

害獣捕獲罠における動物自動計数システムの開発

生産システム工学専攻 山本 陽祐, 指導教員 江崎 修央

Development of Counting System of Harmful Animals in traps

Yosuke YAMAMOTO, Nobuo EZAKI

Advanced Course of Production System Engineering, National Institute of Technology, Toba College

Abstract In recent years, damage to agricultural products by wild animals has become serious. Installing traps is the main method of capturing animals, but humans must continually monitor the state of the trap to ensure they have been captured for certain which takes great labor. There is a system which can monitor and capture animals by remote control, however it is not realistic to continually monitor the state of the trap by mobile data terminal. Therefore, an automatic capturing system, which can count animals in a trap, has been developed. Also an evaluation experiment was conducted about the actual number of animals and the number counted automatically by using an image photographed by a capture trap.

Keyword Vermin Detection, Image Processing, Kalman Filter, Template Matching

1. はじめに

全国各地の集落で害獣による農作物への被害が深刻となっており、国内の害獣による農作物被害額は 191 億円にも及んでいる [1]。その対策として捕獲檻による害獣の捕獲が主流となっているが、常に檻の近くで監視し続ける必要があり多大な労力がかかる。近年、「まる三重ホカクン」 [2] など遠くから害獣を監視し、捕獲するシステムが登場している。このシステムはスマートフォンやパソコンで檻の様子を確認でき、遠隔操作で罠を作動させる。

しかしながら、檻近辺に害獣が存在したとしてもしばらくは檻内に入らないた

め、スマートフォン等で常に確認しながら捕獲タイミングを待つことも中々難しい。そこで本研究では檻周辺と檻内にいる動物数を自動でカウントするシステムを開発する。もしも動物が一定数以上入っていれば猟師にメールを送信し捕獲を促したり、日付・時間と共に動物数をクラウド上のデータベースへ保存することで、WEB ページからいつでも動物数と日時を見たりすることができる。また出没しやすい時間帯の傾向が分かり、罠を作動させる日時の予定が立てやすくなる。最終的には自動での罠作動による自動捕獲を目指していく。

本稿の先行研究として，スマートセンサーを用いて害獣の頭数カウントや獣種判別を行うシステムの開発 [3] や深度情報を利用して害獣の頭数をカウントするシステムの開発 [4] 等がある．前者の研究ではスマートセンサーと呼ばれるセンサーカメラと処理用の CPU が搭載した機器を開発し，害獣の頭数カウントと獣種判別を行っている．後者の研究では Kinect を用いて深度情報を取得し，画像処理を用いて害獣の頭数カウントを行っている．

本稿では，檻周辺の害獣の数を自動でカウントする機能の精度向上を目的とする．

2. システム概要

2.1. システムの構成

現在開発中のシステム構成を図 1 に示す．本システムはカメラで檻を撮影し，画像から動物の数をカウント，捕獲予定数の害獣が檻に入っていれば自動捕獲する．この時，檻外に害獣がいる場合は捕獲してはいけないため，檻外に害獣がいることも認識しておく必要がある．また捕獲する獣種を予め設定しておく事により檻内の獣種を自動で判別し，想定外の動物（クマ等）が入っているときの誤捕獲を防ぐことができるようにする．さらに捕獲時刻を設定しておくことにより人間が近づく恐れのある日中は動作させないなどの設定が行えるようにすることで，誤動作を防ぐ．

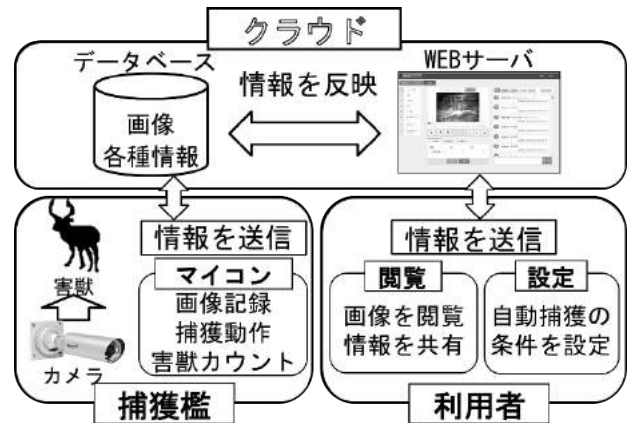


図 1 システムの構成図

これらの情報閲覧は別途開発している WEB サイトで行える．これはパソコン，スマートフォンからアクセスできる．WEB サイトはライブ映像閲覧画面（図 2 (a)）と録画画像閲覧画面（図 2 (b)）の二つから構成される．



(a) ライブ映像閲覧



(b) 録画画像閲覧

図2 WEBサイトの画面例

ライブ映像閲覧画面では捕獲檻の状態をリアルタイムで確認しチャットで情報共有できる機能，および害獣が檻内に入れば「捕獲」を実行できるボタンが用意されている。また，録画画像画面では，チャット履歴やカレンダーから任意のイベント・時刻を選択すると録画画像が表示されるので，害獣の出没グラフも参照しながら，今後の捕獲計画を立てることができる。

2.2.画像処理による害獣抽出の流れ

図3に画像処理による害獣抽出の流れを示す。まずカメラから害獣捕獲檻付近の画像を取得し，背景画像の設定を行う。続いて背景画像を用いて差分処理を行った画像に対して平滑化・二値化・収縮処理・ラベリングを行ない，動体の抽出を行う。

そして抽出した動体に対して，害獣追跡のために前回の動体領域から抽出したテンプレート群によりマッチングを行う。もしも動体にマッチングするテンプレ

トが存在しない場合は，新たな害獣が侵入したと判断し，その動体を次回のテンプレートとして保存しておく。なお，一定回数以上テンプレートマッチングで一致しなかったテンプレートは削除する。

その後マッチングした動体に対して，カルマンフィルタによる位置推定処理を行う。カルマンフィルタを利用する意義は，動体同士が重なってしまうと2頭いるにもかかわらず1頭と判定されるためである。これらの処理を抽出した動体すべてに行う。最後に推定位置をもとに動体が重なっているかを判別し，動体数をカウントする。

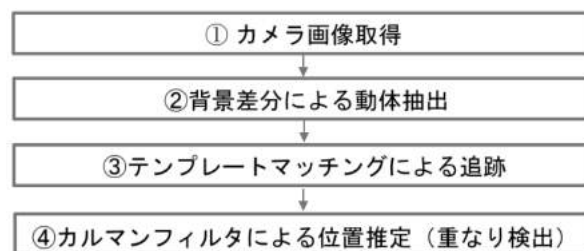


図3 画像処理による害獣抽出の流れ

3. 動体の抽出方法

3.1.背景差分による動体領域抽出

背景差分とは，あらかじめ撮影した背景画像（図4(a)）と入力画像（図4(b)）の差分をとることで，動体領域

(図 4 (c)) を抽出する処理である.

なお, 背景画像は入力画像の平均 (図 4 (d)) をとり, 動的に定期更新を行う.

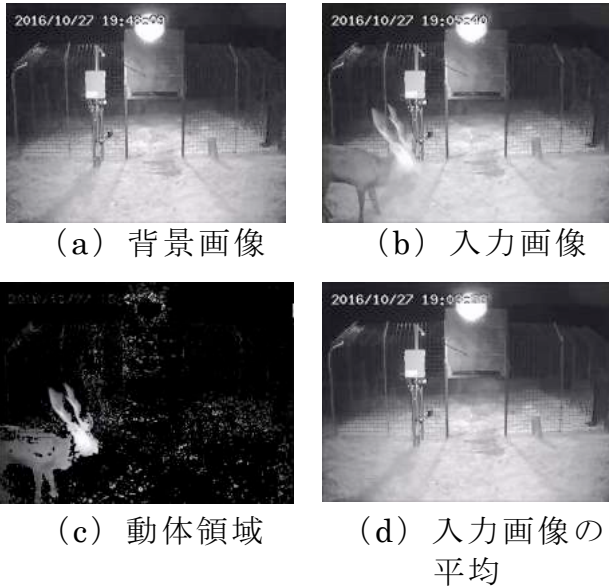


図 4 背景差分による動体領域の抽出

3.2. 平滑化によるノイズ除去

背景差分処理で得られた画像には動物以外のノイズが多く含まれる. そこで, 平滑化処理を行う. 平滑化処理とは雑音をもつ一般的性質に基づいて雑音除去を行う処理である [5]. 背景差分処理後 (図 5 (a)) はゴマ塩ノイズが多くみられるため, メディアンフィルタを用いてノイズ除去を行った. 平滑化処理後 (図 5 (b)) はゴマ塩ノイズが除去できていることがわかる.

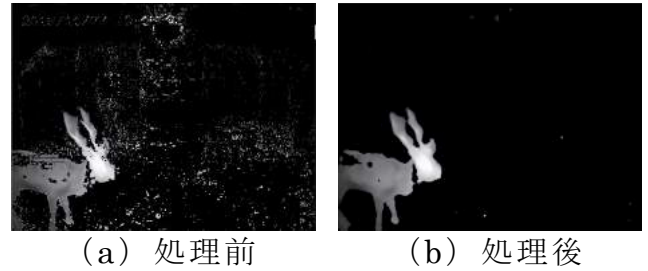


図 5 平滑化によるノイズ除去

3.3. 二値化による動体の特定

二値化処理を行うことで対象とする動物を白領域として表すことができ, 二値化処理後 (図 6 (b)) は白領域と黒領域の二つに分けられていることがわかる.

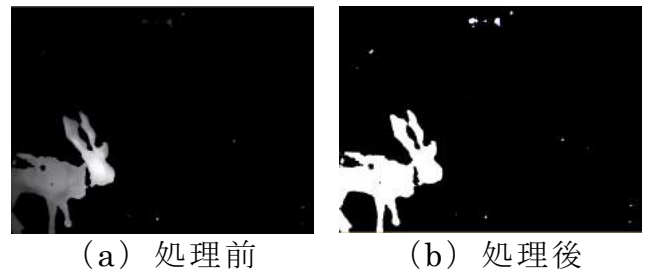


図 6 固定閾値による二値化

3.4. ラベリングによる微小領域の除去

ラベリングによりラベル付けした領域が小さい部分は明らかに動物でないと判断できるため除去する. ラベリング処理後 (図 7 (b)) は白領域がラベル付けされ, 小さい領域は除去されていることがわかる.

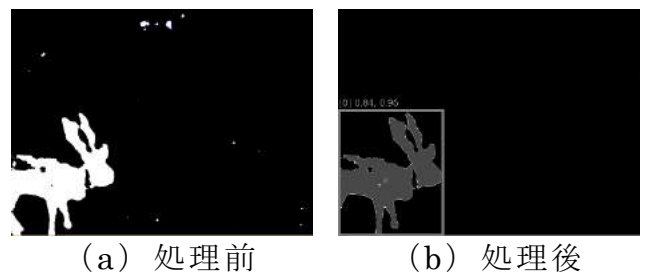


図 7 ラベリングによる微小領域除去

4. テンプレートマッチングによる追跡

4.1. テンプレートマッチングとは

テンプレートマッチングとは, 図 8 に示すように, 検出しようとする対象を表すテンプレート $t(x, y)$ を画像 $f(x, y)$ 中の点にその中心が重なるようにし, テンプレートとそれと重なる画像の部分パターンとの類似度を測り, その値を点 (i, j) に対象が存在する確からしさとする手

法である。

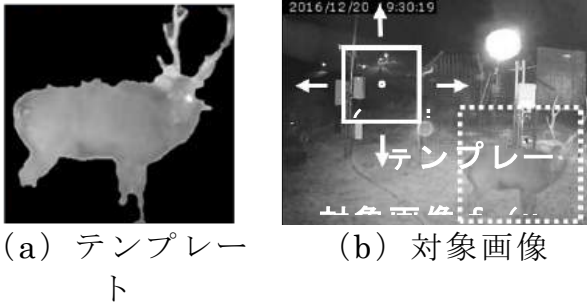


図8 テンプレートマッチング

この注目領域とテンプレートの類似度は式(1)の相関計算によって算出する。類似度 r は $-1 \sim 1$ の値をとり、相関値が 0 の場合は相関が無く、 1 に近づくほど相関が強くなる。また相関値が負の値をとる場合は負の相関があると言える。

$$r = \frac{\sum_{i=1}^w \sum_{j=1}^h (a_{i,j} - \bar{a}) \sum_{i=x}^{x+w} \sum_{j=y}^{y+h} (b_{i,j} - \bar{b})}{\sqrt{\sum_{i=1}^w \sum_{j=1}^h (a_{i,j} - \bar{a})^2} \sqrt{\sum_{i=x}^{x+w} \sum_{j=y}^{y+h} (b_{i,j} - \bar{b})^2}}$$

(

a : テンプレートの濃度値

b : 対象画像の濃度値

\bar{a} : テンプレートの平均濃度値

\bar{b} : 対象画像の平均濃度値

w, h : テンプレートの大きさ

x, y : 探索開始座標

)

4.2. テンプレートマッチングによる動体位置の特定

テンプレートマッチング用いた動体位置を特定する処理の流れは以下の繰返しとなる。

① カメラから画像取得し、動体領域を抽

出する。

- ② 動体の検出がなければ①へ戻る。
- ③ 各抽出した動体に対してテンプレートマッチング処理を行う。
- ④ マッチングしない動体があれば、それをもとに新たにテンプレートを作成する。
- ⑤ マッチングした動体でテンプレートを更新する。
- ⑥ 一定回数以上マッチングのないテンプレートは削除する。

5. カルマンフィルタによる動体領域推定

5.1. カルマンフィルタとは

カルマンフィルタとは、時々刻々与えられる観測データを用いて、システムの状態を逐次的に推定するオンラインアルゴリズムである [6]。

カルマンフィルタには図9のように「更新」と「予測」の二つの手続きがある。更新の手続きでは誤差が最小になるように観測値をもとに推定値を補正する。予測の手続きでは推定値をもとに次の時刻の推定を行う。

5.2. カルマンフィルタによる動体位置推定の流れ

本システムでは動体領域が重なってい

る場合でも正確に頭数をカウントするために、カルマンフィルタを用いて動体の追跡を行う。

動体位置の推定にカルマンフィルタを適用した場合、処理の流れは以下①から⑧の繰り返しとなる。

- ① カメラから画像を取得し、動体領域を抽出する。
- ② 抽出した各動体に対してテンプレートマッチングを行い、実際の動体位置を観測する。
- ③ ここでマッチングしなかった動体がある場合やテンプレートがそもそも存在しない場合は、その動体の情報をもとにカルマンフィルタの初期化処理とテンプレートの作成処理を行う。
- ④ マッチングした動体に対して、これまでの動体位置情報をもとに、次の動体位置はこのあたりだ、という位置予測を行う（カルマンフィルタの予測）。
- ⑤ 実際の位置と予測した位置の誤差をもとに、次の予測ではどれくらい値を進めれば良いかを計算する（カルマンフィルタの更新）。

- ⑥ マッチングした動体領域を次の状態のテンプレートと設定する。
- ⑦ 一定回数以上マッチングしなかったテンプレートは削除する。
- ⑧ 推定した位置情報をもとに、動体数をカウントする。

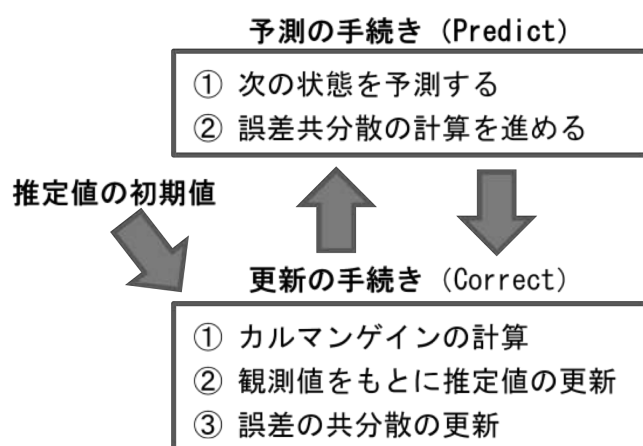


図9 カルマンフィルタの処理の流れ

6. 実験1：テンプレートマッチングによる追跡検証

6.1. 内容

テンプレートマッチングによる位置特定の精度を確認するために、前回の評価実験と同様の画像 1000 枚を対象に評価実験を行ない、テンプレートマッチングの際に算出した類似度の評価を行った。前回と同様に精度の正確さの指標として正解率を算出した。正解率は式 (2) によって求めた。

正解率

$$= \frac{\text{マッチング類似度が 60\%以上の頭数}}{\text{全体の頭数}} \quad (2)$$

6.2. 結果

評価実験の結果を表 1 に示す。テンプレートマッチングの正解率は、64.6%となった。マッチングに失敗した主な原因

としては、鹿が餌の捕食時などに首を上げ下げした際（図 10）にテンプレートと実際の鹿とが大きく異なったり，そもそも背景差分処理の際に鹿領域が正しく抽出できていない（図 11）といったことが挙げられる．

ヒストグラムによる テンプレートマッチング	75.3% (1114/1479)
--------------------------	----------------------

表 1 テンプレートマッチングの実験結果①

項目	正解率
テンプレートマッチング	64.6% (955/1479)



(a) 首を上げている状態

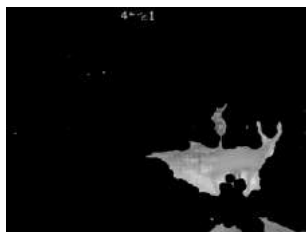


(b) 首を下げている状態

図 10 マッチングに失敗した画像例①



(a) 元画像



(b) 鹿の影を抽出

図 11 マッチングに失敗した画像例②

6.3. 考察

6.2 節で述べた問題を解決するために，ヒストグラムによるテンプレートマッチングを検討した．これはテンプレートと対象の領域のヒストグラムを求め，その相関値によってマッチング判定を行う処理である．この処理を追加し，同様に評価実験を行った結果を表 2 に示す．これにより，正解率を 10%ほど上げることに成功した．

表 2 テンプレートマッチングの実験結果②

項目	正解率
----	-----

7. 実験2：カルマンフィルタによる位置推定検証

7.1.内容

カルマンフィルタによる位置推定の精度を確認するために、2秒間隔で撮影した画像1000枚（図12）を対象に評価実験を行ない、カルマンフィルタの推定位置と実際の観測位置との比較を行った。その際に精度の正確さの指標として正解率を算出した。正解率は式（3）によって求めた。



図12 対象画像の例

正解率

$$= \frac{\text{誤差が 30pixel 以内の頭数}}{\text{全体の頭数}} \quad (3)$$

7.2.結果

図13に、カルマンフィルタによる推定位置、その次の状態の観測位置である動物領域の中心座標及びそれらの誤差とを

比較したグラフを示す。また表3にその正解率を示す。表3より、カルマンフィルタによる位置推定が高い精度で行われていることが分かる。また図13を見るとわかるが、鹿が動き始めたと思われるところで誤差が大きくなっていることが分かる。

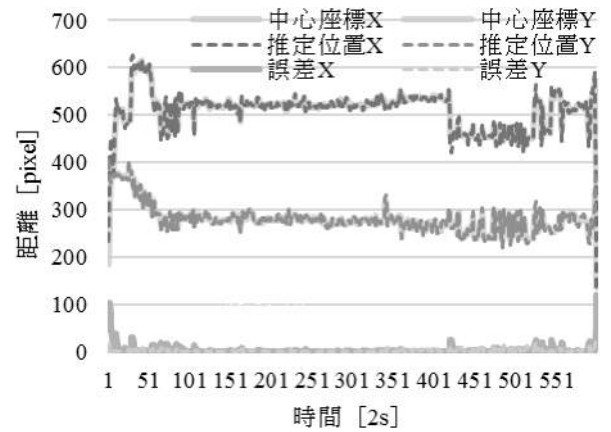


図13 推定位置，観測位置及び誤差の比較

表3 カルマンフィルタによる位置推定の実験結果

項目	正解率
X座標	96.2% (1420/1476)
Y座標	99.7% (1472/1476)

7.3.考察

鹿が動き始めの際に、誤差が大きくなるという問題があった。しかしその後はカルマンフィルタによって補正され、誤差が小さくなっていった。このことから、

鹿を追跡していく上で問題ないと考えている。

8. 実験 3：全体的なカウント精度の検証実験

8.1. 内容

鹿が正しく抽出することができるかを確認するために、今までの評価実験と同様の画像 1000 枚を対象に評価実験を行った。その際、システムによって抽出した鹿の重心座標と、手動で入力した鹿の重心座標との比較を行ない、正解率を算出した。なお正解率は式 (4) によって求めた。

正解率

$$= \frac{\text{重心座標の差が 30pixel 以内の頭数}}{\text{全体の頭数}} \quad (4)$$

8.2. 結果

評価実験の結果を表 4 に示す。全体の正解率は、65.2%とやや低い値となった。また、画面前にいる鹿（図 14 (a)）と、画面奥にいる鹿（図 14 (b)）とに分けて正解率を算出した結果、画面前に鹿がいる場合の正解率は 92.8%と高い値に、画面奥に鹿がいる場合の正解率は 53.4%と低い値となった。

表 4 カウント精度の実験結果

項目	正解率
全体	65.2% (964/1479)
鹿が画面にいる場合	92.8% (410/442)
鹿が画面奥にいる場合	53.4% (554/1037)



図 14 鹿の位置

8.3. 考察

画面奥にいる鹿は、カメラの有効範囲外であるため正しく抽出できない場合が多々あった。このような場合には、檻の中に別のカメラを設置し、檻の中と外で撮影することができれば解決できると考えている。

画面前にいる鹿は、比較的高い精度で抽出することができたが、図15のように誤検出する場合があった。これは、鹿の影の部分が差分処理にて抽出されることが原因である。現在の手法では、このような影を取り除くことができない。



(a) 元画像 (b) 抽出領域

図15 抽出の失敗例

9. まとめと今後の課題

本システムでは、鹿が重なって写っている場合でも正確に頭数をカウントするために、カルマンフィルタを用いて動体領域の追跡を行った。その結果、高い精度で鹿を追跡することができたが、追跡

を行うために必要なテンプレートマッチングや鹿領域の抽出処理で多くの誤検出が発生した。現在の手法ではこれ以上改善することが難しい。

そこでサーマルカメラを用いた動体領域の抽出を提案する。通常のカメラとサーマルカメラとを同じ位置、角度及びタイミングで撮影し、サーマルカメラの画像と通常のカメラの画像との積をとることで、動体領域が正確に抽出できると考えている。

今後の予定としては、サーマルカメラと通常のカメラで同じタイミングで撮影し、その画像をもとに評価実験を行うつもりである。

謝 辞

本研究は、農研機構「革新的技術開発・緊急展開事業（うち地域戦略プロジェクト）」により実施した。

文 献

- [1] 農林水産省：“全国の野生鳥獣による農作物被害状況について（平成26年度）”，http://www.maff.go.jp/j/seisan/tyozyu/higai/h_zyokyo2/h26/，2017年1月19日参照
- [2] 株式会社アイエスイー：“まる三重ホカクン”，<http://www.ise-hp.com/hokakun.html>，2017年1月19日参照
- [3] 阿部豪，長瀬一彦，白井義美，坂田宏志：“野生動物対策におけるスマートセンサーの活用”，兵庫県立大学 自然・環境科学研究所 森林動物系，兵庫県，<http://www.wmi->

hyogo.jp/measures/pdf/capture04.pdf,
(2011.12)

- [4] 北原司, 中井一文, 山本陽祐, 木村佳嗣, 江崎修央, 山端直人: “深度情報を利用した害獣の自動係数システム”, FIT2015 (第14回情報科学技術フォーラム), O-061, pp.577-580 (2015)
- [5] 田村秀行: “コンピュータ画像処理入門”, pp.103-107, 総研出版, 東京 (1985)
- [6] 片山徹: “新版 応用カルマンフィルタ”, pp.6-9, 朝倉書店, 東京 (2008)