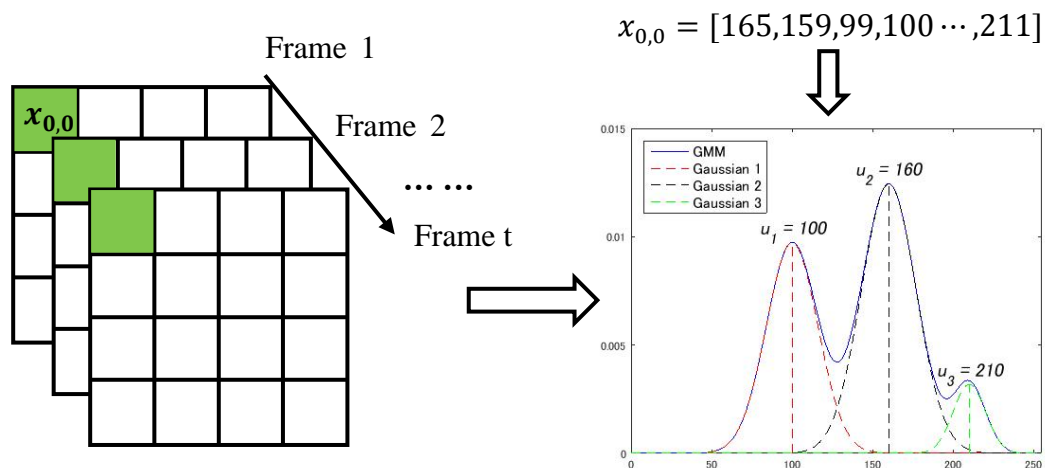
ある座標位置のピクセル x が t 時刻に持つピクセル値 x_t とすると、 x_t の出現確率 $P(x_t)$ を式(1)で推定する。

$$P(x_t) = \sum_{i=1}^K \omega_{i,t} \eta(x_t, u_{i,t}, \sigma_{i,t}) \quad (1)$$

ここで

$$\eta(x_t, u_{i,t}, \sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\sigma_{i,t}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - u_{i,t})^T \sigma_{i,t}^{-1} (x_t - u_{i,t})} \quad (2)$$

であり、 K はガウス分布の数で一般的には3~5とする、 $\omega_{i,t}$ は i^{th} ガウス分布の重みである。 d は次元数であり、グレースケール画像なら1とし、RGB画像なら3とする、 $u_{i,t}$ と $\sigma_{i,t}$ は t 番目のガウス分布の平均値と分散値をそれぞれ表す。



フレームの時間経過に伴う、座標(0,0)のピクセル $\mathbf{x}_{0,0}$ の値の混合ガウス分布によるモデルは、図1の右側の図のように $\mathbf{x}_{0,0}$ の値の頻度分布を3つのガウス分布で表現する。画像のサイズが $m \times n$ であれば、その画像中のすべての($m \times n$ 個の)座標位置に対して右側のモデルができる。

図1 混合ガウス分布によるモデルの構築

初めてモデルを構築する時、ピクセル値を初期値として u_i に与える。 $\omega_{i,t}$, $\sigma_{i,t}$ を既に設定した初期値とし、一つのガウス分布を作る。

2.2.2 モデルの更新

モデルの更新は、フレームをキャプチャする毎に行う。まず、新たにキャプ

チャしたフレーム内のピクセル値 x_{t+1} に対して、これまでに生成されている各分布の平均値から標準偏差 2.5 倍以内の分布を探す。当てはまる分布が背景モデルであれば、そのピクセルは背景とし、そうでなければ x_{t+1} を移動物体とする。

パラメータの更新は、一般的には EM アルゴリズムでパラメータを推定することで行う。ここで、EM アルゴリズムは、入力データの所属を推定する E-step(Expectation)と E-step で推定した所属の尤度を最大化するためにパラメータを調整する M-step(Maximization)とからなり、E-step, M-step を交互に繰り返すことでパラメータを最適化するアルゴリズムである。しかし、EM アルゴリズムは、計算量が大きい弱点があることから、代わりに、データ間の距離を図って距離によりパラメータを更新するオンライン K 平均法(K-means online)を採用した[3]。

パラメータの更新は以下の式のように行う：

$$u_{i,t+1} = (1 - \rho)u_{i,t} + \rho u_{i,t} \quad (3)$$

$$\Sigma_{i,t+1} = (1 - \rho)\sigma_{i,t} + \rho(x_{t+1} - u_{i,t+1})^T(x_{t+1} - u_{i,t+1}) \quad (4)$$

$$\omega_{i,t+1} = (1 - a)\omega_{i,t} + \omega_{i,t}M \quad (5)$$

ここで、

$$\rho = a * \eta(x_t, u_{i,t}, \sigma_{i,t}) \quad (6)$$

a は学習率であり、当てはまる分布に対して、 $M = 1$ 、他の分布は $M = 0$ とする。

当てはまる分布がないときは重み $\omega_{i,t}$ と分散 $\sigma_{i,t}$ の割合 $\omega_{i,t}/\sigma_{i,t}$ の値が最も小さい分布を削除して、ピクセル値 x_{t+1} を平均値として、小さい重みを付けて新たなガウス分布に置き換える。

2.2.3 モデルによる背景/移動物体の判別

1つのガウス分布を背景また前景に判定する基準を式(7)に示す:

$$B = \operatorname{argmin}_b \left(\sum_j^b w_{i,t} > T \right) \quad (7)$$

混合ガウス分布内部には、各分布を $\omega_{i,t}/\sigma_{i,t}$ の値により降順に並べ替えた後、重みの和が閾値 T を満たす最初 b 個の分布を背景とみなし、残った分布を前景とする。

3 関連研究

上記 Stauffer らの手法[3]はガウス分布の数が固定となっているが, Zivkovic らはピクセルの変化によって分布数可変とする改良法を提案している[5]. これによってピクセルの変化に柔軟に対応できるようにしている.

3.1 他の背景モデルにより移動物体の検出

背景の多様性をモデリングするために GMM 手法の他に, コードブック (codebook)手法も提案された. このアルゴリズムは, 画像内の各ピクセルについてコードブックを作成し, 各コードブックは, 学習のための最大および最小閾値検出に用いる最大および最小閾値などに対応する複数のシンボルを含むことができる. 背景のモデリング中, 新しいフレームが入力される毎に, 各ピクセルのコードブックが更新される, すなわち, ピクセル値がコードブックのシンボルの学習閾値内にある場合, それは過去のものと同様にみなされる特定のピクセル値との比較を通じて, 条件が満たされた場合に, 学習閾値および検出閾値に対応するシンボルを更新することができる. 新しいピクセル値がコードブックの各シンボルと一致しない場合は, 動的な背景である可能性があるため, 新しいシンボルを作成し, 対応するシンボルの変数を設定する必要があるものの, 背景モデルを訓練する過程では, 各ピクセルが複数のシンボルに対応できるため, 複雑な動的背景を学習できる利点がある.

3.2 局所的な特徴量を考慮した従来法

ピクセルの画素値により背景をモデリングする手法に対抗して, LBP 法が提案されている. LBP 法はピクセルと隣接領域の大小関係を算出してテクスチャ

をバイナリパターンに変換して、局所的な特徴量を抽出する手法である。

また、SeungJong Noh らは LBP に基づいて画像の局所的な特徴とピクセルの両方をコードブック (code-book) に格納してモデリングすることで、ピクセルのモデルにより、テクスチャのモデルを最適化する手法を提案している [6]。

3.3 従来法の問題点

3.3.1 単一の座標位置のピクセル値分布を生成するに對しての手法の問題点

一般的に混合ガウス分布は単一の座標位置のピクセル値の時間領域の変化だけでモデリングするために、モデルが画像のノイズに影響されやすい。具体的には 2 節で説明したようなガウス分布の数を K 個に限っていることにより背景変化が限度を超えると対応できなくなる。

3.3.2 隣接領域を考慮した手法の問題点

空間特徴に基づく従来法では、移動物体と背景で色とテクスチャが類似している場合に、移動物体を背景に、あるいは背景を移動物体に判定してしまう結果、移動物体を正しく判別できないことがある。

図 2 では、歩行者の衣服の色が木の葉、空の色に似ているので、背景に判定してしまう。



(a)



(b)



(c)

図 2 前景, 背景に生じる誤判定

(a)は移動物体がないシーンである. (b)は歩行者が入るシーンである. (c)は空間特徴に基づく従来法の出力

4 提案手法

4.1 改善の対象

ピクセルレベルで移動物体を高精度で検出するには以下の三点が必要である.

- 移動物体の輪郭を正しく検出する.
- 背景の多様性に対応し, 誤検出しない
- 画像のノイズに対して強い

そこで, 本研究は空間的な情報を利用することによってピクセルの局所的な特徴を捉えられるようにする混合ガウス分布モデル(GMM)に基づくモデルの構築手法を提案する.

画像の局所的な特徴量の抽出のために, 本研究では, 移動平均フィルタと平均プーリング(mean pooling)また最大プーリング(maximum pooling)法を採用する. フィルタリングとプーリングの出力画像と入力画像について, 混合ガウス分布により背景モデルをそれぞれ作る. 次いで, これら 3 つの背景モデルを組み合わせて, 各ピクセルが移動物体か, または背景かを判断する.

一般的に, 入力画像の任意の隣接ピクセルは類似性が高く, 近傍ピクセル間には強い相関性がある. 本研究はこの性質を利用してフィルタで抽出する特徴量を背景モデルの要素とする.

4.2 基本的なアイデア

4.2.1 フィルタにより画像処理

空間フィルタリングは, 画像中の局所の特徴量抽出をはじめとして広く利用されており [6][8], 背景モデルの生成においても, 画像ノイズの影響低減に役立

つ.

そこで、本研究では図 2(a)に示す移動平均フィルタを利用する。このフィルタ処理は、式(8)によって与えられる。

$$C = \frac{\sum_i^n \sum_j^m x_{i,j}}{n*m} \quad (8)$$

ここで、 $x_{i,j}$ は入力画像のピクセル、移動平均フィルタのサイズは $n \times m$ 、 C はフィルタの出力を表す。ここでは、畳み込んだ結果を入力画像のサイズに一致させるため、入力画像の外側にはみ出す画素を参照する場合は 0 で補填する。

4.2.2 プーリングにより画像処理

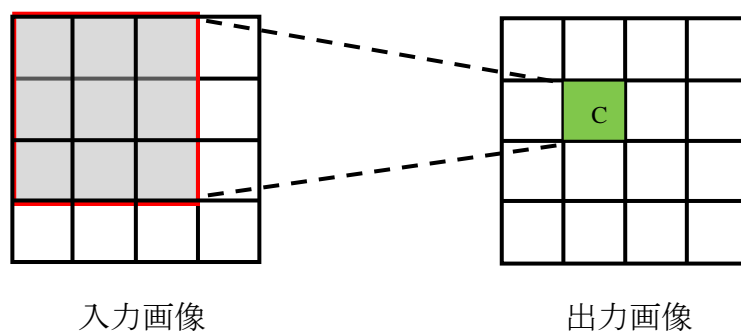
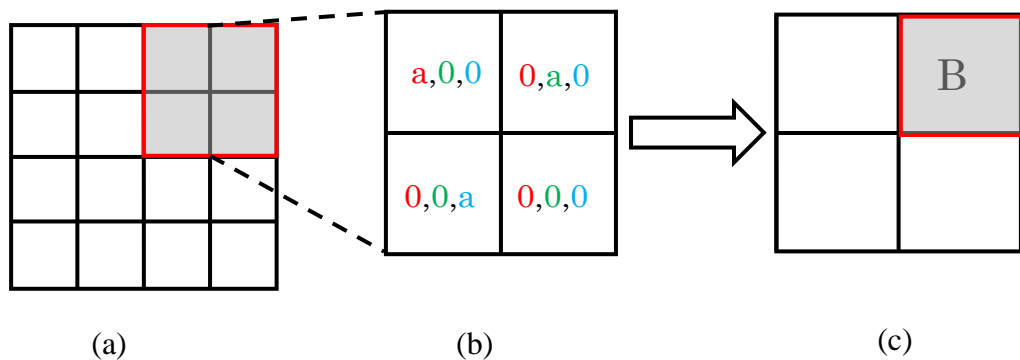


図 3 移動平均フィルタによる画像処理

入力画像を低解像度化 (Down Sampling) するプーリング処理は、特徴量の位置変化によるモデルへの影響の低減に役に立つ[8]。図 3 はそれぞれ平均値プーリング、最大値プーリングを示している、まず入力画像を互いに重なり合わない複数のブロックに分割する。そして、平均値プーリングでは、各ブロック内の全てのピクセルに対して、R、G、B 各分量の平均値を算出する。また、最大値プーリングまた最小値プーリングでは、各ブロック内の最大/最小のピクセル値を取り出す。RGB 画像ならば R、G、B それぞれの最大値/最小値を取り出す(max-pooling/min-pooling)。これらのプーリング処理により、縮小画像が

生成される.

そのうち、普通では、サイズが $n \times m$ の入力画像に対して $n \times m$ 個の混合ガウス分布があるが、プーリング背景モデルは縮小画像により生成したモデルのため $(n/h \times m/w)$ 個の混合ガウス分布がある,ここで、 h,w はプーリングの高さと幅), 入力画像のピクセル $x_{a,b}$ (a,b は行と列の番号)が対応する混合ガウス分布の位置は $G_{a/h,b/w}$ となる.



$$B_{max} = [R = a, G = a, B = a]$$

$$B_{min} = [R = 0, G = 0, B = 0]$$

$$B_{mean} = [R = a/4, G = a/4, B = a/4]$$

(a)は入力画像, (b)は分割したブロック, (c)はプーリング処理で生成した低解像度画像. 下に示す公式はそれぞれ最大値, 最小値, 平均値プーリングの内容を示す.

図 4 プーリング処理

4.3 背景/移動物体の判定

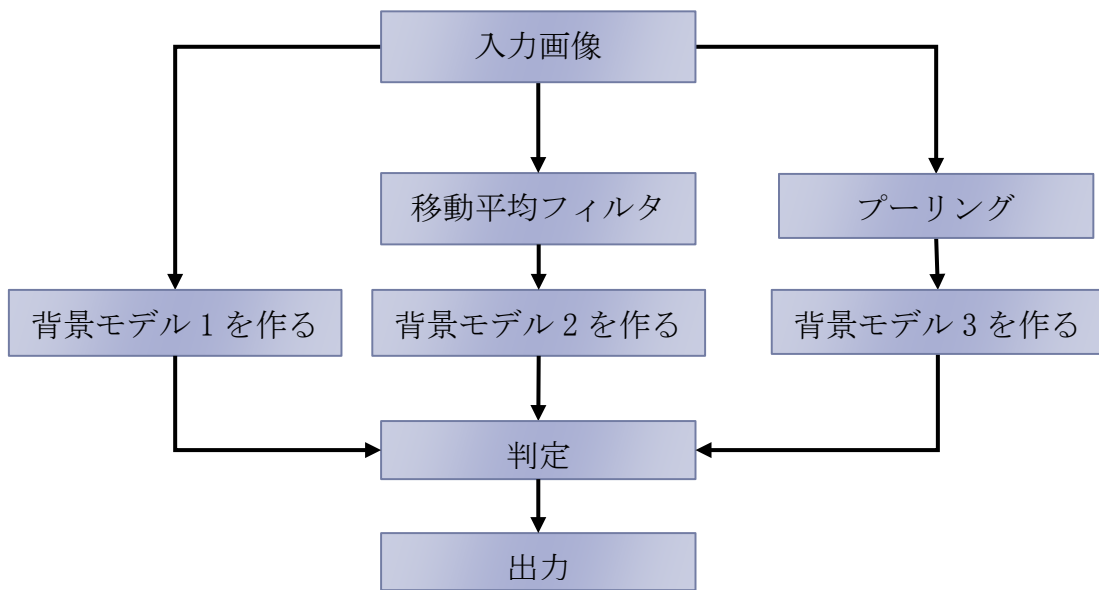


図 5 提案手法の流れ

上の図が示しているように、システムは入力画像とそれを上記 4.2 節で紹介した手法で処理して得る平滑化画像と縮小画像の 3 つの画像を用いて構築する 3 つのモデルによって、入力画像のピクセルを振り分ける。具体的には、複数のモデルで同時に背景(移動物体)と判定されれば、背景(移動物体)ピクセルとする。複数のモデルの間の判定結果に違いがある場合は、このピクセルを原点とし、一定の範囲内で、画素値が類似しているピクセルを探索する。この探索により見つかったピクセルで背景と判定される方が多い場合、このピクセルを背景とし、逆に移動物体と判定される方が多い場合、移動物体とする。また、両方が等しい場合にはモデルの出力の間の多数決で決める。この判定処理のプログラムコードを表 1 に示す。

表 1 判定アルゴリズム

Algorithm Decision Pseudo-code

```

1. function decision(in Frame, in vecModel, out Map)
2.   for all pixels in Frame do
3.     for i=0 to numModel do
4.       if Background == vecModel[i].at(pixelLocation) then
5.         cnt++
6.       end if
7.     end for
8.     if numModel == cnt then
9.       Map(pixelLocation) = Background
10.    elseif 0 == cnt
11.      Map(pixelLocation) = Foreground
12.    else
13.      Map(pixelLocation) = undefined
14.    end if
15.  end for
16. for all pixels in Frame do
17.   if undefined == Map(pixelLocation) then
18.     for all neighbourPixels in neighbor(pixelLocation)
19.       if diff(pixel, neighbourPixel) < T then
20.         if Background == neighbourPixel then
21.           bcnt++
22.         elseif Foreground == neighbourPixel
23.           fcnt++
24.         end if
25.       end if
26.     end for
27.     if bcnt > fcnt then
28.       Map(pixelLocation) = Background
29.     elseif fcnt > bcnt
30.       Map(pixelLocation) = Foreground
31.     else
32.       if cnt > numModel/2 then

```

```
33.         Map(pixelLocation) = Background
34.         else
35.         Map(pixelLocation) = Foreground
36.         end if
37.     end if
38. end if
39. end for
40. end function
```

5 性能評価

5.1 評価環境

提案手法の有効性を示すため、テストビデオ **WavingTree**, **Fountain**, **Traffic** を評価用データセットとして利用した。

WaveingTree: 木の揺らぎを含め、歩行者がカメラの前を通る映像である

Fountain: 噴水を含め、車が噴水の後ろを通る映像である

Traffic: 激しく振動する、カメラの前を車を通る映像である

それらのテストビデオ中で歩行者や運転中の車などの移動物体を前景として、抽出したい領域とする。

提案手法と従来法の出力画像はバイナリ画像であり、白い部分が移動物体の領域を示し、黒い部分が背景であることを示す。また、移動物体を検出する精度を比較するため、出力画像と評価用データセットの解である **Ground Truth** を比較し、式(9)~(11)が示すように適合率(Precision Rate), 再現率(Recall Rate)と F-measure について評価する[9].

$$\text{Precision Rate: } p = \frac{T_p}{T_p + F_p} \quad (9)$$

$$\text{Recall Rate: } r = \frac{T_p}{T_p + F_n} \quad (10)$$

$$\text{F-measure} = \frac{2 * p * r}{p + r} \quad (11)$$

ここで、 T_p とは正確に抽出した前景ピクセルの数、 F_p とは背景ピクセルを前景として抽出されたピクセルの数、 F_n とは前景ピクセルを背景として抽出されたピクセルの数である。

比較対象として3節で述べた文献[5]の改良版 **GMM** 手法と文献[6]の手法を従来法 1 と 2 として採用した。今回の実験では、次節の予備試験結果に基づき、

平均値プーリングを採用した。提案手法のパラメータは文献[5]と同じく学習率 $a = 0.05$, 閾値 $T = 0.9$, ガウス分布の数の上限を $K = 5$ に設定した。実験環境は Intel Core i7-2700k, 16GB メモリ, OS は Windows 10, 開発環境は Visual Studio 2015, C++, また, 画像ファイルの読み込み, 書き込みには, OpenCV ライブラリを使用した[12].

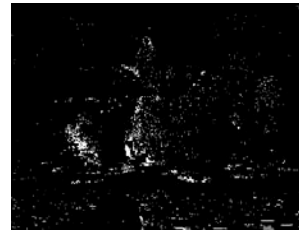
5.2 予備試験

本節では, 節 4.2.2 で紹介した 3 種のプーリングについて議論する。

(a) は入力画像であり, このフレーム, 噴水の量が突然大きくなり, 虹も出現してしまっている。(b) は入力画像のモデルの出力であり, 明らかに誤って検出されたピクセルが多い。(c) は移動平均フィルタで処理された画像のモデルの出力であり, (d), (e), (f) はそれぞれ平均値プーリング, 最大値プーリング, 最小値プーリングで処理された画像により構築したモデルの出力である。(b) より, (c)~(f) の方が, 騒音が少ないのに加え, 突然変化している背景部分が鮮明に見える。また, 明らかに, (b) と (c) より, プーリングモデルの出力 (d), (e), (f) のほうが誤検出少ない。ただし, `down-sampling` された画像から元サイズへ復元することにより, 復元された画像には塊状のノイズが鮮明に見える。この中で最大値, 最小値より平均値プーリングは誤検出がもっと少ないことが分かる。これを考慮して本研究の実験では, 平均値プーリングを採用することとした。



(a)



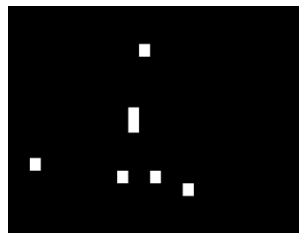
(b)



(c)



(d)



(e)



(f)

(a): 入力画像, (b): 入力画像による構築したモデルの出力, (c): 平滑化された画像による構築したモデルの出力, (d): 平均値プーリングによる構築したモデルの出力, (e): 最大値プーリングによる構築したモデルの出力, (f): 最小値プーリングによる構築したモデルの出力,

図 6 プーリング手法の比較

5.3 評価結果

表 2 と 3 は、節 5.1 が述べた各シーケンスに対して、提案手法と従来法で処理した結果を表している。ここで、システムは 3 つのモデル(入力画像, 平均フィルタ, 平均値プーリング)から成る。

WaveingTree に対して、従来法 1 の結果には、木の揺らぎが、まだはっきり残ってしまっている上に、抽出した部分の中に空洞が明らかに多いことがわかる。それに対して、従来法 2 の結果は木の揺らぎの影響が少ないが、数多くの移動物体のピクセルが誤って背景に分類されてしまっている。提案手法の結果は木の揺らぎの影響がほとんどなく移動物体を正確に検出できている。

Fountain に対して、従来法 1 の結果には、噴水の影響がまだ残っている、他にも、抽出した部分に空洞もある。従来法 2 の結果には噴水の影響がほとんどないが、移動物体の面積は **GroundTruth** と比べ膨らんでしまっている。提案手法の結果は噴水の影響が少ないものの、抽出した移動物体に空洞があり、車両のエッジもやや鮮明さに欠ける。また、表 1 に示すように、提案手法の **Recall rate** は従来法より低い。この原因は、移動物体として抽出されたピクセルの数が **GroundTruth** より少ないことにある。

Traffic に対して、従来法 1 の結果には、車道線の影響がまだ残っている、他にも、抽出した部分に空洞もある。従来法 2 の結果には車道線の影響がほとんどないが、抽出された領域に大きい空洞がある。提案手法の結果は車道線の影響が少なく移動物体を正確に検出できている。

表 2 評価指標の比較













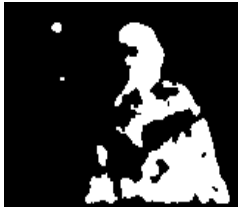


従来法 1:文献[5] 従来法 2:文献[6]

WavingTree	Recall	Precision	F-measure
提案手法	98.77%	98.46%	98.07%
従来法1	60.41%	90.17%	72.35%
従来法2	95.15%	85.07%	89.82%

Fountain	Recall	Precision	F-measure
提案手法	88.83%	96.99%	92.73%
従来法1	97.93%	81.25%	88.81%
従来法2	90.19%	91.47%	90.82%

Traffic	Recall	Precision	F-measure
提案手法	97.01%	96.53%	96.77%
従来法1	81.37%	92.33%	86.50%
従来法2	84.07%	90.72%	87.27%

表 3 処理結果との比較

	従来法 1:文献[5]	従来法 2:文献[6]	
	WavingTree	Fountain	Traffic
入力画像			
Ground Truth			
提案手法			
従来法 1			
従来法 2			

5.3.1 探索範囲について

下の図は F-measure でプーリングのサイズと判定アルゴリズムの探索範囲について議論する。

3つのテストビデオ共に、探索範囲の拡大により、F-measure が増加する傾向があること分かる。また、Waving Tree と Traffic については探索範囲が小さい時、異なるプーリングのサイズのシステムの結果の間に大きい差があるが(最大 5% ぐらい)、探索範囲の増大に伴い、その差が小さくなることが分かる(1% ぐらい)。Fountain については、探索範囲を拡大しても F-measure はあんまり収束してない。この原因は、移動物体が小さくて、移動物体内部にも違う部分が多い、判定が複雑になっていることにある。

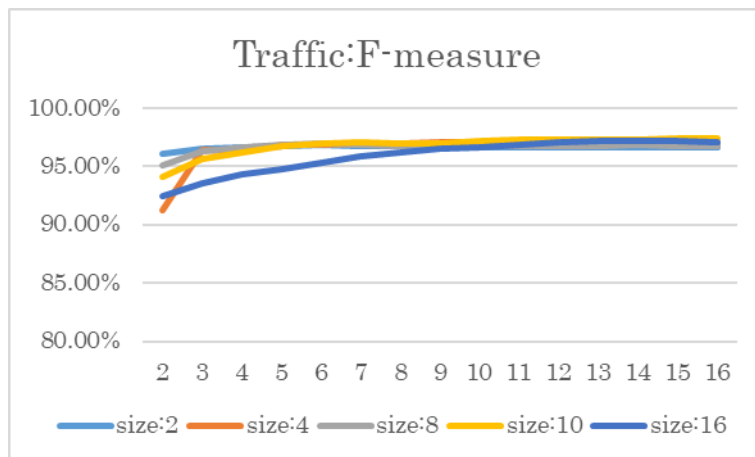
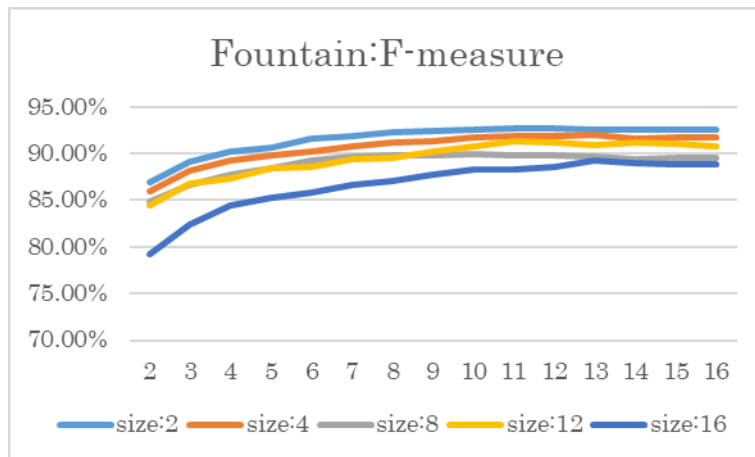
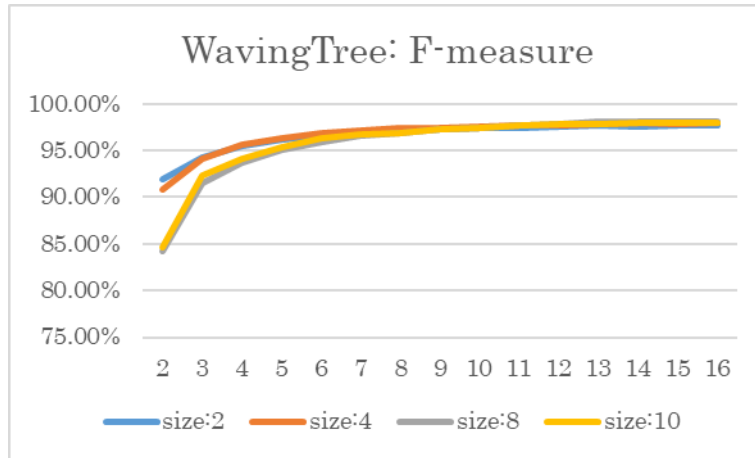


図 7 探索範囲の変化より結果への影響

5.3.2 モデルの数について

ここで、システムには5つのモデルから構成する時の結果について検討する。異なるサイズの移動平均フィルタ 2 つ，最大値と平均値プーリングそれぞれ 1 つと入力画像により合計 5 つのモデルから成る。

実験中，探索範囲とプーリングサイズの変化に伴ってシステムの出力は大きく変動していることが分かる。下の表が示すように，F-measure から見れば，最良と最悪の結果の差が大きい。Fountain においては，最大 25% ぐらいの違いがある一方，モデルの数は 3 つの時，この差は 12% ぐらい，より不安定であることが分かる。

その原因はプーリング処理の出力が塊状であり，プーリング処理のモデルを増やすと，抽出部分の輪郭に影響を与えらる。

表 4 5 つのモデルの性能評価

	Max F-measure	Min F-measure
Waving Tree	97.81%	82.22%
Fountain	90.99%	65.81%
Traffic	97.47%	89.71%

6 あとがき

本論文は、固定されたカメラが撮った画像に向けて、従来の移動物体の検出とそれに用いられる背景モデリングの問題を、局所的な空間特徴を利用した複数の背景モデルを用いて判定することにより、ピクセルレベルで高精度検出する手法を提案した。この提案手法は、画像の平滑化とプーリング処理を施した画像の背景モデルを併用し、最後に、判定アルゴリズムにより最終の結果を出力する。入力画像をそのままモデリングする場合のノイズに対する弱さ、空間特徴をとってからモデリングする場合の抽出された物体中の空洞多発などの問題点を解消できることを確認した。また、性能評価より、提案手法、GMM, LBP-CodeBookに基づく従来法と比較して、3種類の動画像を評価し、全体的に、提案手法の方が F-measure が高いことを明らかにした。しかし、Fountain シーケンスの結果において、Recall Rate が悪化していることから、移動物体を正確な検出ができていないこともある。

今後の課題としては、プーリング処理で画像の解像度が低下することにより抽出領域のエッジや輪郭などの情報を失われる問題をいかに軽減するかが挙げられる。入力画像を異なるサイズのプーリング処理した結果を組み合わせ feature-map を用意し、up-sampling 層を作って統計的な手法(Conditional Random Field など)を使って feature-map からエッジ情報を復元することが可能になると考えられる。また、これらの性能改善と共に、GPU, マルチスレッドなどの高速化手法を加え、実用性の高い高精度検出を確立して行く必要がある。

お謝辞

本論文の執筆にあたり、日頃からご指導、ご助言いただきました近藤

利夫教授，佐々木敬泰助教，深澤祐樹研究員に感謝いたします。また様々な面でお世話になったコンピュータアーキテクチャ研究室の同輩方々に感謝の意を表します。

参考文献

- [1] Wren, Christopher Richard, et al. "Pfinder: Real-time tracking of the human body." *IEEE Transactions on pattern analysis and machine intelligence* 19.7: 780-785, 1997.
- [2] Brutzer, Sebastian, Benjamin Höferlin, and Gunther Heidemann. "Evaluation of background subtraction techniques for video surveillance." *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.
- [3] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.
- [4] Heikkila M, Pietikainen M, Heikkila J, et al. A Texture-based Method for Detecting Moving Objects[C]. *british machine vision conference*, 2004.
- [5] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." *Pattern Recognition*, 2004. *ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 2. IEEE, 2004.
- [6] Noh, SeungJong, and Moongu Jeon. "A new framework for background subtraction using multiple cues." *Asian Conference on Computer Vision*. Springer Berlin Heidelberg, 2012.
- [7] Kim, Kyungnam, et al. "Real-time foreground-background segmentation using codebook model." *Real-time imaging* 11.3 : 172-185, 2005.
- [8] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11: 2278-2324, 1998.
- [9] Ohba, Akio. "Video image processing apparatus including convolution

filter means to process pixels of a video image by a set of parameter coefficients." U.S. Patent No. 5,241,372. 31 Aug. 1993.

[10]Sobral, Andrews, and Antoine Vacavant. "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos." *Computer Vision and Image Understanding* 122: 4-21, 2014.

[11]Piccardi, Massimo. "Background subtraction techniques: a review." *Systems, man and cybernetics, 2004 IEEE international conference on*. Vol. 4. IEEE, 2004.

[12]Bradski, Gary R, and Adrian Kaehler. " *Learning OpenCV: Computer vision with the OpenCV library*." Sebastopol, CA: O'Reilly, 2008. Print.