

## 深度情報を利用した害獣の自動計数システムの開発

### Investigation of Counting System using Depth Information for Harmful Animals

北原 司<sup>†</sup> 中井 一文<sup>†</sup> 山本 陽祐<sup>†</sup> 木村 佳嗣<sup>†</sup> 江崎 修央<sup>†</sup> 山端 直人<sup>‡</sup>  
 Tsukasa Kitahara<sup>†</sup> Kazufumi Nakai<sup>†</sup> Yosuke Yamamoto<sup>†</sup>  
 Yoshitsugu Kimura<sup>†</sup> Nobuo Ezaki<sup>†</sup> Naoto Yamabata<sup>‡</sup>

#### 1. はじめに

平成 24 年度の野生鳥獣による農作物の被害状況は約 230 億円にもおよんでいる [1]。農作物被害金額の推移を見ると、平成 18 年度は 196 億円であったが、平成 22 年度以降 230 億円となっている。中でも、シカとイノシシの被害額は合わせて 140 億円程度となっており、重点的に対策を講ずる必要があるといえる。

シカやイノシシなどの害獣は罠による捕獲が主流となっている。しかし、罠での大量捕獲を実施する場合には、あらかじめ餌付けをし、害獣が出現しやすい時間を調査したうえで、罠を落とす必要があるなど容易な作業であるとは言えない。

近年、これらの労力を削減するために人工知能でシカ・イノシシ・サルを自動捕獲する「かぞえもん」[2]や獣害罠の遠隔監視・操作システムである「まる三重ホカクン」[3]が開発・販売されている。「かぞえもん」は、罠内の害獣が設定数を超えれば罠を自動作動させ大量捕獲を可能とするシステムである。一方、「まる三重ホカクン」は、携帯やパソコンなどの通信端末で罠の様子が確認し、遠隔操作で罠を作動させることができるシステムである。

これらのシステムを用いれば従来の害獣罠の作動装置のように常に罠付近で待機する必要はなくなるが、効率的な捕獲を続けるためには餌付けを行ってから害獣が罠に侵入するまでの罠外の状況の経過監視も必要となってくる。

そこで我々は、罠内外の害獣の計数を自動的にを行い、クラウド上のデータベースへ記録し、グラフと映像データを連動させて提示することで、利用者が適切な捕獲時期の検討に利用可能なシステム開発を行っている。このうち、本論文では Kinect で深度画像を取得し、画像処理を用いて害獣の数を自動カウントする方法について述べる。

#### 2. システム概要

開発中の害獣自動計数システムの構成を図 1 に示す。本システムは、動物のカウントを行うために Kinect と PC を捕獲罠付近に設置、害獣の計数データを保存しておくクラウド上のデータベース、および情報配信を行う WEB サーバによって構成されている。

システムの大きな処理の流れとして Kinect で罠を監視し、画像処理によって一定時間ごとに動物の数をカウントする。カウントを行う際に、クラウドのデータベースへ日付・時間と共に動物の数を保存しておき、もし罠の中に一定数以上動物が入っていれば、猟師の通信端末にメールを送信する。メールを受け取った猟師は、グラフで表示される過去の動物の数と罠の中のリアルタイムな映像を確認し、遠隔

操作で捕獲を行う。

このようにクラウドにデータを保存しておくことにより web ページとして害獣の出没した日時と動物数をグラフで表示でき、害獣の出没しやすい時間帯の傾向を知ること、罠を作動させる日時の予定を立てることができる。

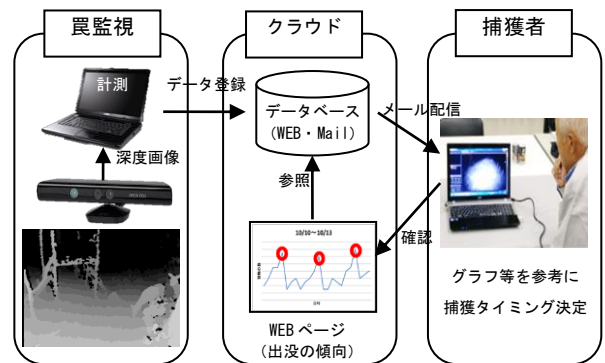


図 1 システム構成

#### 3. 害獣の自動カウント処理

##### 3.1 処理の流れ

動物のカウントを行うための処理の流れを図 2 に示す。まず罠付近の深度画像を取得する。続いて背景消去処理を行ったあと、平滑化処理・2値化処理を行う。2値化処理を行った画像に対してラベリングを行い、動物のカウントを行う。この時、2値化処理については、得られた害獣領域の分散値に基づいて再帰的に実施することで、重なって撮影された害獣の切り分けが行えるように処理を実装した。

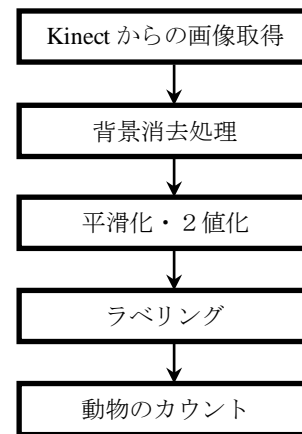


図 2 処理の流れ

<sup>†</sup> 鳥羽商船高等専門学校

National Institute of Technology, Toba College

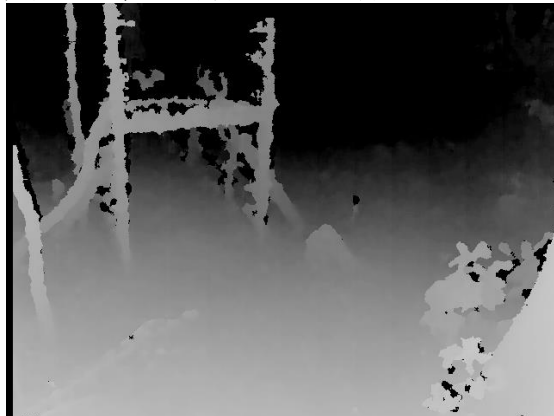
<sup>‡</sup> 三重県農業研究所

Mie Prefecture Agricultural Research Institute

### 3.2 背景除去処理

本稿における背景除去処理とは、同じアングルで撮影された二つの画像において、同じ位置にある画素同士の絶対値の引き算を行うことで画像における変化部分を検出する。まず、事前に何も動物が映っていない背景を深度画像として取得し、背景画像とする。そして、一定間隔で深度画像を取得し、その取得した画像と背景画像で差分を行う。これにより、画素値が一致している背景領域は 0(黒)となり、動物が映っている領域は画素値が高くなる。

図 3(a)に背景画像、図 3(b)に取得した画像、図 3(c)に差分画像を示す。図 3(c)より、変化のあった全ての領域が出力されていることがわかる。なお、背景画像は時間とともに変化するため、一定間隔で自動更新するように設定した。



(a) 背景画像



(b) 取得した画像



(c) 差分画像

図 3 背景除去処理

### 3.3 平滑化・2 値化処理

背景差分を行った画像には、図 3(c)に示す通り雑音が多く含まれるため、メディアンフィルタを使用して平滑化を行った。また、対象領域の切り分けのために固定閾値を用いて 2 値化を行った。



図 4 平滑化と 2 値化の適用

### 3.4 ラベリング処理による不用領域の消去

図 4 の画像に対してラベリング処理を適用し、小さい領域を除去した。図 5 に処理結果を示すが、この場合 2 頭のシカがラベル付けされていることが分かる。

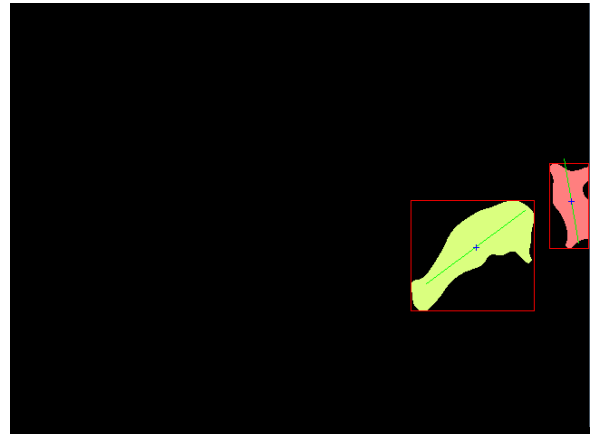


図 5 ラベリング

### 3.5 動物のカウント処理

ここで、図 5 ように出力された画像については 2 頭のシカが別々にラベル付けされているためカウントは容易に行えるが、図 6 のように複数の動物が重なって映ることも考えられる。

しかしながら、今回利用した深度画像の距離情報を用いればこれらの領域を切り分けることが可能であるため、正しい頭数を計測することが可能となる。

以下に、その手法について詳細に述べる。

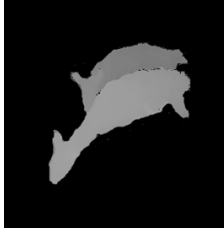


図 6 動物が重なって撮影された例

図 6 を単純に 2 値化すると 1 つの領域として抽出される。そこで、抽出された 2 値化領域に対して深度情報の分散値を計算する。深度画像では、カメラ (Kinect) から動物までの距離に応じて画素値が決まるので、複数頭が重なっている場合でも、それぞれの領域の画素値には差が出る。つまり 2 値化領域の深度画像の分散値が高い傾向が出る。したがって 2 値化領域の分散値をみて、分散値が低ければ一頭であるとし、高ければ複数頭重なっていると判断する。

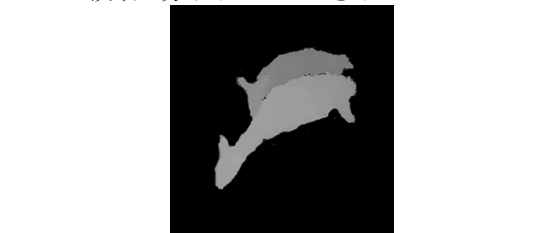
今回は、予備実験の結果から複数頭であると判断する際の分散値の閾値は 1.0 とした。図 7 (a) には一頭である場合の領域 (分散値が 0.8), 図 8 (b) には二頭重なっている領域 (分散値が 1.4) を示す。なお、閾値が 1.0 以上である領域については大津の 2 値化を用いて分割した後に再帰的に処理を繰り返すことで、3 頭以上の害獣が重なっていることも検出可能とした。



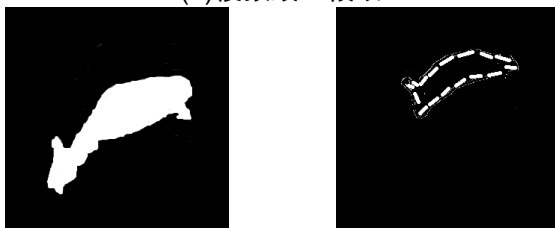
(a)分散値 0.8 の領域 (b)分散値 1.4 の領域

図 7 一頭と複数頭の判断

図 8 (a) の領域について 2 値化した後の分散値が 1.0 以上である場合、大津の 2 値化を適用することで図 8 (b) と図 8 (c) の二つの領域に分けることができる。



(a)複数頭の領域



(b)分割領域 1 (c)分割領域 2

図 8 大津の二値化による分割

## 4. 評価実験

### 4.1 動物のカウントについて概要

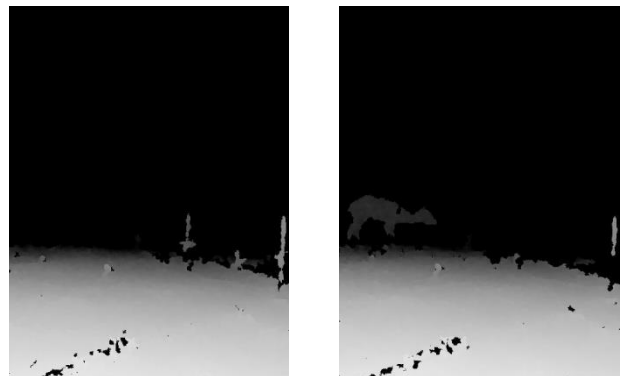
本システムが動物に対してどれほどの精度で自動カウントを行うことができるか確認するため、図 9 のように山中の害獣捕獲罠付近で動作させ、評価実験を行った。



図 9 罠と監視システム  
(PC・Kinect は箱に格納)

評価には、平成 27 年 1 月 20 日の 18 時から 0 時までに撮影された 360 枚を利用した。このうち、114 枚に鹿が映っていた。画像のうち 1 頭のみ鹿の映っていたのは 100 枚、残りの 14 枚に 2 頭が映っていた。

実際に撮影した背景画像 (鹿のいない画像) を図 10 (a) に、動物が映っている際の深度画像を図 10 (b) に示す。



(a)背景画像 (b)取得した画像

図 10 罠付近における深度画像

### 4.2 動物のカウントの結果と考察

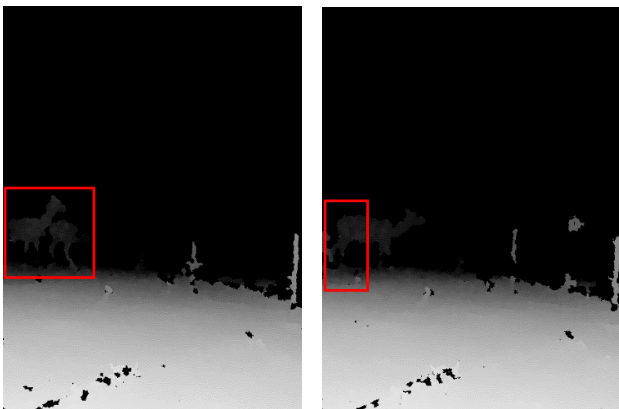
動物の自動計数の結果のグラフを図 11 に示す。結果として、99.2% の鹿の数の検出に成功しており、本手法の有効性が確認された。

図 11 のグラフにおける点線が実際のシカの数の推移で、実線が本手法によって自動計数された数の推移である。グラフを見れば分かる通り、18 時 50 分ごろの計測に失敗している以外は、一致している。



図11 動物のカウント実験結果

正しく鹿が検出できなかった画像の例を図12に示す。図12(a)は、鹿が縦に並んで歩行している瞬間を捉えた画像であり、鹿の持つ画素値（深度情報）がほぼ同じ値になってしまったため、鹿領域の分散値が低くなったため切り分けができなかった。これについては、領域の縦横比や複雑度を指標として加えることにより切り分け可能になると考えている。また、図12(b)は、鹿の脚のみが映っている画像であり、抽出領域が小さかったためノイズと判断された削除された例である。この画像単体で、鹿が含まれていると判断することは極めて困難であるうえ、本研究の目的は鹿の出現時間の推定に使うことであるので、今後の評価においては対象画像としない予定である。



(a) 2頭が並んでいる例 (b) 脚のみが撮影された例  
図12 誤抽出した例

## 5. おわりに

本稿では、Kinectで深度画像を取得し、画像処理によって害獣の数を自動カウントする方法について研究を行った。また、実際に山の中で本システムを動作させ、実際の動物数と自動カウントした数についての評価実験を行った。評価実験の結果より、害獣の付近で撮影した鹿の画像から、99.2%の高確率でカウントすることができた。

今後の課題としては、評価実験で誤検出した問題を解消できるように複数頭の識別の処理を見直しカウントする精度の向上を目指すとともに、クラウド上のデータベースへの保存とWEBによるグラフ描画、メール送信などを組み合わせ利用者に通知を行う機能実装を進めたい。

## 謝辞

本研究は、「革新的技術緊急展開事業（うち産学の英知を結集した革新的な技術体系の確立）」から支援をいただき「ICTを用いたシカ、イノシシ、サルの防除、保護、処理一貫体系技術の実証」の一部として実施している。

## 参考文献

- [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』, [http://www.maff.go.jp/j/seisan/tyozyu/higai/h\\_zyokyo2/h23/index.html](http://www.maff.go.jp/j/seisan/tyozyu/higai/h_zyokyo2/h23/index.html) (2015/04/14)
- [2] 株式会社一成「かぞえもん」, <http://www.issei-eco.com/products/kazoemon/images/kazoemon.pdf>
- [3] 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> (2015/04/15)