

オプティカルフローを用いた画像中の野鳥検出

Bird Detection in Images with Optical Flow

久保山 裕† 三田 長久† 吉岡 俊英†
Yutaka Kuboyama Nagahisa Mita Toshihide Yoshioka

1. まえがき

近年、急速な技術革新に伴い、環境問題が深刻化している。そこで、我々は、自然環境の評価を目的として、環境の変化に敏感な野鳥を対象とした調査を行うことを考えている。今回は、この調査を行う際の科学技術の応用として、画像中の野鳥のカウントと種類識別を行う手法について提案する。

本研究では、動画画像中の動きをベクトルで表示するオプティカルフロー^[1]を用いて野鳥の検出を行う。ただし、フローの検出にはブロックマッチング法を用いている。まず、ブロックマッチング法のブロックごとに画素値の標準偏差を求めて、この値が閾値以上の複雑なブロックのみを対象に処理を進めていく。次に、検出されたフローの向きに条件を設定して、連結ブロックごとにラベリングを行う。最後に、動物体が検出された領域に対して識別を行い、野鳥の種類や背景を判別する。

第2章では、動物体検出として、まず、オプティカルフローについて説明し、更に、画像の複雑度を用いたフロー検出処理の改善手法や、フローの情報を用いて動きが検出された領域をクラスタリングする手法について述べる。第3章では、野鳥を識別する手法について説明し、第4章では、今回の提案手法を用いて行ったシミュレーションの結果について考察する。第5章では、以上を総括した結論を述べる。

2. 動物体検出

2.1. オプティカルフロー

オプティカルフローとは、動画画像中の物体の動きを検出して、速度をベクトルで表示する手法で、フローの推定法としては、勾配法とブロックマッチング法が提案されている。勾配法は、短い処理時間で検出が可能だが、検出の精度が低いという欠点がある。一方、今回使用するブロックマッチング法は、一般的に長い処理時間が必要になるが、濃度値の変化や雑音に強く、高い精度で検出を行うことができる。そこで、今回は、欠点である処理時間の改善を考慮しながら、ブロックマッチング法によってフローの検出を行う。

ブロックマッチング法は、画像をある大きさのブロックで分割して、次のフレームの画像中から注目ブロックとの類似度が最も大きいブロックを検出する手法である。今回は、物体の大きさが異なる画像でも良い結果が得られる値を実験的に決定し、ブロックの大きさを5×5画素、検索範囲を注目ブロックの座標から5画素以内としている。また、類似度の評価関数には、次式を用いている。

$$e_{x,y}(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |Y_t(x+k,y+l) - Y_{t+1}(x+i+k,y+j+l)| \quad (1)$$

ただし、 (x,y) は注目ブロックの座標、 (i,j) は検索ブロックの座標、 t はフレーム番号、 N はブロックの一边の画素数を示しており、グレイスケールの濃度値 Y を用いて評価を行う^[2]。

2.2. 画像の複雑度

ブロックマッチング法において、画像をブロックごとに分割した際に、ブロックごとに画像の複雑度を計算して、その値が閾値以上のブロックのみを対象にすることで、処理時間の削減を図る。画像の複雑度が低いブロックでフローを検出しても、その信頼度は低く、誤検出であることが多い。また、対象とする物体の周囲では、物体の内部や背景との境界で変化が生じるので、複雑度の閾値で排除される可能性は低い。したがって、処理の高速化に加えて、検出精度の向上も実現できる。ただし、画像の複雑度には、次式のような濃度値の標準偏差を用いている。

$$c_{x,y} = \sqrt{\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} (Y(x+k,y+l) - Y_m)^2} \quad (2)$$

Y_m はブロック内の画素の濃度値の平均である。

画像によって多少の変動はあるが、CPU: Pentium4 3.2GHz, メモリ: 1024MB, OS: WindowsXP の計算機で、Visual C#を用いてシミュレーションを行った結果、フローの算出にかかる時間は、320×240画素の画像で230msから31msに削減された。シミュレーションに用いた連続するフレームの画像を図1に、実行結果を図2に示す。

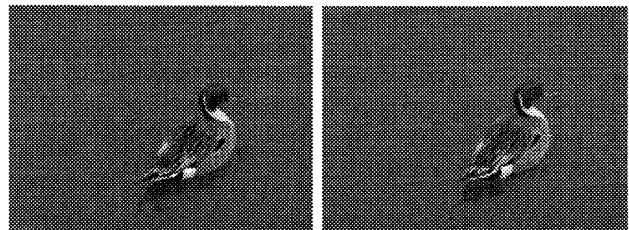


図1 原画像

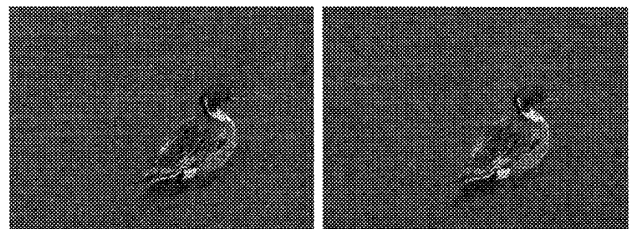


図2 画像の複雑度によるフローの削除
(左: 閾値なし, 右: 閾値 20)

†熊本大学大学院自然科学研究科

2. 3. クラスタリング

オブティカルフローを検出した後は、フローが検出されたブロックを連結させてクラスタリングを行い、クラスタの大きさが閾値以上の領域のみを対象にして以降の処理を行う。ただし、連結させる際の条件としては、フローが0でないという条件以外に、フローの向きがそのクラスタの移動方向から見て閾値以内であるかという条件も付加している。基準となるクラスタの移動方向は、そのブロックが判定されるまでにクラスタリングされたブロックのオブティカルフローの向きの平均を用いている。その結果、野鳥が重なって写っている画像に対して、2羽の動く向きに差がある場合、分離して検出することができる。実行結果を図3に示す。

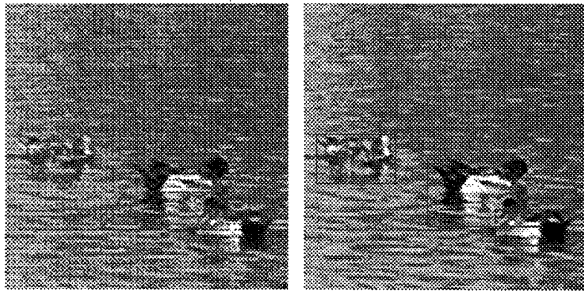


図3 重なった野鳥の分離

また、水面などのランダムに変化する領域に対して、クラスタを細かく分離させることで、クラスタの大きさの閾値で排除できるという利点もある。フローの向きの閾値を変化させたときの様子を図4に示す。

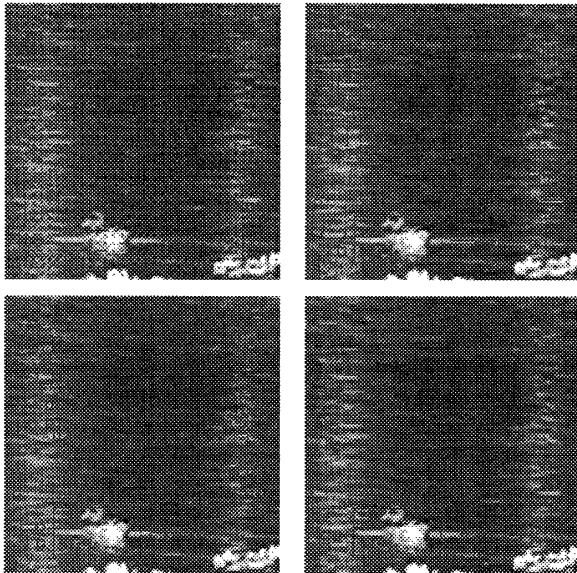


図4 背景領域の分離 (右上: 閾値なし, 左下: 閾値 30° , 右下: 200画素以下の領域を除去)

3. 野鳥の識別

最後に、検出された動物体の識別を行う。この段階まで背景が残っていたとしても、対象とする野鳥のみを識別することで、背景領域は除去される。

今回は対象が野鳥ということで、色情報が重要になるため、1画素につき3次元の特徴を用いる。ただし、次元数削減のため、動物体として検出された領域を 10×10 画素に規格化して、それぞれの画素の R, G, B の値で 300次元の特徴ベクトルを作成する。更に、主成分分析を用いて、累積寄与率が 80%までの主成分を用いた特徴ベクトルを作成し、SVMによって識別を行う。

4. シミュレーション

今回の手法を用いて検出を行った結果を以下に示す。ただし、複雑度の閾値は 20、フローの向きの閾値は 30° 、クラスタの連結画素数の閾値は 200画素としている。また、識別については、今回の対象としたヒドリガモの画像と背景画像をそれぞれ 100パターンずつ用意し、主成分分析で 20次元の特徴ベクトルを作成して、識別を行った。実行結果を図5に示す。

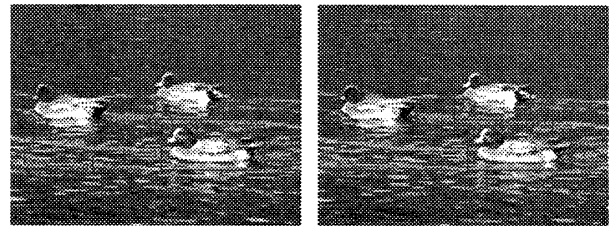


図5 野鳥の検出 (左: 識別前, 右: 識別後)

動物体検出で残っていた背景領域が、識別によって除去されているのがわかる。また、今回の手法を用いることで、野鳥に対して、任意の大きさの矩形で検出することが可能である。

5. 結論

今回は、オブティカルフローを用いた野鳥の検出手法をテーマに、ブロックマッチング法の処理時間を削減する手法や、背景領域の誤検出を減らす手法について提案した。

今後の課題としては、野鳥が重なって写っている画像に対して、今回の手法では分離するのが困難な場合の対策や、リアルタイム処理に向けて、システム全体の処理速度の向上、適切な閾値の決定方法などが挙げられる。また、野鳥の種類識別についてもシミュレーションを行っていく。更に、音声情報のスペクトログラムを画像として扱い、音声による野鳥の種類識別方法への適用も検討する。

本研究は、一部、環境技術開発等推進費の補助を受けて実施した。

参考文献

- [1] H.J. Chen, Y. Shirai and M. Asada, "Detecting multiple rigid image motions from an optical flow field obtained with multi-scale filters", IEICE Trans. Inf. & Syst. Vol. E76-D, No. 10, pp. 1253-1262, 1993.
- [2] M.Watanabe, N.Takeda and K.Onoguchi, "Moving obstacle detection and recognition by optical flow pattern analysis for mobile robots", Advanced Robotics, Vol.12, No.7, 8, pp.791-896, 1999.