Predictive Animal Tracking for Invasive Species Identification and Elimination

Ben McEwen
University of Canterbury
Christchurch, New Zealand
bmc142@uclive.ac.nz

Professor Richard Green
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

Menno Finlay-Smit
Clare McLennan
The Cacophony Project
Christchurch, New Zealand
menno@cacophony.org.nz
clare@cacophony.org.nz

***Abstract*—This paper presents the development of a predictive tracker for animal identification and the elimination of pest species. The methods proposed in this paper aim to extract tracking data about the movement of animals from thermal camera recordings to provide additional data for a machine learning algorithm to improve the identification of animal species. It also aims to determine a method of predicting the future location of animals to control an automated turret used to eliminate invasive predator species. Some of the limitations of existing methods are their ability to handle noise and occlusion and also operate robustly in a variety of dynamic environments. This was successfully reduced through the use of an Ordinary Kalman filter. A combination of a moving average filter and a Kalman filter was implemented to successfully determine the velocity and future position of the animal.**

Keywords – Animal Identification; Predictive Tracking; Kalman filtering; Noise/Occlusion Reduction.

I. Introduction

Invasive predators such as possums, stoats and rats pose a significant threat to New Zealand's native birds. Wildlife ecologist, John Innes, estimates that over 26.6 million native chicks and eggs are lost each year [1]. Existing technologies such as traps and poison drops are limited by cost, resource shortages and effectiveness. Most traps requiring manual reset and poison has a number of environmental implications [2].

The Cacophony Project (1) is developing technologies to overcome these limitations. They estimate an increased trapping efficiency of up to 80,000 times [3].

Figure 1: The Cacophony Project logo

The current system under development, The Cacophonator [4], uses advanced image processing to automatically detect animals and send recordings to an online database where a machine learning algorithm is trained to identify the animal's species. The aim is to develop a system that can be deployed across New Zealand to automatically lure, identify and track animals in real-time. The system will predict the future location of the animal and use this information to control an automated turret to eliminate animals identified to be invasive predators.

A number of factors limit the effectiveness of the current identification algorithm. Camera resolution and noise introduced by the dynamic environment make it difficult to consistently identify and track animals. Tracking animal targets adds additional complexity as it is not a static shape. As the animal moves through space the angle at which it presents itself to the camera changes meaning that physical features may not be recognisable. Another factor is the effects of occlusion. As the animal passes behind objects, features become hidden and the animals form is broken up resulting in the misidentification of multiple animals when there is only one.

This paper proposes a method of improving tracking data by reducing the effects of occlusion through Kalman filtering. This improves the validity of training data and therefore the consistency of animal identification. This allows movement pattern-based identification when physical features are not identifiable. This paper also researches methods of predicting the future location of the animal. The methods discussed in this paper quantify the performance of tested and researched methods to provide efficient, real-time tracking applicable to systems with computational constraints.

II. Background

Background research was undertaken to understand what existing animal tracking methods are available. The following two papers discuss methods of animal tracking in laboratory environments. Animal tracking is an important field of research as it is used in animal behavior studies and drug testing.

A. Machine Vision Application in Animal Trajectory Tracking

This study [5] compares five different methods for tracking the trajectory of an animal, in this case guinea pigs. The researchers were interested in monitoring the movement of laboratory animals to understand the correlation between the animal's movement and its health. In this study a visual method was selected over more conventional methods such as GPS or RFID tracking due to the size of the sensors which would need to be attached to the animal and the cost.

One of the most successful methods outlined in this paper was using differential methods [6]. Due to the relatively homogeneous testing environment the previous frame could be subtracted from the current frame. The result was then thresholded, producing a group of white pixels representing the change in the system and therefore the animal's movement. Various methods of removing noise such as adaptive thresholding [7] and taking the standard deviation of the pixels to identify outliners were implemented.

The centroid of the animal was then determined. The researchers used the maximal and minimal pixel positions to draw a bounding box around the animal. The centroid of the animal is then approximated to the centroid of the bounding box. The current and previous centroid locations are then compared along with the number of white pixels to determine the magnitude of the movement.

This study does not attempt to predict the future position of the animal but instead explores methods of monitoring the animal's path for later analysis. While not directly applicable to predator tracking much of the pre-processing methods such as the differential and thresholding methods have been implemented by The Cacophony project. The method for finding the centroid of the animal was explored further.

B. *Improving Animal Tracking Algorithms with Adaptive Search Window and Kalman Filters*

This research [8] investigates ways of improving animal tracking algorithms using adaptive windows and Kalman filters by the University of Lisbon. The focus of this research was the tracking of insects and fish for animal behaviour studies with the aim of recording the distance travelled, the direction and velocity of the target. This study uses commercial software LabVIEW/IMAQ Vision with proprietary image processing using adaptive search windows and the Kalman filter. This paper is relevant to the application being researched because similar issues with occlusion were experienced. The shape of fish changes drastically as they change direction and occlusion would occur when fish or insects moved behind objects.

The researchers found that commercial software alone was not sufficient and could not be used in real-time analysis. Adaptive search windows were used to reduce the amount of data being processed for faster computation and improved target recognition. The window is centered on the centroid of the target and the dimension of the window is dependent on the velocity of the target. A 2-dimensional ordinary Kalman filter [9] was implemented to determine the position and velocity of the animal. The Kalman filter is a recursive algorithm which is used to estimate the state of a system. It is a powerful tool due to it being computationally efficient and able to predict future states even when the exact nature of the system is not known.

This research concluded that the commercial software package LabVIEW/IMAQ Vision was not robust enough for reliable animal tracking due to the processing time. Significant improvements were experienced after adaptive windowing and Kalman filtering were implemented and meaningful data was able to be extracted. The paper also outlines an alternative method for determining the centroid of the animal. The pixels in the x and y direction were summed and then divided by the area to find the centroids in the x and y direction. The area was assumed to be elliptical and was calculated using $A = \pi ab$. Related research can be found here [10], [11].

C. *Limitations*

A number of limitations with existing research were found. Most predictive tracking of animals was undertaken in homogeneous lighting conditions. Testing took place in controlled laboratory environments where animals were confined to a small space. The camera and items within the cage were positioned to reduce occlusion. As the system must be capable of operating in a broad range of dynamic environment the proposed methods must overcome these limitations.

III. Methods

The methods proposed in this paper were implemented in Python using OpenCV libraries [12]. Software was written using the Wing 101 IDE. The operating system was Linux Mint 19.1 Cinnamon with an Intel Core i7-8700 CPU. All animal recordings were captured using a FLIR Lepton 3 thermal camera [13] with a focal plane of 160x120 pixels and a frame rate of 8.7 Hz. A sample frame from a recording of a possum is shown below in Figure (2).

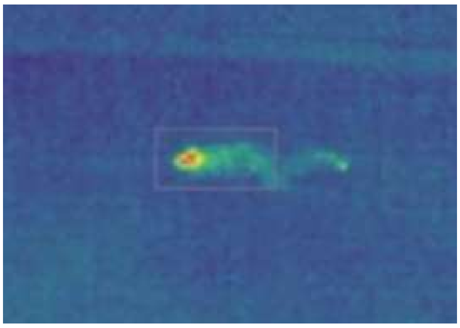

Figure 2: Recording of possum using FLIR Lepton 3 camera.

A. *Pre-processing*

Recordings underwent pre-processing using existing open-source Cacophony Project software [14] to remove noise and distinguish the animal from the surroundings. This application is particularly challenging as the software must perform reliably in a range of environments. The system stores the initial frame before the animal enters the field of view and then differential methods are applied to subtract the current and initial frames, removing background features as shown in Figure (3).

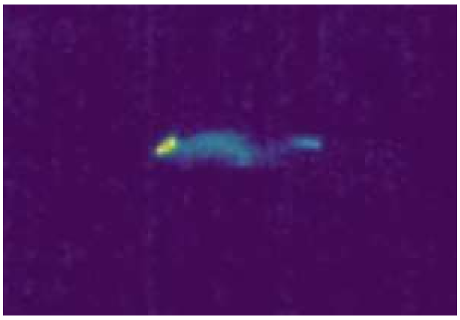

Figure 3: Recording after subtraction of current and initial frame.

As the environment is dynamic and factors such as wind can introduce noise, thresholding is used to binarize the frames. Pixels above a specified value are assigned a value of one (white). Any pixels with a value below the specified threshold are assigned a value of 0 (purple).

Erosion and dilation are used to remove noise and join broken sections of a shape. Erosion uses kernel convolution to iterate through pixels within the frame. If all pixels within the kernel are binary 1 then the pixel in the original image will be set to 1, otherwise the pixel will be set to 0 eroding the boundary of the shape and removing any stray white pixels. Dilation is then applied. The method is very similar except only one pixel within the kernel is required to be 1 for the pixel in the original image to be set to 1. This returns the shape to its original size and joins disjointed parts of the shape. This is also useful in reducing the effects of occlusion. For example, possums are often misidentified as two separate animals due to the body of the animal and its tail appearing as two disjointed shapes as shown in Figure (4). Dilating the frames results in the shapes area increasing, reducing the likelihood that the body and tail will appear as two separate shapes.

Figure 4: Occlusion between the body and tail of possum.

B. *Centroid*

Two methods of determining the centroid of the animal were considered. Initially the method described in research paper B was implemented. The centroid of the binarized frame was determined by finding the moment of the shape. The second method was described in research paper A where the centroid of the animal was assumed to be the centre of the bounding box surrounding the animal.

C. *Kalman Filter*

A two-dimensional ordinary Kalman filter was implemented to estimate the x and y coordinate of the centroid of the animal. As shown in research paper B, the Kalman filter is a highly computationally efficient algorithm which is capable of making a prediction about the current and future state of a system. This is especially useful when there is a noisy or unreliable signal such as when occlusion occurs. In the case of tracking a possum occlusion would commonly occur between the main body of the animal and its tail. When this occurred the centre point of the animal would shift backwards, incorrectly showing that the animal had changed direction.

The Kalman filter predicts the position of the animal by calculating a weighting between the previous estimate and the sensor reading, in this case the result of the centroid function. This weighting dynamically changes depending on the uncertainty associated with the previous estimate and the uncertainty associated with the measured value. The Kalman filter is an iterative process which uses the current estimate and gain value to determine the current uncertainty and this value dictates the gain in the next iteration. For example, if the current estimate has a high level of uncertainty then the gain will shift to give sensor readings a higher weighting but when occlusion occurs, and the estimate uncertainty is lower than the sensor uncertainty, the previous estimate will have a higher weighting.

D. *Velocity Prediction – Moving Average Filter*

A moving average filter was implemented to predict the velocity of the animal. The current estimate position and the previous estimate position were stored after each iteration and subtracted to determine the displacement in the x and y directions of the animal's centroid between frames. Dividing this value by the framerate produced the instantaneous velocity of the animal. The estimated position was too erratic for the instantaneous velocity to be useable, therefore a moving average filter was used to find the average velocity. Instantaneous velocity values were stored in a buffer with a sample length of 50. Summing the readings and dividing by the sample length produced the average instantaneous velocity, smoothing the predicted position.

E. *Velocity Prediction – Kalman Filter*

The Kalman Filter can be used to predict the velocity in addition to the position of the animal. As with the previous method the instantaneous velocity in the x and y direction can be calculated by subtracting the current and previous estimate positions. This time, the velocity values are fed back into the Kalman filter. Using the uncertainty of the previous velocity estimate and the uncertainty of the instantaneous velocity the gain can be calculated and used to estimate the next velocity. The predicted position could then be calculated using the filtered velocity. A variable time can be set allowing the program to predict the animals position at a set point in the future. The future position of the animal was assumed to be linear relative to its current position so the instantaneous velocity in the x and y position was multiplied by time to project a point in front of the animal.

F. *Position Prediction – Kalman Filter*

The Kalman filter can be used to predict the future x and y coordinate of the centroid of the animal. This was the simplest method to implement as it did not require the calculation or prediction of the velocity. This can be achieved by looping over the prediction step shown in (1).

$$P_{K+1} = \boldsymbol{A}P_K + \boldsymbol{B}c \quad (1)$$

$\boldsymbol{A}$ is the state transition matrix, P_K is the prediction vector which includes the x and y coordinates, $\boldsymbol{B}$ is the input control matrix and c is the control vector. As there are no control actuators the control vector is not used in this case and the second term is ignored. Increasing the number of iterations increases how far the program is predicting ahead of the animal.

IV. Results

A. Centroid

The centroid was determined using the moment of area method, achieving good results, but the bounding box method was selected for the final version as it utilized existing Cacophony project software. Also, the moment of area method was not well suited in this situation as the dilation pre-processing used to link sections of the mask alters the area and shape of the animal reducing the centroids accuracy.

B. Filtering

The implementation of an ordinary Kalman filter successfully reduced the effects of noise and occlusion. The following Figure (5) shows the real position of the animal produced by the centroid function, shown in blue, and the Kalman filters estimate, shown in red. The animal is moving across the cameras field of view to the left at a constant speed.

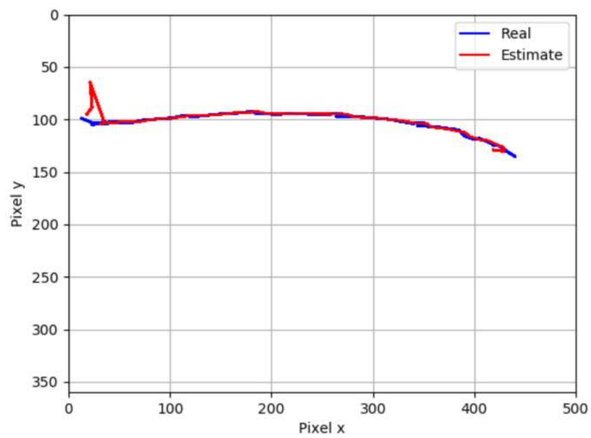

Figure 5: The coordinate of the animal calculated by the centroid function and estimated by the Kalman filter.

Note that the initial estimated position of the animal is at (x,y) = (0,0) before the animal enters the frame. The filter quickly converges, on average taking two iterations to converge with the real value. Erroneous estimates as the animal is entering or leaving the frame were removed by trimming the first and last estimate data points.

A significant reduction in erratic movement patterns was observed confirming the results shown in paper B. Figure (6) shows a magnified version of the original figure. The raw centroid coordinates produced by the centroid function are highly susceptible to the effects of occlusion. As the tail of the animal becomes visible, the centroid position shifts backwards resulting in the erratic jumps visible in the animal's position.

C. Predictive tracking

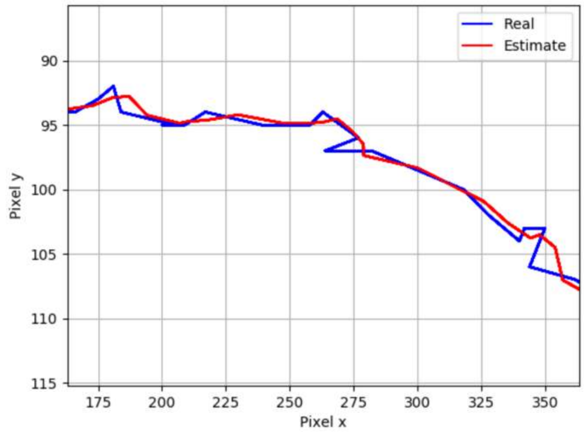

Figure 6: Magnified plot of the calculated and estimated centroid position of the animal.

Three methods of predicting the future location of the animal were implemented and tested. Each method was able to successfully predict the location of the animal in time. Using a typical recording each method was compared to determine the most accurate and computationally efficient prediction method.

1) Method 1 – Moving Average Filter

This method produced average results. A moving average filter with a sample length (M) of 50 elements was implemented. When the average of the absolute difference between the predicted location and the estimated centroid was calculated over a section of the recording an error of 31.3 and 6.8 pixels in the x and y directions respectively. Figure (7) shows the estimated centroid positions and the predicted positions.

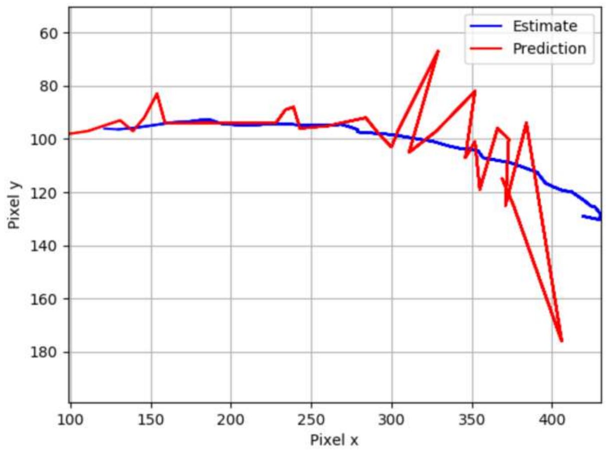

Figure 7: Estimated and predicted position of the animal using a moving average filter.

The moving average filter takes M – 1 iterations to converge. This was observed as there were initially large fluctuations in the predicted positions, but the prediction improved over time as the number of samples increased. Experimenting with different sample lengths and prediction times were found to have a significant impact on the convergence time and accuracy of the prediction. While these parameters can be optimized to produce adequate results, this is not recommended as a it is not robust under all conditions and takes too long to converge. The average elapsed time per iteration is 0.1399 s.

2) Method 2 – Kalman Predicted Position

This method produced poor results as shown in Figure (8). An average error of 80.5 and 19.7 pixels in the x and y directions was calculated when predicting five frames ahead. Reducing the prediction range significantly improved results with a Two frame prediction resulting in an error of 36.7 and 9.5 which is comparable to the results of method one. This method has a fast convergence but is not recommended as the prediction accuracy is significantly reduced the higher the prediction range. As this methods prediction range is in terms of iterations not time it is also difficult to quantify how the animals predicted position relates to its actual position at a given time point. The average elapsed time per iteration was 0.1099 s.

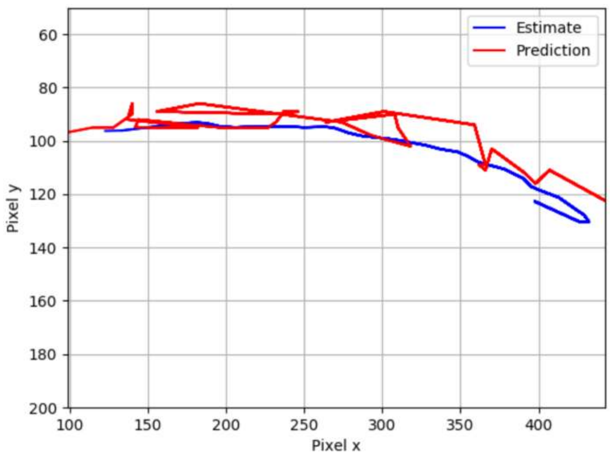

Figure 8: Estimated and predicted position of the animal using the position predicted by a Kalman filter.

3) Method 3 – Kalman Predicted Velocity

This method produced poor results with an average error of 72.4 and 16.8 pixels in the x and y directions when predicting five seconds into the future as shown in Figure (9). This method experienced a high amount of fluctuations in the x direction but had a fast convergence time similar to method two. The average elapsed time per iteration was 0.1119 s.

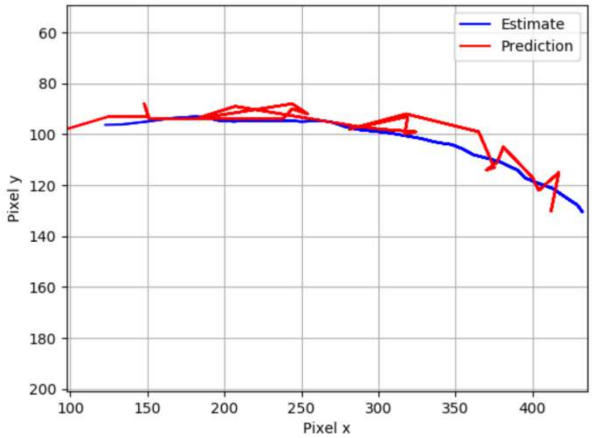

Figure 9: Estimated and predicted position of the animal using the velocity predicted by a Kalman filter.

4) Final - Combined Method

Of the three methods initially implemented the moving average filter had the highest steady state accuracy. Due to the slow convergence, this method is not applicable to a real-time system. Method two and three both used Kalman filtering to predict the position and velocity respectively. Due to the high efficiency of the Kalman filter algorithm both methods had very fast convergence. These methods were not applicable due to their low accuracy. Hence, a combination of these methods 1 and 3 was implemented to achieve fast convergence and high accuracy as shown in Figure (10). Due to the fast convergence of the Kalman filter algorithm a reduced sample length of M = 5 samples can be used. This improves the stability of the prediction, resulting in a high accuracy while still maintaining a fast convergence.

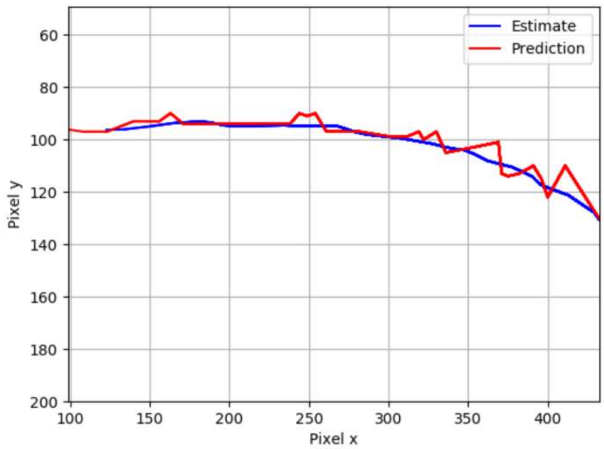

Figure 10: Estimated and predicted position of the animal using a combination of methods.

This method resulted in an average error of 15.4 and 2.5 pixels in the x and y directions and an average elapsed time of 0.1198 s. There was a significant improvement compared to previous methods and a similar efficiency was achieved.

D. Further Testing

Preliminary testing of the software was undertaken using a recording of a possum moving across the field of view at a constant speed. To determine the software robustness further testing was undertaken on multiple recordings. Multiple animal species such as hedgehogs, birds and more possum recordings were tested. The software performed well and was able to predict the future position of the animal while moving in multiple directions and at varying speeds. The following Figure (11) shows the estimated and predicted plots of a bird. The bird moves at various speeds, frequently stopping.

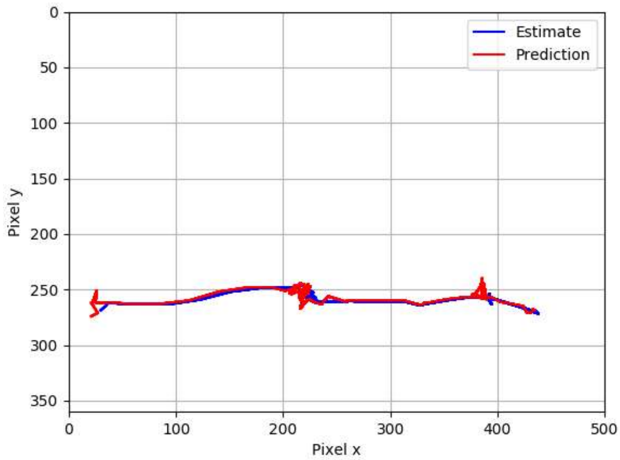

Figure 11: Estimated and predicted position of the animal using a combination of methods.

V. Conclusion

The methods proposed in this paper attempt to improve tracking data and the identification of animal species through movement patterns. The centroid was calculated and an ordinary Kalman Filter was used to filter the animals x and y centroid coordinates and was successfully able to reduce the effects of occlusion. Further testing is needed to determine if this improves the identification of animal species. These results confirm the validity of methods outlined in the research papers [5] and [8] and also overcome the limitations. The proposed methods can be used in real-time and can be applied in dynamic environments.

Three methods of predicting the future location of the animal were initially implemented. The methods were moving average filtering of the velocity, position prediction using a Kalman filter and velocity prediction using a Kalman filter. Method one produced the best results, producing a highly accurate prediction. This method had the disadvantage of requiring a significant amount of iteration to converge causing it to be less effective when implemented on a real-time system. Methods two and three both produced poor results. Unlike method one they were quickly able to converge within a reasonable accuracy, but the predicted position jumped erratically, resulting in high error in the x and y directions. Due to the fast convergence of these methods they are effective on real-time systems, but the lack of accuracy was a major disadvantage.

The accuracy of method three was improved by implementing a moving average filter. Due to the fast convergence of the Kalman filters a much smaller sample size could be used resulting in improved accuracy and a faster convergence time compared to method one.

There is the potential for further improvements in performance using a non-linear Kalman filter such as an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF). More information about these methods can be found here [15].

VI. References

[1] James C. Russell, John G. Innes, Philip H. Brown, Andrea E. Byrom, “Predator-Free New Zealand”, *Conservation Country, BioScience*, Vol. 65, Issue 5, pp. 520–525, May 2015.

[2] Charles Eason, Aroha Miller, Shaun Ogilvie, Alastair Fairweather. “An updated review of the toxicology and ecotoxicology of sodium fluoroacetate (1080) in relation to its use as a pest control tool in New Zealand”, *New Zealand Journal of Ecology,* Vol. 35, No. 1, 19 Dec. 2010.

[3] Grant Ryan, The Cacophony Project. “Making Predator Eradication 80,000 Times More Efficient”, Internet: https://cacophony.org.nz/eradicate-predators-80000x-more-effectively [02/05/19].

[4] The Cacophony Project, Technology, The Cacophonometer, The Cacophonator. Internet: https://cacophony.org.nz/technology, [28/03/19].

[5] Koniar, D., Hargaš, L., Loncová, Z., Duchoň, F., & Beňo, P.. “Machine vision application in animal trajectory tracking”. *Computer Methods and Programs in Biomedicine*, vol. 127, pp. 258-272, Apr. 2016.

[6] M. Piccardi, "Background subtraction techniques: a review". *2004 IEEE International Conference on Systems, Man and Cybernetics.* vol. 4, pp. 3099-3104, Aug. 15, 2004.

[7] N. Bhardwaja, S. Agarwalb, V. Bhardwaj. “An imaging approach for the automatic thresholding of photo defects”. *Pattern Recognition*, pp. 32-40, Aug. 2015.

[8] Trindade, H & Evans, G & Soares Augusto, José & Fonseca, P.J. “Improving Animal Tracking Algorithms with Adaptive Search Windows and Kalman Filters”. *Design of Circuits and Integrated Systems,* Nov 1, 2010.

[9] Welch, G, Bishop, G. “An Introduction to the Kalman Filter”. *Proc. SIGGRAPH* Course. 8, pp. 1-16. July 24, 2006.

[10] Z. Kalafatic, S. Ribaric and V. Stanisavljevic, "A system for tracking laboratory animals based on optical flow and active contours", *Proceedings 11th International Conference on Image Analysis and Processing*, 26-28 Sept. 2001. pp. 334-339.

[11] Z. Kalafatic, "Model-based tracking of laboratory animals", *The IEEE Region 8 EUROCON 2003,* 22 Sept. 2003. *Computer as a Tool.* vol. 2, pp. 175-178.

[12] Sphinx-quickstart, OpenCV 2.4.13.7 Documentation, https://docs.opencv.org/2.4.13.7/, Febuary, 2011. [28/03/19]

[13] FLIR, Lepton LWIR Micro Thermal Camera Module. Internet: https://www.flir.com.au/products/lepton/, [20/05/19].

[14] The Cacophony Project, Technology, solftware. Internet: https://github.com/TheCacophonyProject/cacophony-api, [28/03/19].

[15] J. Simon, Daniel. *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. Wiley-Interscience, edition 1, June 23, 2006.