Multiple Animal Species Detection Using Robust Principal Component Analysis and Large Displacement Optical Flow

Pooya Khorrami, Jiangping Wang, and Thomas Huang
Beckman Institute, University of Illinois at Urbana-Champaign, USA
{pkhorra2, jwang63, huang}@ifp.uiuc.edu

Abstract

This paper examines the problem of detecting various types of animals in video sequences taken in the wild. Given that manually extracting and labeling an animal's position in a video sequence is very labor-intensive, an automatic solution would be extremely useful. We propose a method that provides accurate localization of the animals using a very general approach. Our technique first separates the background from the video frames using Robust Principal Component Analysis. It then locates and filters candidate regions containing the animal using local entropy and large displacement optical flow respectively. We also verify the effectiveness of our algorithm on video sequences of various types of animals.

1. Introduction

For several years now, detection of wildlife in images and video has been an area of great interest amongst biologists. Often times, scientists desire to track and analyze the movement and behavior of various types of animals by viewing videos from camera traps [9]. Unfortunately, locating and identifying the animal in each individual video frame is extremely labor intensive and, at times, difficult. Also, camera trap videos tend to have very low frame rates which cause popular tracking and supervised learning algorithms to fail.

Despite being of great importance, there has been surprisingly little work on the problem of animal detection. One method proposed by Forsyth and Ramanan [8] builds an appearance model for the animal using low-level detectors and mean shift and uses it to detect the animal in future frames. While their method exhibits impressive results, it only deals with three different animal species. Another technique by Burghardt and Calic [2] tracks animals by first detecting their faces via Haar features, however such an approach would fail when detecting animals whose faces are not visible. Other approaches [4] [6] have the user mark or extract the location of the animal by hand. This, of course, is extraordinarily time-consuming for multiple species.

In this paper, we propose a method that is able to detect multiple types of animals in a completely unsupervised fashion. Our algorithm can be summarized as a three step process. First, we use Robust Principal Component Analysis (RPCA) [3] to remove the background of each video sequence. We then take take the remaining foreground regions and isolate the animals using local entropy and connected component analysis. Finally, we incorporate motion information using large displacement optical flow to retain areas in the frames corresponding to large changes in velocity.

The remainder of this paper is organized as follows: Section 2 will describe the details of the algorithm. Section 3 will provide some experimental results. Section 4 will present directions for possible future work.

2. Proposed Method

2.1 Background Subtraction

When dealing with object detection in video, the first step is to locate areas of interest via background subtraction. The key objective of background subtraction is to form a model that accurately captures the information corresponding specifically to the background of a video scene. As a result, any foreground activity can be thought of as a strong deviation from the established model. Background subtraction via adaptive Gaussian mixture models [10] is a commonly chosen method, however it fails when dealing with relatively short length video sequences with low frame rates. Therefore, we must consider another approach.

Let us consider a K frame video sequence with N pixels per frame and a matrix $M \in R^{N \times K}$ where each column of M represents the vector form of each frame in the video sequence. Principal Component Analysis (PCA) attempts to model data using a low-dimensional subspace with entries that are slightly corrupted by Gaussian noise. Although a low-dimensional representation of a video scene is highly desirable, the assumption that entries are only slightly corrupted is invalid when trying to model the foreground. Foreground pixels can be thought of as gross deviations from the background meaning that they will cause the low-dimensional space provided by PCA to not accurately represent the background data.

Robust Principal Component Analysis, on the other hand, states that the data matrix M can be expressed as the sum of a matrix of low-rank L and a sparse matrix S. If we were to inspect the matrix M more closely, we would notice that the columns are highly correlated with each other. Therefore, the low-rank matrix L produced by RPCA will contain columns corresponding to the background of each frame in the video sequence. Meanwhile, the columns of the sparse error matrix S will contain the foreground activity (i.e. the outliers to the background model).

To accomplish this goal, RPCA attempts to solve the following optimization:

$$\begin{aligned} \underset{L,S}{\text{minimize}} \quad & rank(L) + \lambda ||S||_0 \\ \text{subject to} \quad & M = L + S \end{aligned} \tag{1}$$

where $||S||_0$ represents the number of non-zero elements in matrix S. Unfortunately, the above optimization is NP-hard and requires some simplifications. The optimization can be made convex by replacing the $||S||_0$ term with the L1 norm. Also, the rank of the matrix L can also be replaced with its L1 equivalent: the nuclear norm. This yields the following objective function:

$$\begin{aligned} \underset{L,S}{\text{minimize}} \quad & ||L||_* + \lambda ||S||_1 \\ \text{subject to} \quad & M = L + S \end{aligned} \tag{2}$$

After finding the matrices L and S that minimize the above function, we obtain results like the ones shown in figure 1. The leftmost column displays frames from the original video sequence (i.e. a column of the M matrix) while the center and rightmost columns show the corresponding columns in the low-rank and sparse error matrices respectively. Given that the animal is not a part of the background, it is always found in the sparse error component.

2.2 Extracting Candidate Regions

After each frame has been split into its low-rank and sparse component, we do further processing on each sparse error frame to locate the animal. Upon closer inspection of each sparse error frame, the viewer will notice that the majority of its pixels are zero, or normalized to 128, as desired. However, there are smooth regions corresponding to the animal and moving objects in the background. To extract the contours separating these smooth regions and the background, we compute the local entropy of each pixel in the frame.

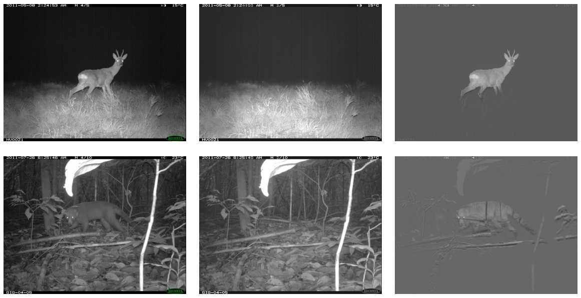

Figure 1. Application of Robust Principal Component Analysis to Specific Video Frames with Corresponding Low-Rank and Sparse Error Output Frames

Traditionally, entropy is defined as the amount of disorder within a given probability distribution. One can verify that distributions with higher entropy tend to have higher variance as well. When considering images, we can model the pixels as a probability mass function. However, instead of computing the entropy of the entire image, we instead consider a small $N \times N$ neighborhood of each pixel and calculate the entropy locally [5]. From this, we can see that areas of similar intensity will have relatively low entropy while sharp changes in pixel intensity such as the animal's boundary will correspond to high entropy. A visualization can be seen in the upper-right image in figure 2. The brighter pixels correspond to regions of high entropy.

In order to isolate these high entropy regions, a threshold is applied to create a binary image similar to the one shown in lower-left corner of figure 2. Afterward, every connected component in the image is extracted and used to create a general bounding box. These boxes, as shown in the lower-right corner of figure 2, are used as candidate regions when locating the animal, thereby minimizing the search space within a given frame.

2.3 Large Displacement Optical Flow

In order to select the bounding box containing the animal, the candidate regions must be filtered based on some criterion. Our approach considers the amount of motion contained within each candidate region. If we consider the bounding boxes that result from the local entropy filter, shown in the lower-right image of figure

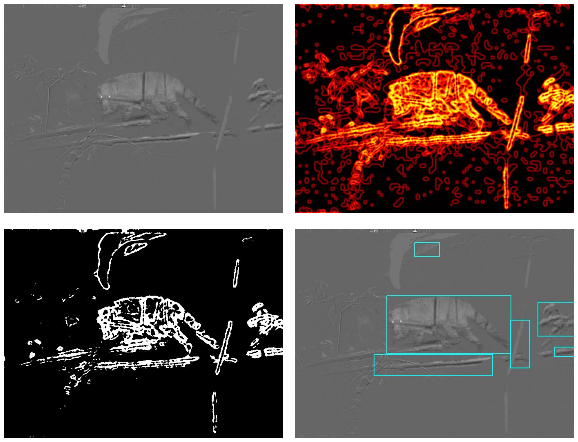

Figure 2. Local Entropy Visualization - White-nosed Coati Sequence

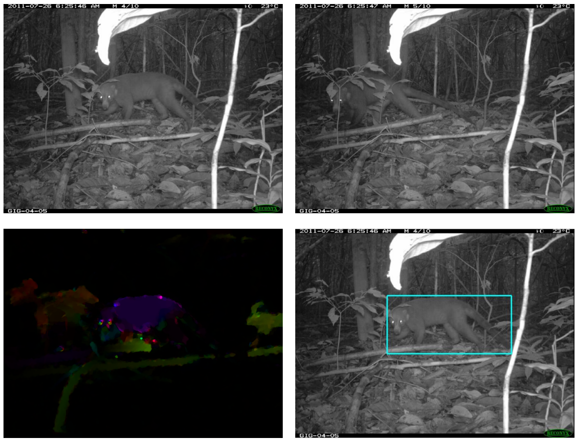

Figure 3. Large Displacement Optical Flow - White-nosed Coati Sequence

2, we can see that they either contain the animal or small moving parts of the background. The fact that each box contains some degree of motion information is a convenient result of the camera being stationary. We contend that the region corresponding to the animal will contain the largest amount of motion.

When modeling the movement of pixels from one frame to the next, the most common practice is to compute the optical flow of the sequence. Optical flow techniques attempt to estimate the velocity of each pixel given a set of adjacent frames. Several algorithms for computing optical flow have been proposed in the past several years, however we selected the technique by Brox and Malik called Large Displacement Optical Flow [1].

In their paper, the authors describe how their method is able to allow for large displacements while still exhibiting high accuracy. They accomplish this by first forming a region hierarchy for each pair of adjacent frames. Then each region is assigned a descriptor containing an orientation histogram and color information. They then perform descriptor matching to establish a set of correspondences between the two images. These descriptor correspondences allow for large displacement in motion, however some extra conditions must be applied in order to preserve the smoothness of the flow field. This process can be represented using an energy minimization function:

$$
\begin{aligned}
E(\mathbf{w}(\mathbf{x})) &= \int \Psi(|I_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - I_1(\mathbf{x})|^2)d\mathbf{x} \\
&+ \gamma \int \Psi(|\nabla I_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - \nabla I_1(\mathbf{x})|^2)d\mathbf{x} \\
&+ \beta \sum_{j=1}^{5} \int \rho_j(\mathbf{x})\Psi((u(\mathbf{x}) - u_j(\mathbf{x}))^2 + (v(\mathbf{x}) - v_j(\mathbf{x}))^2)d\mathbf{x} \\
&+ \alpha \int \Psi(|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2 + g(\mathbf{x})^2)d\mathbf{x}
\end{aligned}
\tag{3}
$$

In the above function, I_1 and I_2 represent the input frames, **x** := (x,y) represents a point in the image, and **w** := (u,v) is the desired velocity vector. α, β and γ are tuning parameters that control the weight of the smoothness, region correspondences, and gradient constancy in the objective function respectively.

Large displacement is extremely useful for our purposes because often times the animals move very quickly from frame to frame creating large changes in position. This could also be because the camera trap is unable to fully capture the animal's motion given its low frame rate. Despite this, we can still attain an accurate flow field to track the animal's motion. We apply large displacement optical flow to every pair of adjacent video frames and extract a flow field. Some examples of images used are shown in the top row of figure 3. The extracted flow field for these two frames is shown in the lower-left corner of figure 3. In these images, the hue indicates the direction of the motion while the brightness indicates the magnitude.

Upon retrieval of the flow field, we compute the maximum magnitude within each candidate bounding box. Since our hypothesis was that the animal exhibits the most motion within a scene, we retain the bounding box with the largest maximum magnitude. A sample result can be seen in the lower-right of figure 3.

3. Experiments and Analysis

To assess the accuracy of our technique, we applied our algorithm to several video sequences of animals

Table 1. Detection Accuracy within 50 Pixel Radius over 10 Different Animals

Animal Type	Frames Correct	Frames Total	Detection Accuracy
Agouti	268	310	86.45%
Coati	89	130	68.46%
Bird	140	210	66.66%
Tinamou	202	230	87.80%
Roe Deer	254	435	58.39%
Capuchin	324	420	77.14%
Coyote	225	315	71.43%
Raccoon	180	247	72.87%
Turkey	252	275	91.64%
Crow	114	140	81.43%
Overall	**2048**	**2712**	**75.52%**

taken at several times of day and in a variety of weather conditions. The sequence lengths varied from 10 to 100 frames and were captured at 1 FPS. This set of data contained approximately 2700 video frames spanning 10 different animal species. We created our own ground truth images by first labeling and storing the bounding box containing the animal, if present. We then ran our algorithm on the same video frames and compared the output bounding boxes with the ground truth. If the centroids of the two bounding boxes were within some threshold, then it was considered a correct detection.

When running our algorithm, we used the Augmented Lagrange Multiplier (ALM) Method provided by Yi Ma and his group at the University of Illinois - Urbana Champaign [7]. Meanwhile, for the local entropy filter we considered a 5x5 local neighborhood around each pixel. To compute the optical flow of the frames, we used the code provided by Brox [1] and set the tuning parameters, α, β and γ to 15, 300 and 5 respectively.

Our method achieves a detection accuracy of 75.52% when considering bounding boxes within a 50 pixel radius of the ground truth. A detailed breakdown of the results can be seen in table 1. From these detection accuracies, we can see that the algorithm performs quite well for a variety of animals. If the radius is increased to 100 pixels, the accuracy increases to 79.42%.

Closer inspection of our results indicated that the algorithm performed poorly in video sequences with a high-degree of background motion such as rain or snow. In these particular instances, objects that would normally be classified as background (snow, rain, etc.) would be considered corruptions to the scene and would subsequently be placed in the sparse error component. As a result, further processing on the sparse error frames would generate bounding boxes that did not contain the animal and ended up generating false detections.

4. Future Work

In this paper, we have shown how an extremely general framework can be used to detect various different animals reliably from camera trap video with low frame rate. Some possible directions for improvement include making the technique more robust to dynamic backgrounds and also finding a way to detect a variable number of animals within a given sequence.

Acknowledgments

This work was supported in part by the National Science Foundation Grant DBI 10-62351.

References

[1] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. PAMI*, 33(3):500–513, 2011.

[2] T. Burghardt and J. Calic. Analysing animal behaviour in wildlife videos using face detection and tracking. *Vision, Image and Signal Processing, IEEE Proceedings*, 153(3):305 – 312, June 2006.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *CoRR*, abs/0912.3599, 2009.

[4] L. Gamble, S. Ravela, and K. McGarigal. Multi-scale features for identifying individuals in large biological databases: an application of pattern recognition technology to the marbled salamander ambystoma opacum. *Journal of Applied Ecology*, 45(1):170–180, 2008.

[5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, Aug. 2007.

[6] M. Lahiri, C. Tantipathananandh, R. Warungu, D. I. Rubenstein, and T. Y. Berger-Wolf. Biometric animal databases from field photographs: identification of individual zebra in the wild. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 6:1–6:8, New York, NY, USA, 2011. ACM.

[7] Z. Lin, M. Chen, L. Wu, and Y. Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *Mathematical Programming*, 2010.

[8] D. Ramanan, D. Forsyth, and K. Barnard. Building models of animals from video. *PAMI*, 28(8):1319–1334, August 2006.

[9] J. M. Rowcliffe and C. Carbone. Surveys using camera traps: are we looking to a brighter future? *Animal Conservation*, 11(3):185–186, 2008.

[10] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 28 – 31 Vol.2, aug. 2004.