

FINDING AREAS OF MOTION IN CAMERA TRAP IMAGES

Agnieszka Miguel^{*}, Sara Beery^{**#}, Erica Flores^{*}, Loren Klemesrud^{**#}, Rana Bayrakcismith^{**}

Seattle University
Departments of ^{*}Electrical and Computer Engineering and [#]Mathematics
902 12th Ave., Seattle, WA 98122

^{**}Panthera
8 West 40th Street, 18th Floor, New York, NY 10018

ABSTRACT

Camera trapping is used by conservation biologists to study snow leopards. In this research, we introduce techniques that find motion in camera trap images. Images are grouped into sets and a common background image is computed for each set. The background and superpixel-based features are then used to segment each image into objects that correspond to motion. The proposed methods are robust to changes in illumination due to time of day or the presence of camera flash.

Index Terms— Camera Trapping, Conservation Biology

1. INTRODUCTION

Snow leopards (*Panthera uncia*) are listed as endangered on the International Union for the Conservation of Nature Red List of Threatened Species [1]. Their range of more than 2 million km² spans 12 countries in Central Asia. Snow leopards are very elusive and seldom seen by people. In 2003, scientists estimated that there are between 4,500 and 7,350 snow leopards in the wild, although this estimate may be too low because as population studies have expanded over the past decade, scientists are often finding more cats than expected [2]. Conservation activities include methods to better understand this species. Researchers use "camera traps" by placing cameras in remote areas inhabited by snow leopards. Such cameras capture photographs when a source of heat passes in front of them. These pictures are then used to recognize specific cats in order to track populations and movement patterns. Because each snow leopard has a unique coat, snow leopards are identified based on the characteristics of their spot patterns such as their size, shape, orientation, and coloration [3].

Files from camera traps are collected by conservation biologists who study the location and behavior of snow leopards. Their first task is to sort the images into sets with snow leopards and those with other animals. Currently, they have to do it manually, which takes up time that could be spent on more advanced data analysis. The work presented here is a first step in creating an autonomous method of sorting the camera trap images. Further research will analyze the sorted image sets to determine which images contain snow leopards.

The funding for this work has been provided by Center for Environmental Justice and Sustainability at Seattle University (SU), Clare Booth Luce Scholar Foundation, SU Summer Research Fellowship Program, and Francis P. Wood Chair in Electrical and Computer Engineering at SU. Thank you to Panthera, SLT, and SLCF Mongolia for providing the camera trap images used in this research.

This paper presents techniques to find areas of motion within sets of camera trap images. Images are grouped into sets and a common background image is computed for each set. The background and superpixel-based features are then used to segment each image into objects that correspond to motion. The motion template images are then post-processed using morphological operations.

2. PRIOR WORK

There are many different methods of unsupervised image segmentation. We are interested predominantly in those that have been applied to the segmentation of camera trap images. Most notably, Reddy et.al. [4] use a method based on both texture and color features in order to segment tigers in camera trap images. Their method is based on multi-level nonlinear diffusion as described in [5]. In [6], Zhelezniakov et. al. propose a unsupervised two-step segmentation method. The first step uses the method of superpixel classification described in [7] and [8], then texture analysis of each superpixel is input into a Support Vector Machine in order to classify that pixel as belonging to the seal or background.

The image data used in these papers are characterized by greater differentiation in animal texture and pixel intensity from the surrounding environment than in the case of snow leopard images. The methods also do not take into account potential information provided by the continuity of the background in each image. Our method proposes to use the fact that the images in our data are taken in sets after each trigger of the motion-sensor to recognize areas of motion in each image. This information will add to that provided by texture characteristics, in order to improve segmentation results when working with very well-camouflaged animals in grayscale images.

3. DATA

A camera trap is a remotely-triggered camera that is used as a non-invasive method to determine population data for a species difficult to otherwise quantify in the wild. Camera traps are invaluable in analyzing snow leopard populations, as the cats are notoriously elusive and seldom seen by researchers. The cameras are placed in known snow leopard habitats, and use either a motion or infrared sensors to trigger the shutter. To take pictures at night, some types of cameras use white or incandescent flash while others record pictures with only an infra-red light [3].

The camera trap images used in this research were provided by Panthera, a nonprofit wild cat conservation organization. The images were taken during a three year period (2009-2011) in Tost Mountains, South Gobi, Mongolia. The image dimensions are 1280 x

1024 pixels. There were between 11,000 to over 27,000 files produced each year. The images are stored in folders that correspond to different camera locations. Cameras were placed in 40-41 locations each year. The cameras used to take these images were the RECONYX RapidFire Professional Digital Infrared Cameras.

The cameras were programmed to take a sequence of five photos after the sensor is triggered. The cameras can be triggered by not only a snow leopard but by any other animal that passes in front of the sensor, as well as sudden weather changes, or wind moving the vegetation. This results in many images in each data folder that do not contain snow leopards. Fig. 1 shows examples of images used in this research. There are three daytime images of a snow leopard that are part of a sequence of 10 images taken within a very short period of time. The snow leopard moves slowly across the camera’s field of view (Fig. 1(a)-(c)). We also show two examples of night images. In the snow leopard image (Fig. 1(d)), the flash illuminated mainly the body of the animal. In the fox image (Fig. 1(f)), both the ground and the fox have been illuminated by the flash. Finally, Fig. 1(e) shows an example of a daytime image of an ibex which is a part of a sequence of 45 images.

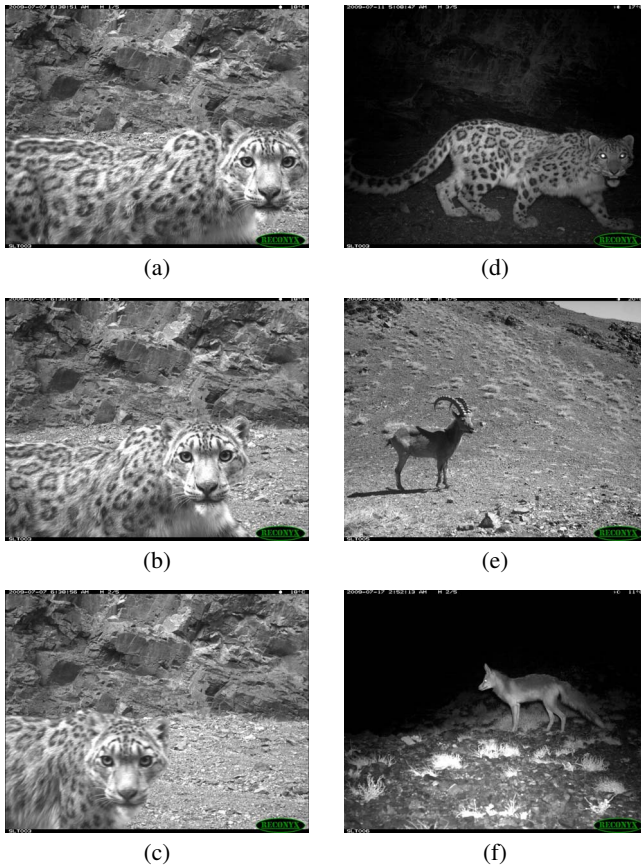


Fig. 1. Examples of images used in our research. (a)-(c) Snow leopard images that are part of a sequence of 10 images. (d) Snow leopard in a night picture. (e) Ibex. (f) Fox in a night picture.

4. PROPOSED MOTION DETECTION METHOD

To locate the areas of motion in each image, we use concepts from background subtraction [9]. Traditional background subtraction

methods used to separate background from foreground in video sequences perform well when the background is static, for example indoors where the source of illumination does not vary. In the outdoor environments of camera trap images, changing weather (sun, rain, and wind) makes these methods very challenging to use. To account for the environment-related changes in the background, our first step is to sort all images from one location into *sets*. For each set, we then compute a common background image. Following, we use one of three methods to find the location of motion in each image (feature thresholding, k-means, and fuzzy k-means clustering).

4.1. Sets of Images

We determine which images should be grouped into a single set for background computation by comparing the conditions under which each photo was taken. We then group daytime images into sets according to the time when these images were taken. Initially, all daytime images that were taken within 90 seconds of each other belong to the same set. Following, we split daytime image sets using information about the ISO and exposure settings of the camera. We assume that a change in camera settings results in significantly different image that should be part of a different set. We group nighttime images into sets based only on the camera settings. To reduce the run time of our background computation, no sets can have more than 30 images. The minimum number of images in a set is 3.

4.2. Computing Backgrounds

We compute a background image for each set of images using median filtering which was shown to be very robust compared to higher complexity methods. The background image B_k for set k containing K images is defined as

$$B_k(x, y) = \text{median}\{I_1(x, y), I_2(x, y), \dots, I_i(x, y), \dots, I_K(x, y)\}.$$

Fig. 2 shows examples of backgrounds computed for sets of camera trap images. The first background, computed for a set of 17 nighttime images (Fig. 2a), is clear of foreground objects. The second background, computed for a set of 10 daytime images (Fig. 2b), contains a faint outline of an animal which illustrates how it is sometimes challenging for the median filter to deal with slow motion.



Fig. 2. Examples of background images computed from a sequence of (a) 17 nighttime images and (b) 10 daytime images.

4.3. Motion Estimation

Background subtraction in which each image is subtracted from its corresponding background is a typical approach to obtain the foreground mask. However, for snow leopard camera trap images, such approach proves to be inadequate and leads to very noisy results

where large sections of motion are missing. Instead, we use the concept of superpixels and image features.

4.3.1. Superpixels

Superpixels were developed as an alternative to the traditional pixel grid. Superpixel algorithms group pixels into perceptually meaningful regions. In this work, we use the SLICO algorithm to form 10,000 superpixels in our camera trap images [10]. We found that this large number of superpixels, though computationally expensive, results in more accurate final motion masks.

4.3.2. Features

We use several types of superpixel-based texture features. Assume that image $I_i(x, y)$ has been segmented into N superpixels. In this work, $N = 10000$. Subscript i indicates that image $I_i(x, y)$ is the i th image in set k . Its corresponding background image is $B_k(x, y)$. Let $S^n(I_i)$ be the set of all pixel coordinates that belong to the n th superpixel computed for image $I_i(x, y)$. We then define $I_i^n(x, y)$ as the segment of image $I_i(x, y)$ that corresponds to the n th superpixel:

$$I_i^n(x, y) = \{I_i(x, y) : (x, y) \in S^n(I_i)\}$$

and

$$I_i(x, y) = \bigcup_n I_i^n(x, y).$$

Next, we define the operation of creating a new image by assigning constant values to all pixels in each superpixel. Let

$$m = \{m(1), m(2), \dots, m(n), \dots, m(N)\}$$

be a vector of length N and

$$I_i^n(x, y) \leftarrow m(n)$$

correspond to replacing all pixel values in $I_i^n(x, y)$ with $m(n)$. We can create a new image with N values as

$$M_i(x, y) = \bigcup_n \{I_i^n(x, y) \leftarrow m(n)\}.$$

The *Mean of Differences* motion feature is computed as follows:

$$\text{MoD} \{I_i(x, y)\} = \bigcup_n \left\{ I_i^n(x, y) \leftarrow \frac{1}{|S^n(I_i)|} \times \sum_{(x, y) \in S^n(I_i)} |B_k(x, y) - I_i(x, y)| \right\},$$

where $|S^n(I_i)|$ is the number of pixels in superpixel n .

The *Difference of Means* motion feature is defined as:

$$\text{DoM} \{I_i(x, y)\} = \bigcup_n \left\{ I_i^n(x, y) \leftarrow \frac{1}{|S^n(I_i)|} \times \left| \sum_{(x, y) \in S^n(I_i)} B_k(x, y) - \sum_{(x, y) \in S^n(I_i)} I_i(x, y) \right| \right\}$$

In addition to MoD and DoM, we also use mean, range, and median of superpixels as our features:

$$\text{Mean} \{I_i(x, y)\} = \bigcup_n \left\{ I_i^n(x, y) \leftarrow \frac{1}{|S^n(I_i)|} \times \sum_{(x, y) \in S^n(I_i)} I_i(x, y) \right\}$$

$$\text{Range} \{I_i(x, y)\} =$$

$$\bigcup_n \left\{ I_i^n(x, y) \leftarrow \left(\max_{(x, y) \in S^n(I_i)} I_i(x, y) - \min_{(x, y) \in S^n(I_i)} I_i(x, y) \right) \right\}$$

$$\text{Median} \{I_i(x, y)\} =$$

$$\bigcup_n \left\{ I_i^n(x, y) \leftarrow \text{median}_{(x, y) \in S^n(I_i)} I_i(x, y) \right\}$$

4.4. Feature Thresholding

Our first method of finding motion in camera trap images computes MoD and DoM features for each image. Then, for each pixel, motion decision is made according to the values of MoD and DoM. If one or both of these features are greater than their corresponding threshold, the pixel is classified as a motion pixel. If both features are lower than their respective thresholds, the pixel is classified as a background. The thresholds are found empirically for each given camera trap data set.

4.5. K-means Clustering

Our second method of finding motion uses k-means clustering. For daytime images, our feature set includes only the DoM and the MoD. For nighttime images, we increase our feature set by adding the range, mean, and median. These metrics are used as features for the k-means clustering algorithm, and each superpixel is classified as belonging to one of two clusters (foreground or motion and background).

4.6. Fuzzy K-means Clustering

Our third method of finding motion uses fuzzy k-means classification. We use the same features as in the case of traditional k-means clustering above. Fuzzy k-means classification results in the probability that each superpixel belongs to motion within the image. We chose which classified portion belongs to the foreground by taking the sum of the confidence values for each label over all superpixel regions. The label with larger sum of confidence values is considered the background, and the smaller set is considered the foreground. We then threshold the fuzzy k-means results based on a percentage of the average confidence value of the animal label across the entire image.

4.7. Method Selection and Postprocessing

We process each image using the three methods described above. To determine which motion detection result is preferred, we assume that the desired motion template should contain a small number of smooth objects and should be noise-free. Therefore, we count the number of objects in the template, analyze the smoothness of the chain code that describes their boundaries, and compute a frequency-based measure of noise in the template. To improve the results from our classification algorithms, we use morphological operations to fill all holes smaller than the size of five superpixels and remove all binary objects smaller than the size of five superpixels.

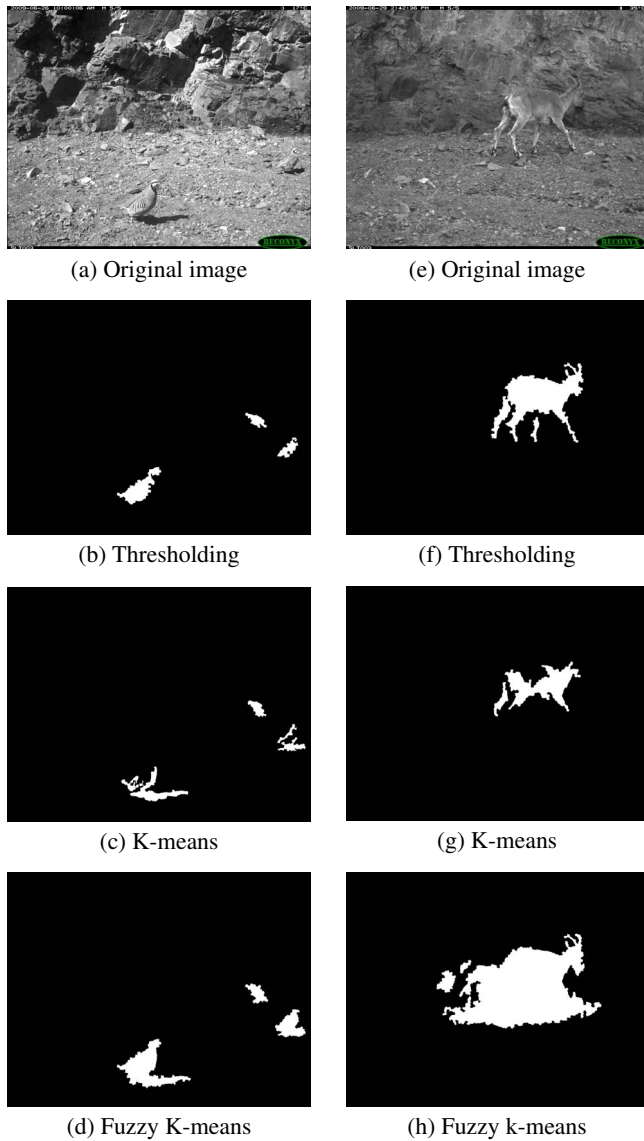


Fig. 3. Results for daytime birds and ibex images.

5. RESULTS

We present examples of our promising results in Figures 3 and 4. For each original image, shown in Fig. 3(a, e) and Fig. 4(a, e), we apply the three proposed methods and show results in the corresponding rows of these two figures. No one method produces the best results for all types of images. In the case of the birds and the ibex images, the feature thresholding method produces superior results. Fuzzy k-means method gives the best results for the daytime snow leopard image while the traditional k-means technique is best in the case of the nighttime snow leopard image. This variability is present across all camera trap images and underscores the need to explore different segmentation methods for this particular application.

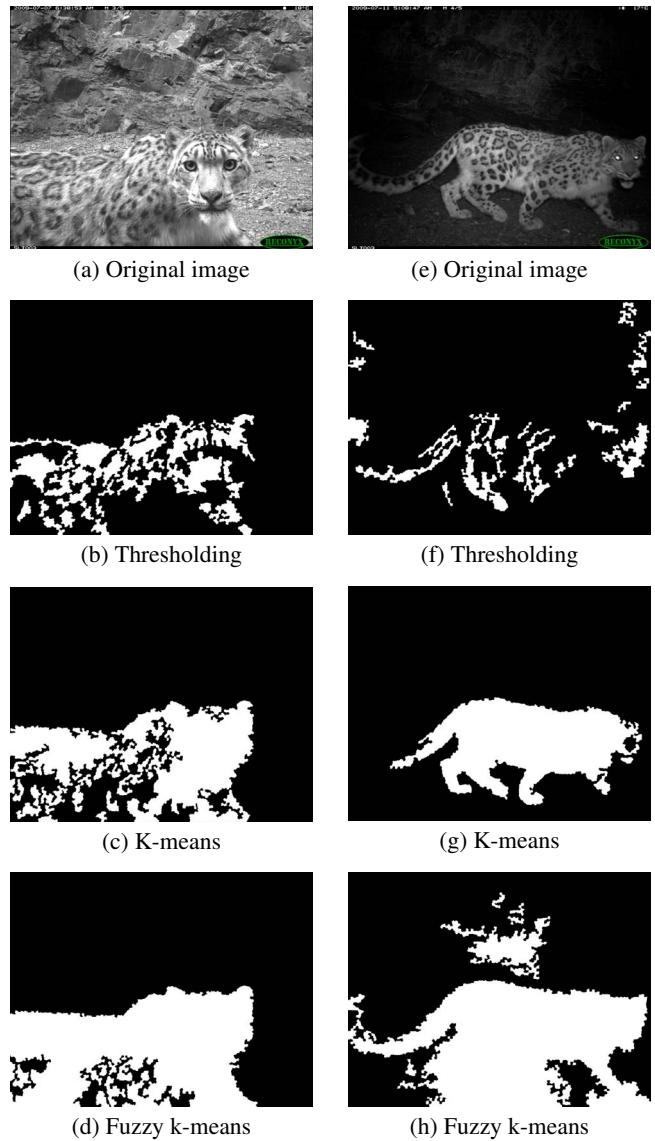


Fig. 4. Results for daytime and nighttime snow leopard images.

6. CONCLUSIONS AND FUTURE WORK

The methods described here find motion in images from camera traps set to capture snow leopards. By using three different approaches, we produce results that are robust with respect to changes in illumination due to weather and time of day. This research will open the door to image classification based on type of animal sighted, and eventually to individual snow leopard recognition.

7. ACKNOWLEDGEMENTS

Photos used in this research were taken in a project undertaken by Panthera, Snow Leopard Conservation Foundation in Mongolia, and the Snow Leopard Trust. We acknowledge the following students who briefly worked on this project: E. Sblendorio, K. Sherpa, and N. Wichers. We also thank Dr. McLean Slaughter for his input regarding the next stage of this work.

8. REFERENCES

- [1] International Union for the Conservation of Nature, “The IUCN red list of threatened species. Version 2015-14.,” <http://www.iucnredlist.org>, Accessed: 2016-01-08.
- [2] Thomas McCarthy and Guillaume Chapron, “Snow leopard survival strategy,” Seattle, USA: ISLT and SLN, 2003.
- [3] R. M. Jackson, J. D. Roe, R. Wangchuk, and D. O. Hunter, “Estimating snow leopard population abundance using photography and capture-recapture techniques,” *Wildlife Society Bulletin*, vol. 34, no. 3, pp. 772–781, 2006.
- [4] K Reddy and R Aravind, “Segmentation of camera-trap tiger images based on texture and color features,” in *Communications (NCC), 2012 National Conference on*. IEEE, 2012, pp. 1–5.
- [5] Mikael Rousso, Thomas Brox, and Rachid Deriche, “Active unsupervised texture segmentation on a diffusion based feature space,” in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*. IEEE, 2003, vol. 2, pp. II–699.
- [6] Artem Zhelezniakov, Tuomas Eerola, Meeri Koivuniemi, Miina Auttila, Riikka Levänen, Marja Niemi, Mervi Kunasranta, and Heikki Kälviäinen, “Segmentation of saimaa ringed seals for identification purposes,” in *Advances in Visual Computing*, pp. 227–236. Springer, 2015.
- [7] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jagannath Malik, “From contours to regions: An empirical evaluation,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2294–2301.
- [8] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, “Contour detection and hierarchical image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011.
- [9] H. Hassanpour, M. Sedighi, and A. Manashty, “Video frames background modeling: Reviewing the techniques,” *Journal of Signal and Information Processing*, vol. 2, no. 2, pp. 72–78, 2011.
- [10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, 2012.