Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Jens Wawerla · Shelley Marshall · Greg Mori ·

Kristina Rothley · Payam Sabzmeydani

BearCam: Automated Wildlife Monitoring At The Arctic Circle

Received: date / Accepted: date

Abstract In this paper we describe the development of the BearCam, a camera system which was deployed in Fall 2005 to monitor the behaviour of grizzly bears at a remote location near the arctic circle. The system aided biologists in collecting the data for their study on bears' behavioural responses to ecotourists. We developed a camera system for operating in the challenging arctic conditions. We describe a novel "motion shapelet" algorithm for automatically detecting bears in the video captured by this camera system. This algorithm is an extension of the shapelet features Sabzmeydani and Mori (2007), which are mid-level features capturing pieces of shape. Our extension of this technique incorporates motion information and proves effective at automatically detecting the occurrence of bears. We present quantitative results demonstrating that our algorithm can reliably detect bears in the vast amounts of video footage collected by our system.

Keywords object recognition · surveillance · video analysis · boosting · wildlife monitoring

Jens Wawerla, Greg Mori, Payam Sabzmeydani

School of Computing Science

Simon Fraser University

Burnaby, British Columbia, Canada

E-mail: {jwawerla,psabzmey,mori}@sfu.ca

Shelley Marshall, Kristina Rothley

School of Resource and Environmental Management Simon Fraser University

Burnaby, British Columbia, Canada

1 Introduction

Wildlife-based ecotourism is rapidly increasing in popularity, especially when featuring large mammals in their natural environment (Jelinski et al., 2002; Dyck and Baydack, 2004; Nevin and Gilbert, 2005) Wildlife viewing was once considered a non-consumptive human activity with little or no impact on animals. However, recent research has invalidated this assumption and instead revealed how wildlife-viewing activities have negatively affected animals, e.g. (Duchesne et al., 2000; Crupi, 2003; Walker et al., 2006; Olson and Gilbert, 1994; Chi and Gilbert, 1999; Nevin and Gilbert, 2005). In some instances, ecotourism has benefited animals (e.g., female grizzly bears with young at Glendale Cove in southwest British Columbia; Nevin and Gilbert (2005)). However, these positive effects are site-specific, not wide ranging, and not shared by all individual animals at these sites. Thus, monitoring of human impacts on wildlife populations that are the focus of ecotourism activities is necessary to ensure that the health of individual animals and populations is not sacrificed for the economic, social, and educational gains of ecotourism.

Wildlife managers use a variety of techniques that vary in their efficiency and effectiveness to monitor wildlife around ecotourism sites. The traditional techniques for population-scale data collection, such as mark-recapture or aerial counts, are labour intensive and extremely costly. As an alternative or complement to these methods, the use of camera systems, which collect information largely in the absence of human operators, is increasing in popularity (Roberts et al., 2006). Cameras have the additional advantage that their means of data collection is less intrusive or disruptive to the animals being monitored. For example, if animals' behavioural responses to ecotourists are of interest, a common data collection technique is to use an on-site observer to perform the necessary close-range observations. These observers face two problems. First, it may be impossible to collect the data that represents animal behaviour in "ecotourist absence" if the observer's presence generates a similar behavioural response as an ecotourist. Second, topography and vegetation at a site may severely limit the observer's ability to see the focal animals. Two techniques to resolve the topography issue are the employment of additional observers stationed at strategic locations (e.g., Olson and Gilbert (1994); Olson et al. (1998); Crupi (2003)) or to divide the single observer's time between various areas (e.g., Pitts (2001)). Use of additional observers increases human activity in the area which can confound any

Fig. 1 The BearCam camera deployed for viewing grizzly bears in the Yukon.

human impact monitoring being undertaken. The latter option decreases observer time spent in each area, which reduces sample sizes making signal detection and analysis more challenging. Therefore, it is advantageous to use camera systems to remotely collect data, without interference from human observers. However, cameras generate large amounts of data, which are typically sorted manually to collect the required data. As computer vision researchers, there is a great opportunity to aid natural scientists by automating parts of the video analysis process.

In this paper, we describe the development of the BearCam (pictured in Figure 1), a camera system deployed in Fall 2005 to monitor the behaviour of grizzly bears at the Ni'iinlii Njik (Fishing Branch) Park. The system aids biologists monitoring grizzly bear behaviour at a new ecotourist destination, a bear viewing site, in the Ni'iinlii Njik Park. The main objective of the biologists' study was to assess whether ecotourists negatively affected grizzly bear feeding behaviour at this salmon spawning stream, which could potentially reduce their survival and reproductive success (Hilderbrand et al., 1999). This camera system served two purposes: 1) to increase the observation area without additional observers or without reducing the researcher's time at the primary observation area, and 2) to record bear behaviour in an area of minimal ecotourist activity without requiring the researcher's physical presence, which would effectively render this no longer a minimal human activity area.

This park is a remote wilderness area in the Yukon, Canada, just below the arctic circle. The only man-made structures are a handful of wooden huts without running water and electricity is provided

by a small gas-powered generator, which can be operated for only about 4 hours a day due to the expense of airlifting gasoline into the park. Lights in the huts are propane powered. The propane, gasoline and the remaining equipment must be airlifted into the study site since the closest road is approximately 30 helicopter flight minutes away. The nearest settlement is about 100km way.

One faces numerous challenges while designing and building a camera system to operate in this remote environmentally protected, arctic site in the cold fall weather, including:

- Severe weather conditions (rain, snow, wind)
- Energy is sparse and expensive to generate
- Temperatures of $-15°C$ exceed the specified range of off-the-shelf electronic components and require components with an extended temperature range
- The system has to be air-portable, which constrains size and weight
- Minimize impact of the system on the habitat and animals in the area
- Storage and processing of large amounts of video data.

The main contribution of this work is developing a system to aid biologists in data collection in Ni'iinlii Njik Park. We describe the development of a camera system for operating in these challenging conditions. We also develop a novel "motion shapelet" algorithm for automatically detecting bears in the video captured by this camera system. This algorithm is an extension of the shapelet features described in Sabzmeydani and Mori (2007). These shapelets are mid-level features, capturing pieces of shape which are more descriptive than the small-scale local features used in other algorithms. These shapelets have been applied to the related task of detecting pedestrians in still images[1], and here we extend them for use in detecting bears in videos.

In this paper we review related work in Section 2, provide an overview of our system in Section 3, detail the recording system in Section 4, outline the bear detection algorithm in Section 5, present experimental results from video collected in Fall 2005 in Section 6, and conclude in Section 7.

[1] There was an error in the experiments reported in Sabzmeydani and Mori (2007). In particular, this method does not outperform the HOG method of Dalal and Triggs (2005).

2 Related Work

2.1 Wildlife Monitoring Data Collection in Biology

Use of camera systems in wildlife monitoring is an option that allows data collection largely in human absence (ecotourist or observer). Cameras can also increase the size of the area sampled without requiring additional observers or the attenuation of a single observer's field time. The difficulty is then in finding the interesting events (animal presence) in the volume of video data collected. Often, remote triggers (e.g. motion or heat) are used to activate the camera. MacHutchon et al. (1998) and Johnson et al. (2006) used triggered still cameras to record bear activity in areas of varying human density and tiger activity, respectively. Their simplicity makes these cameras attractive but the number of false positives is quite high because the trigger cannot distinguish between the target animal, other animals, or motion artifacts. In addition, analyzing animal behaviour from just one still image is extremely challenging.

Song and Goldberg (2006) built a camera system to assist the search for the Ivory Billed Woodpecker, which is believed to be extinct. Their system was deployed in Fall 2006 in the Bayou DeView wildlife refuge in Arkansas, USA. The cameras and the hard-drives are mounted on a power line pole, which provides all the electric energy required for the system. Full details are not available, but video frames are automatically analyzed using motion detection.

Every summer National Geographic operates a 'webcam' allowing Internet users to view Alaskan grizzly bears from the comfort of their living room (Geographic, 2006b). SeeMore Wildlife Systems [2], a commercial provider of remote wildlife viewing systems, built this system. The video signal is transmitted via multiple microwave links and made publicly accessible by National Geographic. We were unable to find information on this system's power source. National Geographic operates a second remote viewing system to allow Internet users to view seals about 90 km south of San Francisco (Geographic, 2006a). Technical details are not available.

¿From September 2002 to September 2005 Dodd et al. (2007) operated four low-lux black-and-white cameras at two bridged wildlife highway underpasses in Arizona, USA, to assess elk use of

[2] www.seemorewildlife.com

these underpasses. They used infrared illuminators to ensure images had sufficient light, powered their system from the grid and buried power supply wires and data-lines. To cope with the vast amount of potential video data, break beams triggered recording.

Our recording system differs from those mentioned previously because (1) the remote location requires power grid independence and economical use of energy, (2) the protected area status limits instrument options such as long cables, large microwave transmitters and similar equipment are not viable options and (3) the system has automatic video processing to detect video scenes of interest to biologists.

2.2 Related "Looking at People" Work

Our system automatically detects bears within the captured video, which is essentially identical to many surveillance tasks involving human subjects. The computer vision community has substantial amounts of work on this problem and Gavrila (1999); Moeslund and Granum (2001) summarize this work.

We based our bear detection algorithm on our previous work on detecting pedestrians in still images (Sabzmeydani and Mori, 2007). Pedestrians have also been detected using a combination of parts - legs, torso, and arms (Mohan et al., 2001), where individual Support Vector Machine (SVM) detectors are trained for each part and their outputs are combined into a final classifier after applying geometric constraints. Gavrila and Philomin (1999) and Felzenszwalb (2001) compare edge maps of pedestrian templates to edges found in images. Gavrila and Philomin use the Chamfer distance, and develop an efficient hierarchical system. Felzenszwalb uses a generalization of the Hausdorff distance, and learns pedestrian templates from training images. Viola et al. (2003) compute spatial and temporal rectangle filters efficiently using the integral image technique, and learn a pedestrian classifier using a variant of AdaBoost. Mikolajczyk et al. (2004) use a part-based detector to detect humans. They model humans as assemblies of parts that are represented by SIFT-like orientation base features. Feature selection and the part detectors are learned using AdaBoost. Fink and Perona (2004) describe the MutualBoost algorithm for building a combined detector from a set of pre-defined and labeled object parts.

Leibe et al. (2005) start with a local feature detection to generate a set of pedestrian hypotheses. By using a training set which contains foreground masks for pedestrians, segmentation masks are computed

for these hypotheses. A top-down verification step, using these segmentation masks and Chamfer matching is then applied. Impressive results on images with substantial overlap of pedestrians are presented. Wu and Nevatia (2005) also address the problem of overlapping pedestrians and formulate a joint likelihood that is optimized using a greedy algorithm. Dalal and Triggs (2005) use Histogram of Oriented Gradient (HOG) descriptors and SVMs for building a pedestrian detector. By tuning all the parameters of their HOG features, they compare the use of a variety of feature configurations to find the best configuration for pedestrian detection, on a challenging dataset of human figures. Dalal et al. (2006) extend this line of work to include motion features. Munder and Gavrila (2006) studied the problem of pedestrian classification with different features and classifiers. They find that local receptive fields can do a better job in representing pedestrians and also SVMs and AdaBoost classifiers outperform the other classifiers tested.

Other approaches use video sequences and apply background subtraction to reduce clutter. The Pfinder system of Wren et al. (1997) is an early example of such an approach. Zhao and Nevatia (2003) work on difficult scenes involving large crowds of people, and segment foreground blobs into individual people using a Markov chain Monte Carlo technique. Elgammal and Davis (2001) handle overlapping people by segmenting foreground regions according to a colour model that is initialized when the people are unoccluded.

2.3 Computer Vision for Biology Data Collection

Other computer vision researchers have also developed algorithms for biological research. Khan et al. (2005) developed methods for tracking multiple ants, and suggest the use of Hidden Markov Models for analyzing their behaviours (Balch et al., 2001). Khan et al. (2004) also developed a Rao-Blackwellized Particle Filter for tracking bees with complex appearance changes in cluttered environments. Dollar et al. (2005) and Belongie et al. (2005) analyze the behaviour of mice in caged environments by first tracking and then computing spatio-temporal patch features. Betke et al. (2007) develop data association techniques for large numbers of objects, and apply them to tracking bats viewed in infra-red video.

The BearCam is part of the Scientific Data Acquisition, Transportation and Storage (SDATS) project, with the aim of reducing the expense and manual labour involved in gathering scientific data

in the field. Other SDATS projects include (1) stereo tracking of grasshoppers in cages and attempting to recognize their actions by analyzing their 3D movement (Naeini et al., 2007) and (2) developing a method for automatically determining the species of fish from images collected with an underwater video camera (Rova et al., 2007).

3 System Overview

The biological research objective is to assess whether ecotourists negatively affect grizzly bear feeding behaviour. To this end, the biologists require data collection at a variety of resolutions. At the coarsest level, they must determine the amount of time bears are active at the study site. Next, they desire a measure of the time bears spend performing various behaviours (e.g., walking, sitting, foraging). Finally, at the most detailed level, they must estimate the caloric intake of individual bears. This estimate typically involves counting the number of fish caught by bear and the proportion of each fish consumed.

As argued above, there are numerous drawbacks to the typical method of acquiring these data, namely using a human observer stationed on site. In particular, for our site, a map of which is shown in Figure 2(a), stationing human observers at the required viewpoint would be non-trivial, and certainly intrusive. In order to address this problem, we have constructed a remote monitoring system that can be used by a human observer to efficiently gather these data off-site. Our system consists of two main components. The first is a video recording system, which records video from the monitoring site noted in Figure 2(a). Given the volume of data acquired by this recording system, manually searching through the video to find the occurrences of bears would be a time-consuming, error-prone process. Instead, we have developed algorithms for automatically detecting the bears in the captured video sequences. A user can then manually collect the required data regarding activity levels, behaviours, and estimate caloric intake by inspecting only the relevant clips. In the following sections we describe each of these two components in detail.

4 Video Recording System

The video recording system consists of two main components. One is the remotely deployed camera and the other is the base station located in the main hut of the wildlife viewing area. Figure 3 shows

(a) (b)

Fig. 2 (a) Bear viewing site Ni'iinlii Njik (Fishing Branch) Park. Being at any one viewing site, it is impossible to monitor bears at the othersides due to trees in the line of sight and due to distance between the sites. (b) Camera and battery pack mounted on a tree

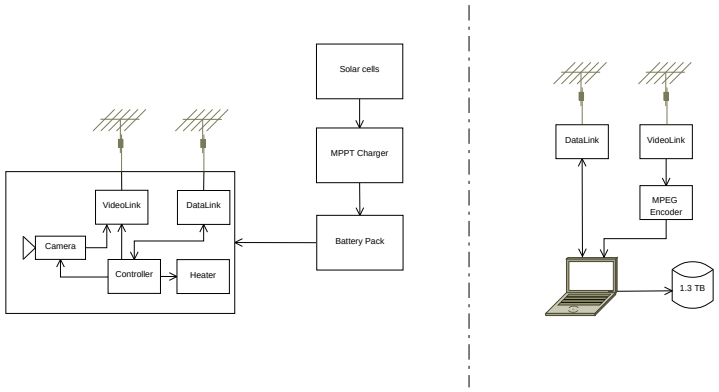


Fig. 3 System overview, left: solar powered camera system. Right: base station to control the cameras and store the mpeg encoded video

an overview and depicts the subcomponents of the system. For clarity we only show one camera, but we actually built four camera systems and deployed one in 2005 and two in 2006 in the Yukon.

The operator controls (enables or disables) and monitors each camera from the base station. Once enabled a camera sends the video signal via a 2.4 GHz radio signal to the base station. Here the video is MPEG-2 encoded by an off-the-shelf hardware encoder [3] and then stored on portable hard drives. Each of the six hard drives holds 230 GB of data, which allows us to store about 600h of video in total.

In the base configuration each camera is battery powered. One battery can power a camera for at least 24h, depending on the ambient temperature 48h or longer. To be on the safe side each camera has two battery packs, which we swapped every day. This allows one pack to recharge while the camera is

[3] WinTV-PVR-USB2 from Hauppauge, http://www.hauppauge.com

still powered by the other. Electricity for recharging and operating the base station was provided by a small gasoline powered mobile generator.

The heart of the base station is a laptop that runs custom-made software to control the cameras and monitor their state, that is battery level, temperature, state of video camera and heater. In addition off-the-shelf software handles storing compressed video data on hard drives.

4.1 Camera

A wildlife monitoring camera for the arctic circle (as shown in Fig. 2(b)) must be able to withstand rain, snow, wind, intrusion attempts from noisy animals, deal with an extended temperature range between $-15°C$ to $30°C$, handle condensation, be affordable, light and small enough to be airlifted as well as operatable by biologist without interfering with their research.

Due to cost considerations we used consumer products whenever possible. If those were not available or not rated for our purposes we devised our own components.

We choose WV-CP484 (wvc) cameras from Panasonic, housed in EH4718 (Pelco, 2003) enclosures for weatherproofing. These cameras are intended for outdoor surveillance applications and therefore handle varying light conditions well, are designed for longterm continuous use, and a variety of lenses are readily available. With a specified temperature range from $-10°C$ to $+50°C$, these cameras almost meet our requirements.

A set of electrical heating elements were integrated into the enclosure. These heaters kept the temperature inside the enclosure above the minimum operational temperature of all components and eliminated any build-up of condensation or frost on either the camera lens or the glass window of the housing. The two foil based heating elements have a power of 12W each, which was enough to keep the inside temperature well above freezing.

To regulate the temperature and remotely operate the video camera, we custom built a small microcontroller board. This controller is equipped with a thermometer to measure the in-house temperature, furthermore it has a 900 MHz radio modem (MaxStream, 2005). During camera operation the controller regulates the temperature, monitors the supply voltage, protects the batteries from over discharge and sends video data to the base station. The controller also regulates the voltage for the

camera and the video link. While the camera is off-line, the controller is in an energy saving mode waiting for commands from the base station. All parts used for the controller, including the radio, are rated up to $-40°C$. Therefore the control does not require the energy expensive heater, thus allowing it to be in stand-by for several days on one battery charge.

Video data are wirelessly transmitted to the base station because lack of power does not permit video storage at the camera location. An off-the-shelf 3 watt video transmitter operating at 2.4 GHz is used here. Although the distance between cameras and base stations is relatively small, less than 500m, transmitting the video signal in the forested wildlife refuge required mounting the receiver antenna well above the tree tops.

The final component, the battery packs, powers the whole system. We use a standard 12V 33Ah lead acid battery (Panasonic, 2003). Although this is not the ideal battery for our needs, we choose it with cost in mind. For example Streeter (2005) used a battery for his Antarctica robot that is much more suitable for cold environments. The main problem with our battery is that the capacity drops to 65% at $-15°C$. A simple solution is to select a battery with enough capacity even at low temperatures. To waterproof and protect the battery from impacts, we modified second hand, metal ammunition boxes and used them as housing for the batteries.

We charge the batteries using standard car battery chargers, each capable of recharging one battery at 10A. Fast charging was required since the availability of electricity was strongly rationed. Due to fuel limitations the operational time of the generator was about 4-5h per day.

4.2 Solar Power

¿From an engineering point of view, energy is one of the main challenges in this project, the other being temperature. From the first deployment in 2005, we learned that exchanging the batteries and charging the batteries was inconvenient and time consuming for the biologist. The batteries needed to be at room temperature for charging and unexpectedly, it took approximately 48h to warm batteries to this temperature in the hut. Thus, we developed a solar energy extension for the most remote camera in 2006.

Solar power use had several problems:

- sun energy is limited in fall at the arctic circle because the days are short and the sun's elevation angle is small
- modelling or even estimating the available solar energy proved difficult since reliable weather or solar irradiation data for the Ni'iinlii Njik Park is not available
- testing a solar power generator under realistic conditions almost 17° south of the intended deployment site four months ahead of time is very difficult.

Because solar irradiation data for the Fishing Branch River was not available we used the National Renewable Energy Laboratory (NREL) 30 year average for Fairbanks, Alaska and Bettles, Alaska (International, 2004). Both cities are a close approximation in latitude and elevation but neither account for local weather patterns or geographical properties such as shadows cast by hills or trees. We estimated the average solar radiation for a fix angle solar panel at $3.3KWhm^{-2}$ in September, and $1.9KWhm^{-2}$ in October.

Operating the whole system, including the heater, for four hours per day requires 10 Ah per day at 12V. Assuming a battery efficiency of 80% and solar radiation of $2KWhm^{-2}$ the solar pannels must be able provide a peak current of 6.25 A. Therefore, we selected two 50W solar panels (Kyocera). Two panels instead of one were used due to the size limitation of the helicopter.

As mentioned earlier the battery used during the first deployment was not rechargeable in a cold state. We required a different battery for the solar system because it was intended to operate without maintenance for two months. A growing number of batteries rechargeable in low temperature environments are available mainly for defence and aerospace purposes. We choose a 42 Ah lead battery rechargeable at $-40°C$ (EnerSys, 2006). At $-15°C$ the capacity drops to about 70% of the nominal capacity at $25°C$, which still provides about three days of autonomy, meaning the system can be powered for three consecutive days without sunlight.

We designed, built and airlifted the solar powered camera to the Ni'iinlii Njik Park but due to low bear activity in 2006, we did not test it. It remains for a future research mission to determine if the solar system will generate enough electric energy in the high northern regions of Canada to power a grizzly bear monitoring camera.

5 Detecting Bears in Recorded Video

Our system monitors the river site for 4 hours per day. Biologists require enormous amounts of time to manually search these videos for bear activity. For this reason, we developed an algorithm for automatically detecting the presence of bears in the recorded video.

The algorithm we use is an extension of the shapelet features described in Sabzmeydani and Mori (2007). The shapelet features were designed for the task of detecting pedestrians in still images. We adopt this approach to the related task of detecting bears. We extend it to incorporate the additional information which is present in a video dataset, calling these *motion shapelet* features. In the following sections, we describe our use of features from video data in the motion shapelet learning algorithm.

5.1 Motion Shapelet Feature-based Detection

A major drawback in many object detection algorithms is the fixed set of feature descriptors they use. The problem with defining features before training the classifier is that there could be some discriminative information that is missed by those features. In the shapelet feature work (Sabzmeydani and Mori, 2007), a specific feature set is learned while using information about the object classes in building these features.

The approach is related to that of Viola and Jones (2001); Viola et al. (2003), who use AdaBoost for face and pedestrian classification, using Haar-like wavelet features. The features they use are low-level, and AdaBoost selects a subset of these features to form the final classifier. Wu and Nevatia (2005) use AdaBoost with a set of hard coded mid-level features, called "edgelets", as its weak classifiers. These edgelets are a set of pre-defined patterns of edges in different locations. Unlike the Haar-like features, they contain more information, but since they are fixed a priori they may not capture all the available information that is useful to discriminate between our object classes.

In the shapelet feature approach, a set of informative mid-level features are automatically learned, rather than hand-coded. These mid-level features, called *shapelet features*, are constructed from low-level features. Shapelet features which best discriminate between object and non-object classes are built. The AdaBoost algorithm is used as the core computational routine, using it once to build the

shapelet features, and again to build a final classifier from these shapelets. Our method directly follows the variant of AdaBoost developed by Viola and Jones (2001).

The training phase of our algorithm consists of three steps:

1. **Low-level Features:** The input to this step is labeled training image sequences. We compute two types of low-level feature on the input sequences. First, we extract the gradient responses of each image in different directions, and compute local average of these responses around each pixel. Second, we compute a background difference, comparing pixels values in each image to the estimated background computed from the entire sequence. These low-level features will be used to build more sophisticated mid-level features (motion shapelets).
2. **Motion Shapelet Features:** For each of a number of small sub-windows inside the detection window, we run Adaboost to select a subset of its low-level features to construct a mid-level motion shapelet feature. By only using the features inside each sub-window, we force AdaBoost to extract as much information as possible at local neighbourhoods of the image. This process will provide us with a motion shapelet feature for each sub-window. Each of these is intended to be more descriptive than the low-level features and discriminative regarding our object classes. Each motion shapelet feature consists of a combination of motion responses and gradients with different orientations and strengths at different locations within the sub-window.
3. **Final Classifier:** The motion shapelet features only describe local neighbourhoods of the image and therefore their individual classification power is still limited. By merging these features together we can combine the information from different parts of the image. In order to achieve this goal, we use AdaBoost for the second time to train our final classifier, using motion shapelet features as its input.

5.2 Low-level features

Many detection approaches capture local information as their lowest level features. Approaches include computing image gradients (Dalal and Triggs, 2005), computing wavelet coefficients (Schneiderman and Kanade, 2000), and applying simple rectangular filters spatially (Viola and Jones, 2001) or temporally (Viola et al., 2003) or more sophisticated features such as edgelets (Wu and Nevatia, 2005). In our

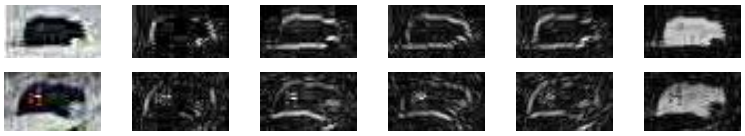


Fig. 4 Low-level features of two sample images. First column shows input image. Next four columns show gradients in each of four directions. Final column shows background difference.

work, we use background differences and gradient responses as our lowest level features. We use the absolute value of gradient responses, computed in four different directions. The absolute value is used because the sign of the gradient is uninformative due to varying background colours. To reduce the influence of small spatial shifts in the detection window, we locally average each of these cues by convolving the responses with a box filter:

$$S_d(x) = (|I * G_d| * B)(x) \text{ where } d \in \mathcal{D} \tag{1}$$

$$S_m(x) = (|I - I_b| * B)(x) \tag{2}$$

where $*$ denotes convolution, I is the intensity image. G_d is the gradient kernel (e.g. $[-1, 0, 1]$ or $[-1, 0, 1]^T$) that we use to get derivatives in direction $d \in \mathcal{D}$. I_b is the background image, computed by taking a median over a sampled set of the frames from the image sequence being analyzed. B is a 2-D box filter (e.g. a 5×5 matrix with all the elements $\frac{1}{25}$) used for averaging. $S_d(x)$ captures the amount of gradient at every pixel in direction d. $\mathcal{D}$ is the set of directions in which we are computing the gradients. In our experiments we use four directions; $\mathcal{D} = \{0°, 45°, 90°, 135°\}$. $\{S_d(x)\}$ and $S_m(x)$ are our final low-level features. Figure 4 shows two examples of bears and their low-level features.

The information captured about the classes by each of the low-level features is very little. If used as a classifier, each of these low-level features (an $S_d(x)$ or $S_m(x)$), can only separate our two classes (bear and background) slightly better than random classification. To make our features more informative, we use AdaBoost to combine them together to create more informative mid-level features, the motion shapelet features.

5.3 Motion Shapelet Features

We define a *motion shapelet feature* as a weighted combination of low-level features. Each low-level feature consists of a location, a direction or motion, and a strength. Each motion shapelet feature will

cover a small sub-window of the detection window, and its low-level features are chosen from that sub-window.

We will consider k sub-windows $w_i \in \mathcal{W}$, $i = 1, \ldots, k$ inside our detection window. Selecting the set of sub-windows $\mathcal{W}$ will be explained in detail in Section 6.1. We will build a separate mid-level motion shapelet feature for each sub-window w_i. To do this, we collect all the low-level features that are inside that sub-window $\{f_u^p = S_u(p) : p \in w_i, u \in \mathcal{D} \cup \{m\}\}$ and consider them as potential weak classifiers of an AdaBoost run.

In each iteration t of the AdaBoost (Viola and Jones, 2001) training algorithm, one of the features $f_t \in \{f_u^p\}$ is chosen as the feature of the weak classifier $h_t(x)$ to be added to the final classifier. This weak classifier is of the form:

$$h_t(x) = \begin{cases} 1 \text{ if } p_t f_t(x) < p_t \theta_t \\ 0 \quad \text{otherwise} \end{cases} \tag{3}$$

for an image detection window x, where $\theta_t \in (-\infty, \infty)$ is the classification threshold of the classifier and $p_t = \pm 1$ is a parity for the inequality sign.

AdaBoost calls the weak learner algorithm repeatedly in a series of rounds. The algorithm maintains a set of weights v_l over the training set. On each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set. In every iteration, the best $h_t(x)$ is added to the set of selected weak classifiers, where the best is that which minimizes has minimum error. The error ϵ_j of a weak classifier h_j is calculated as:

$$\epsilon_j = \sum_l v_l |h_j(x_l) - y_l| \tag{4}$$

This is a sum of the weights of examples x_l with labels $y_l \in \{0, 1\}$, mis-classified by weak classifier h_j.

After choosing this best classifier h_t, the new weight $v_{t+1,l}$ for the l-th sample is recalculated according to its classification result:

$$v_{t+1,l} = v_{t,l} e^{(-\alpha_t h_t(x_l))} \tag{5}$$

where $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_j}{\epsilon_j})$.

After all T iterations of the algorithm, we get the final classifier $H_i(x)$ for sub-window w_i. This classifier is of the form:

$$H_i(x) = \begin{cases} 1 \sum_{t=1}^{T} \alpha_t^i h_t^i(x) \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

where α_t^i is the selected weight for classifier $h_t^i(x)$, as chosen by the AdaBoost algorithm. We train such a classifier for every sub-window w_i.

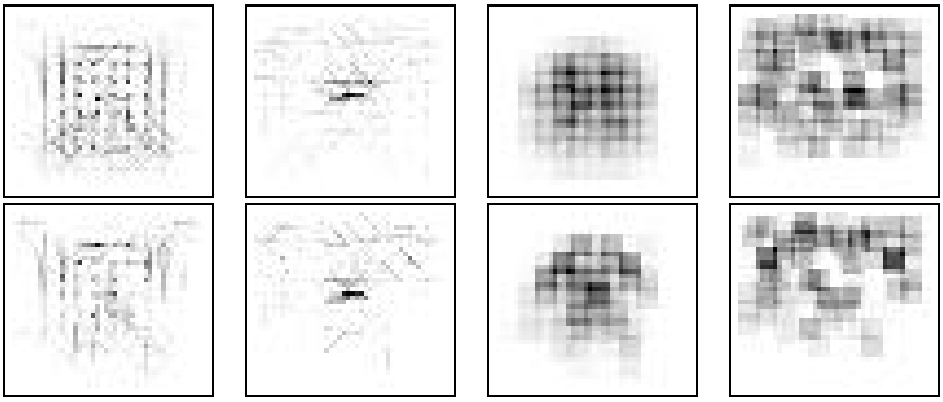
Each $H_i(x)$ is a local classifier, containing some of the low-level features inside the sub-window w_i. If we take a second look at the classifier form in equation 6, it can be seen that the weighted sum of weak classifiers is a continuous value. Let us call this sum $s_i(x) = \sum_{t=1}^{T} \alpha_t^i h_t^i(x)$. A useful characteristic about these classifiers is that this $s_i(x)$ contains more information than only specifying the class by its sign. The further away the value of $s_i(x)$ from the zero, the more certain we are about its estimated class. Therefore this value can be used as a confidence measure of the classification. This is similar to the confidence prediction AdaBoost that has been developed by Schapire and Singer (1998).

We define our motion shapelet features as these $\{s_i(x) : i \in \{1, 2, \ldots, k\}\}$. The index i corresponds to one of the sub-windows $w_i \in \mathcal{W}$, and $h_t^i(x)$ and α_t^i are the parameters associated with the classifier $H_i(x)$. Note that $s_i(x)$ is a motion shapelet feature that is trained specifically to distinguish between the two classes, based on background differences and gradients from its sub-window.

We train these motion shapelet features for a set of sub-windows inside the detection window. We visualize the results of the motion shapelet learning algorithm in Figure 5, which shows the sum of all the low-level features selected inside all the motion shapelet features over the entire the detection window. The selected low-level features are separated in two groups according to their classification parity p_t. This parity shows whether the selected feature is part of a positive (bear) or negative (non-bear) discriminating motion shapelet feature.

5.4 Final Classifier

Now that we have defined our motion shapelet features $s_i(x)$, we use AdaBoost to create a final classifier from them. The details of creating weak classifiers $g_t(s)$ are the same as the previous step. Each $g_t(s)$ consists of one of the motion shapelet features $s_t(x)$, a threshold θ_t, and a parity p_t.



(a) (b) (c) (d)

Fig. 5 Illustration of learned motion shapelets. Column (a) shows low-level gradient features with positive parity, (b) those with negative parity. Column (c) shows motion features with positive parity, (d) those with negative parity. The first row shows all motion shapelets, the second only those chosen for use in the final classifier.

The final classifier is in the form of:

$$C(s) = \begin{cases} 1 \sum_{t=1}^{T} \alpha_t g_t(s) \geq \lambda \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

where $s = (s_1(x), s_2(x), \ldots, s_k(x))$ denotes all the motion shapelet features for input image detection window x. λ is the final classifier's threshold; originally zero but can be adjusted to get different detection and false positive rates. Note that this time the weak classifiers $g_t(s)$ (equivalent of $h_t(x)$ in the previous step) are applied in the new feature domain s instead of the low level features x, and therefore the final classifier will be a combination of weighted thresholded motion shapelet features. Again, the selection of the motion shapelet feature, threshold, and parity for use in classifier $g_t(s)$, along with its weight α_t, are chosen using the variant of AdaBoost used by Viola and Jones (2001). When the final classifier is trained, one can illustrate all the low-level features that are inside the selected motion shapelet features. Depending on the parity p_t assigned to each of the features (classifiers), these features are indicative of either bears or non-bears. An illustration of both sets of features are shown in Figure 5. Note that the gradient features with positive parity tend to correspond to the shape of the bear. The negative gradient features are horizontal gradients in the centre of the window (where there should be no edges for a bear image) and gradients around the outside of the window, likely denoting

confounding background clutter. Similarly, the motion features with positive parity depict the rough silhouette of a bear, while the negative parity features are on the boundary of the window.

6 Experiments

The BearCam system was deployed in the Yukon from October 10-25, 2005, to monitor the river. Each day approximately four hours of video were recorded, over a time frame of approximately 15 days. As described above, our system streams video from the cameras over a wireless link, and records it onto hard drives contained in a hut adjacent to the river. A generator outside the hut powers the recording computer and hard drives. The cameras, mounted across the river, are powered by rechargeable batteries. Due to the inclement weather conditions, with temperatures ranging between $-15°C$ and $5°C$, the batteries were difficult to charge. We changed the batteries daily despite requiring up to 48 hours before reaching room temperature. As mentioned earlier the batteries could only be safely fasted charged at room temperature.

In order to test the efficacy of our bear detection algorithms we manually annotated frames from the video. Rectangular windows from the image were manually cropped and labeled as either containing a bear or not. For training, we marked 451 windows containing bears from 6 different clips of video as our positive set, and extracted 45100 non-bear windows from the same videos. A separate test set consisting of 400 bear and 40000 non-bear windows from a separate set of 4 video clips was extracted. The mean size of the bear windows in the training set was 36x40, and all windows were resized to this size for the experiments described below.

In cases where the (darkly-coloured) bear appears on regions of white snow, the detection problem is relatively straight-forward, with the background difference features providing adequate information for detection. However, in some cases the movement of the water and trees, or camera motion caused by wind will create imperfections in the background difference features. Examples of such difficult cases are shown in Figure 6.

In the following sections we describe our experiments using these manually labeled ground truth sets. We performed cross-validation using the training set in order to experiment with and tune the parameters of our detector. Then, we performed two experiments on the test set, using the best

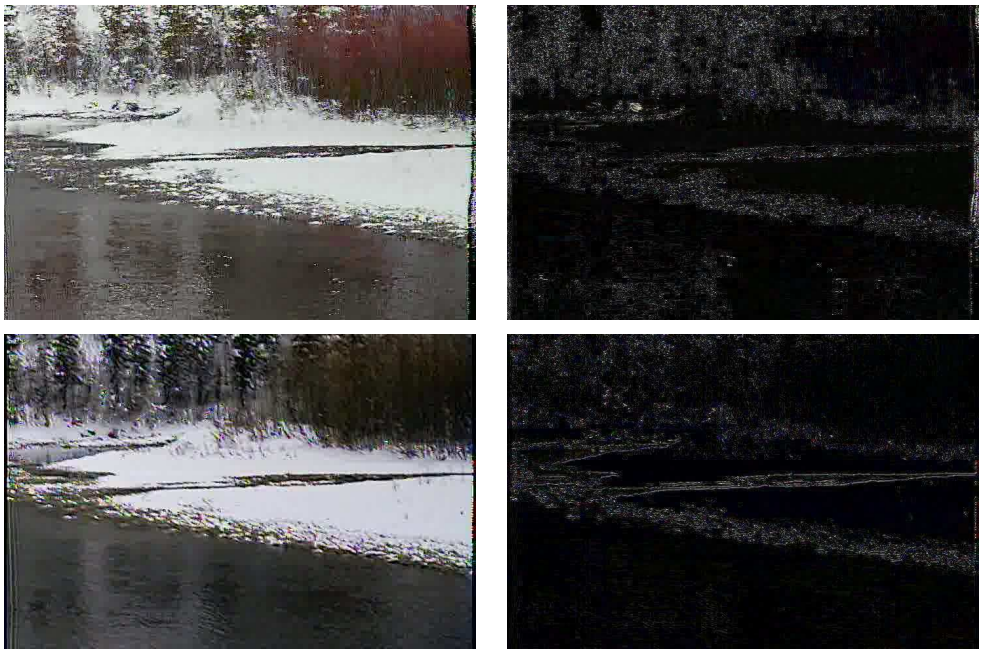


Fig. 6 Examples of difficult cases for bear detection using background difference features. False positives are often found along the river bank in examples such as these.

parameters from the cross-validation runs. The first experiment tested classification performance on the manually labeled, cropped windows. The second tested a full window-scanning detector that processes entire input video frames.

6.1 Motion Shapelet Parameters

We perform two experiments using the training set to explore the sensitivity of our algorithm to various parameters. The first experiment jointly explores the use of different sub-window sizes for computing motion shapelets, and the spacing of these sub-windows within the detection window.

As we described in section 5.3, there is a sub-window set $\mathcal{W}$ that defines the area of influence of each of the mid-level motion shapelet features. In our experiments, each sub-window in $\mathcal{W}$ is a square window; we used windows of size 5x5, 10x10, and 20x20. For each sub-window size, we experimented with a variety of values for the stride parameter, which determines how far apart to place each sub-window. We used equal values for strides in the horizontal and vertical direction, experimenting with stride values of 2, 4, and 8. For example, a stride of 4 would place the upper-left corner of the sub-windows at pixels (1,1), (1,5), (1,9), ..., (5,1), (5,5), ...

When comparing these detectors using different sub-window sizes and strides, we ensure that they are able to use the same amount of computational resources. We fix the total number of weak classifiers that is used in constructing all of the motion shapelet features to be 1000. For example, for 5x5 sized motion shapelets, tiled at a stride of 4 across the 36x40 detection window, there would be a total of 72 sub-windows within the detection window. We would therefore select 14 ($\approx 1000/72$) weak classifiers from low-level features to construct a motion shapelet for each sub-window.

Our results are presented using Detection Error Tradeoff (DET) curves (Martin et al., 1997). DET curves plot miss rate versus false positives per window (FPPW), on a log-log scale. The log-log scale enables one to more easily view differences that can be obscured in standard Receiver Operating Characteristic (ROC) curves.

Figure 7(a) shows DET curves for the 9 parameter settings, all combinations of motion shapelet sub-window size and strides. DET curves show average miss rate at FPPW values. Miss rate is averaged over 6 folds of leave-one-out cross validation. A bear detector with the given parameters is computed using 5 of the 6 sets in the training set, and a miss rate is calculated on the one held-out set.

From Figure 7(a), the algorithm is relatively insensitive to the settings of the parameters, provided a smaller sub-window size for the motion shapelets is used. Motion shapelets of sizes 5x5 and 10x10, within the 36x40 detection window, have similar performance.

The tradeoff between the strides and the and the number of weak classifiers used in each motion shapelet feature is less clear. For both 5x5 and 10x10 motion shapelets, a stride of 4 (resulting in 14 and 18 low-level features per motion shapelet respectively) gives the best performance, presumably striking a balance between motion shapelet overlap, and the redundancy in choosing too many similar features in one motion shapelet.

The second experiment we perform on the training set is to explore how the performance of our algorithm varies with the total number of weak classifiers or low-level features used in constructing the motion shapelets. In the previous experiment, we kept this number fixed at 1000, and varied the motion shapelet size and spacing. In this experiment, we use the best values (5x5 motion shapelet, stride of 4) from the previous experiment, and train bear detectors using 200, 600, 1000, 1400, and

1800 low-level features for constructing the motion shapelets. Figure 7(b) shows DET curves for these detectors.

As expected, performance increases as the total number of features used increases. However, this improvement plateaus after 1000 features. The difference between 1000 and 1400 or 1800 low-level features is marginal at the low FPPW values, which are likely to be of interest. For this reason, we choose 5x5 motion shapelets, with a stride of 4, and 1000 total low-level features, as our parameters for our experiments on the test set.

In addition, we performed an experiment to measure the relative contributions of the gradient (shape) and motion features in classifying bears in our videos. Figure 7(c) compares the results of 3 different detectors. All use the same 5x5 shapelet size, with a stride of 4, and 1000 total low-level features. We build detectors selecting only from the gradient, only from the motion, and both features respectively. As one would expect, the motion features are more important, but adding gradient features does improve performance. In regions where moving trees or rippling waves cause motion, the spatial gradient features can help to reduce false positives.

Qualitative examples of performance are shown in Figure 8. False negatives from the window-based classifier are shown. High-ranked false negatives include frames with noise in the video data and those where bears are near or in the river, in which reflections and the motion of water cause extraneous patterns of motion.

6.2 Results on Cropped Test Set

The first experiment on the test set was done on the manually cropped windows, 400 positive which contain bears, and 40000 negative which contain background. The purpose of this experiment is to test the generalization performance of our detector to unseen data, while remaining in the cropped-window framework for ease of comparison.

We took the best detector parameters from the previous cross-validation phase, namely 5x5 motion shapelets sampled at a stride of 4 over the detection window, with 1000 total low-level features used in all of the motion shapelets. We trained a final detector using these parameters and all of the 451

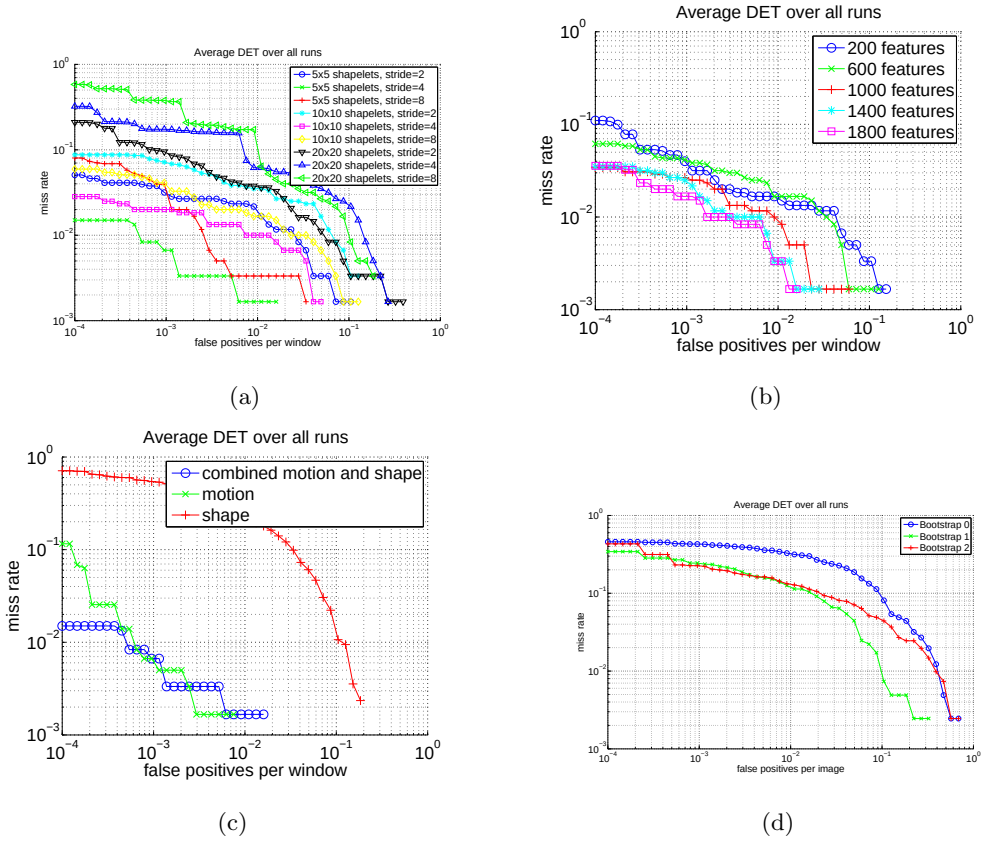


Fig. 7 (a) Detection Error Tradeoff (DET) curves averaged over 6 folds of leave one out cross-validation. Error rate versus false positives per windows is shown on a log-log scale. 9 different parameter settings, with 3 sizes of motion shapelets and 3 stride sizes are shown. (b) DET curves averaged over 6 folds of leave one out cross-validation. Error rate versus false positives per windows is shown on a log-log scale. All detectors use 5x5 motion shaplets, with a stride of 4 over the detection window. Results are shown for 200, 600, 1000, 1400, and 1800 total low-level features in all motion shapelets. (c) DET curves averaged over 6 folds of leave one out cross-validation. Error rate versus false positives per windows is shown on a log-log scale. Results for detectors selecting only from the spatial gradient, only from the motion, and from both features are shown. (d) DET curves on the test set of full video frames. Error rate versus false positives per image is shown on a log-log scale. All detectors use 5x5 motion shaplets, with a stride of 4 over the detection window, and 1000 total low-level features. Results are shown for initial detector with 2000 negative windows, and first and second round bootstrap-trained detectors with 5000 and 8000 total negative windows respectively.

positive bear windows. The negative set for this detector was selected by randomly choosing 2000 negative windows from the training set.

This test set proved to be too simple for our algorithm. Near-perfect performance was obtained, with zero error rate with a negligible number of false positives. For this reason, we proceeded to a more

Fig. 8 Examples of false negatives from window-based classifier. High-ranked false negatives include frames with noise (e.g. top row), and those where the bears are near or in the river, with reflections and motion of water.

difficult test, scanning entire video frames, running our detector at every possible location to look for bears.

6.3 Results on Video Frames

The final test we performed processed whole video frames, a realistic test of the final system. We ran our detector in a "window-scanning" fashion, sliding it across the input frame and running our classifier at each location.

We created a test set from the same set of videos used in the cropped test set. For the purposes of this experiment, an image is considered a true positive if it contains at least one bear, and a false positive if it contains no bears. 405 images containing at least one bear were labeled as our positive set, while 16000 images not containing bears comprise the negative set.

We marked a region of interest (ROI) within the video, in order to ignore areas where bears cannot appear. The ROI consists of the entire frame, except the upper regions of the frame where only trees are present. We ran our detector over each image, and kept the single highest-valued response over all detection windows within the ROI as the response for each image. This process is quite efficient, and our MATLAB implementation processes a frame in less than 3 seconds, on a 2.4GHz Opteron processor. A real-time implementation would likely be feasible, particularly if one employed cascades (Viola and Jones, 2001) for efficiency.

The parameters for the detector and the positive training set are the same as those described in the preceding subsection. The negative set for this detector was selected using bootstrapping on the training set. In the first round, 2000 negative windows were randomly selected from the negative training set. We performed two rounds of bootstrapping. In each round we added 3000 additional

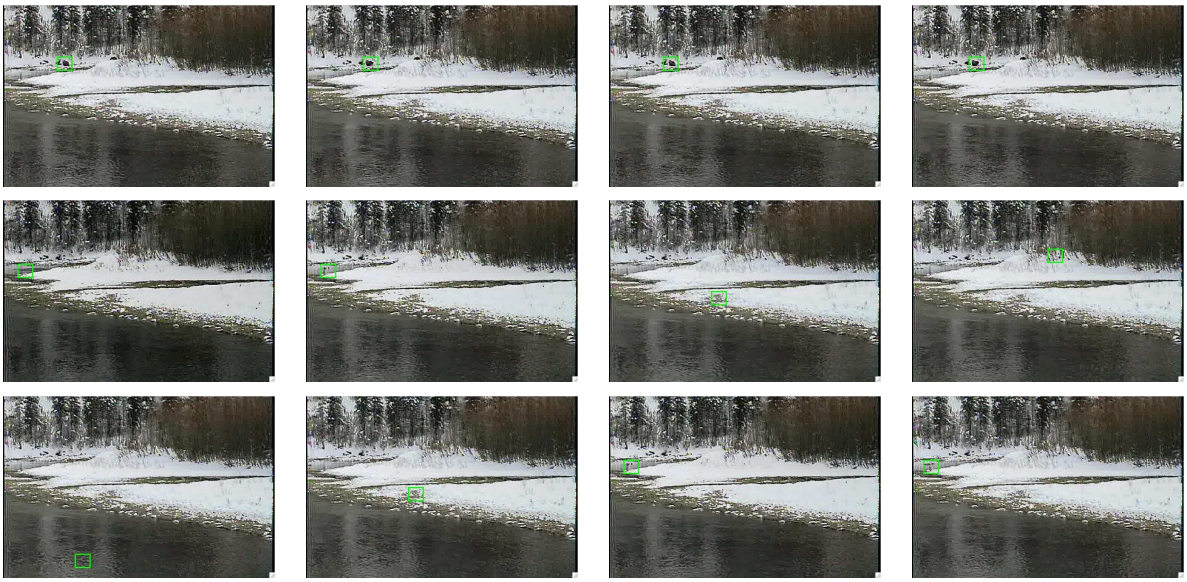


Fig. 9 Top row: Examples of correct bear detection results. Second and third row: Examples of bear detection failures. The top ranked false positives are shown.

negative windows, chosen as the most difficult negative examples for the previous classifier (those with highest score under the previous classifier).

DET curves showing the results of these experiments are shown in Figure 7(d). We note that the first initial run of bootstrapping significantly improves the performance of our detector, for example reducing the error rate from 0.43 to 0.24 at 10^{-3} false positives per image (FPPI). The second stage of bootstrapping does not significantly alter the results at low FPPI rates, and performs slightly worse at higher FPPI rates (note the log scale, the difference is of a few hundreths in error rate).

Example frames with detection results are shown in the first row of Figure 9. The second and third row of Figure 9 shows the worst false positives, i.e. the frames without bears for which the detector returned the highest value. The false positives are often found in textured areas, such as those on the banks of the river, or regions of the water, which contain large amounts of gradient information. In addition, areas of motion, such as the birds which appear in the upper left of the frames, or the ripples in the water, also lead to false positive detections.

7 Conclusion

In this paper we described the development of the BearCam, a camera system used to monitor the behaviour of grizzly bears at a remote location near the arctic circle. The system aided biologists in data collection for their study on behavioural effects of ecotourists on bears. We developed a camera system for operating in the challenging arctic conditions. The system collected vast quantities of video data, much of it devoid of bears. In order to reduce the amount of manual labour that would be required to process these data and extract information regarding the behaviour of the bears, we developed a novel algorithm for detecting the presence of bears.

This "motion shapelet" algorithm is an extension of the shapelet features (Sabzmeydani and Mori, 2007), which are mid-level features capturing pieces of shape. Our extension of this technique incorporates motion information, and proves effective at automatically detecting the occurrence of bears. We presented experiments on tens of thousands of frames of video, showing that we can obtain low false negative rates for bears while eliminating large amounts of data. For example, we can detect 76% of the frames containing bears at 0.001 false positive images per image examined, or 88% at 0.01. Such a detection rate would be useful for building an interactive system, where a user could be guided to frames in the video where bears are likely to be present.

Future work includes automating more of the data collection for our biologist partners. For example, one could attempt to extract some of the more detailed data, such as the percentage of time the bears spend walking, fishing, or feeding, automatically. One could build a tracker on top of the current bear detection algorithm, and attempt to use action recognition algorithms to classify the resulting tracks of the grizzly bears.

Acknowledgements The authors would like to thank Pawel Zebrowski for all the help with the metal work and the paint job. Also thanks to Sarah Brown for the wonderful art work. Funding for the BearCam project was provided by the Canadian Foundation for Innovation (CFI) and the British Columbia Knowledge Development Fund (BCKDF) as part of the Scientific Data Acquisition Transportation and Storage (SDATS) project. Funding for the biological field research provided by Yukon Government Department of Environment; Northern Scientific Training Program, Indian and Northern Affairs Canada; and Northern Research Institute, Yukon College.

References

WV-CP480 Series datasheet. `http://www.panasonic.ca`.

T. Balch, Z. Khan, and M. Veloso. Automatically tracking and analyzing the behavior of live insect colonies. In *Fifth International Conference on Autonomous Agents*, 2001.

Serge Belongie, Kristin Branson, Piotr Dollar, and Vincent Rabaud. Monitoring animal behavior in the smart vivarium. In *MB*, 2005.

M. Betke, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and T. H. Kunz. Tracking large variable numbers of objects in clutter. In *CVPR*, 2007.

D. K. Chi and B. K. Gilbert. Habitat security for Alaskan black bears at key foraging sites: are there thresholds for human disturbance? *Ursus*, 11:225–238, 1999.

A. P. Crupi. *Foraging behavior and habitat use patterns of brown bears (Ursus arctos) in relation to human activity and salmon abundance on a coastal Alaskan salmon stream.* PhD thesis, Department of Fisheries and Wildlife, Utah State University, Logan, USA, 2003.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

N. Dalal, B. Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

Norris L. Dodd, Jeffrey W. Gagnon, Amanda L. Manzo, and Raymond E. Schweinsburg. Video surveillance to assess highway underpass use by elk in Arizona. *J. of Wildlife Management*, 71(2):637–645, 2007.

Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

M. Duchesne, S. D. Cote, and C. Barrette. Responses of woodland caribou to winter ecotourism in the Charlevoix Biosphere Reserve, Canada. *Biological Conservation*, 96:311–317, 2000.

M. G. Dyck and R. K. Baydack. Vigilance behavior of polar bears (U. maritimus) in the context of wildlife-viewing activities at Churchill, Mb, Canada. *Biological Conservation*, 116:343–350, 2004.

Ahmed M. Elgammal and Larry S. Davis. Probabilistic framework for segmenting people under occlusion. In *ICCV*, 2001.

EnerSys. Genesis purelead application manual. `http://www.enersys.com`, 2006.

Pedro F. Felzenszwalb. Learning models for object recognition. In *CVPR*, 2001.

M. Fink and P. Perona. Mutual boosting for contextual inference. In *NIPS*, 2004.

D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *ICCV*, 1999.

D. M. Gavrila. The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, 1999.

National Geographic. The elephant seal cam. `http://magma.nationalgeographic.com/ngm/sealcam/index.html`, 2006a.

National Geographic. Wildcam grizzlies. `http://www9.nationalgeographic.com/ngm/wildcamgrizzlies/technology.html`, 2006b.

G. V. Hilderbrand, C. C. Schwartz, C. T. Robbins, M. E. Jacoby, T. A. Hanley, S. M. Arthur, and C. Servheen. The importance of meat, particularly salmon, to body size, population productivity, and conservation of North American brown bears. *Canadian Journal of Zoology*, 77:132–138, 1999.

Solar Energy International. *Photovoltaics : design and installation manual*. New Society Publishers, September 2004. ISBN 0865715203.

D. E. Jelinski, C. C. Krueger, and D. A. Duffus. Geostatistical analyses of interactions between killer whales (Orcinus orca) and recreational whale-watching boats. *Applied Geography*, 22:393–411, 2002.

A. Johnson, C. Vongkhamheng, M. Hedemark, and T. Saithongdam. Effects of human-carnivore conflict on tiger (Panthera tigris) and prey populations in Lao PDR. *Animal Conservation*, 9:421–430, November 2006.

Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. In *CVPR*, 2004.

Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE T-PAMI*, 27(11):1805–1819, 2005.

Kyocera. Manual of KC-Series solar photovoltaic power modules.

Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.

A. G. MacHutchon, S. Himmer, H. Davis, and M. Gallagher. Temporal and spatial activity patterns among coastal bear populations. *Ursus*, 10:539–546, 1998.

Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The det curve in assessment of detection task performance. In *EuroSpeech*, volume 4, pages 1895–1898, 1997.

MaxStream. Product manual, xstream oem rf module. `http://www.maxstream.com`, 2005.

K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *Proc. ECCV*, 1:69–81, 2004.

Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Trans. PAMI*, 23(4):349–361, 2001.

S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. PAMI*, 28(11):1863–1868, 2006.

Maryam Moslemi Naeini, Greg Dutton, Kristina Rothley, and Greg Mori. Action recognition of insects using spectral clustering. In *IAPR Conference on Machine Vision Applications*, 2007.

O. T. Nevin and B. K. Gilbert. Perceived risk, displacement and refuging in brown bears: positive impacts of ecotourism? *Biological Conservation*, 121:611–622, 2005.

T. L. Olson and B. K. Gilbert. Variable impacts of people on brown bear use of an Alaskan river. In *International Conference on Bear Research and Management*, volume 9, pages 97–106, 1994.

T. L. Olson, R. C. Squibb, and B. K. Gilbert. Brown bear diurnal activity and human use: a comparison of two salmon streams. *Ursus*, 10:547–555, 1998.

Panasonic. LC-R1233 lead acid battery datasheet. `http://www.panasonic.com`, August 2003.

Pelco. EH4700 Series environmental enclosures. `http://www.pelco.com`, November 2003.

A. Pitts. Effects of wildlife viewing on the behavior of grizzly bear (Ursus arctos) in the Khutzeymateen (K'tzim-a-deen) Grizzly Bear Sanctuary, British Columbia. Master's thesis, Faculty of Agricultural Sciences, University of British Columbia, Vancouver, Canada, 2001.

C. W. Roberts, B. L. Pierce, A. W. Braden, R. R. Lopez, N. J. Silvy, P. A. Frank, and D. Ransom Jr. Comparison of camera and road survey estimates for white-tailed deer. *J. of Wildlife Management*, 70(1):263–267, 2006.

Andrew Rova, Greg Mori, and Lawrence M. Dill. One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. In *IAPR Conference on Machine Vision Applications*, 2007.

P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *CVPR*, 2007.

R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 80–91, 1998.

Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, volume 1, pages 746–751, 2000.

Dez Song and Ken Goldberg. Acone: Automated collaborative observatory for natural environments. `http://www.c-o-n-e.org/acone/`, 2006.

Alex Streeter. The design, construction, and control of a photovoltaic power system for an autonomous antarctic rover. Master's thesis, Thayer School of Engineering, Dartmouth College, 2005.

P. Viola and M. Jones. Robust real-time object detection. In *2nd Intl. Workshop on Statistical and Computational Theories of Vision*, 2001.

P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, pages 734–741, 2003.

B. G. Walker, P. D. Boersma, and J. C. Wingfield. Habituation of adult Magellanic Penguins to human visitation as expressed through behavior and corticosterone secretion. *Conservation Biology*, 20(1): 146–154, 2006.

C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780–785, July 1997.

Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005.

Tao Zhao and Ram Nevatia. Bayesian human segmentation in crowded situations. In *CVPR*, 2003.

Shelley Marshall followed her BSc degree in Biology from Simon Fraser University in 1999 with a Technical Diploma in Fish, Wildlife, and Recreation from British Columbia Institute of Technology in 2003. She received her Master of Natural Resource Management from the School of Resource and Environmental Management at Simon Fraser University in 2008. Shelley is now a Wildlife Technician working with the Yukon Governmen's Carnivore Biologist in Whitehorse, Yukon.

Jens Wawerla received his undergradute degree in Computer Science from the University of Applied Siences Konstanz in Germany. After working for 2 years in industry on unmanned aerial vehicles, he is currently a PhD student in the School of Computing Science at Simon Fraser University in Vancouver, British Columbia. His research interests are long lived, adaptive, and self-sustaining systems.

Payam Sabzmeydani received his BSc degree in computer Engineering from Sharif University of Technology in 2003. He received his MSc degree in Computing Science from Simon Fraser University in 2007. His thesis was on pedestrian detection in still images. He is currently a software engineer at Koolhaus Games Inc.

Greg Mori was born in Vancouver, BC. He received an Hon. B.Sc. with High Distinction, in Computer Science and Mathematics, from the University of Toronto in 1999. During his undergrad years, he spent one year ('97-'98) as an intern at Advanced Telecommunications Research (ATR) in Kyoto, Japan. He obtained his Ph.D. in Computer Science from the University of California at Berkeley in 2004. After graduating from Berkeley, he returned home to Vancouver and is currently an Assistant Professor in the School of Computing Science at Simon Fraser University. His research interests are in computer vision, and include object recognition, human activity recognition, and human body pose estimation. The main thrust of his research has been in exploring methods for analyzing images and videos of people. He has also developed methods for object recognition in cluttered scenes. He has applied those techniques to break the "CAPTCHA" word-recognition puzzles, work that was featured in the New York Times. Greg serves on the program committee of major computer vision conferences (ICCV, CVPR, ECCV), and was the program co-chair of the Canadian Conference on Computer and Robot Vision (CRV) in 2006 and 2007.

Kristina Rothley received the degrees of SB in Mechanical Engineering, Massachusetts Institute of Technology, 1986, ME in Mechanical Engineering and MBA in Marketing, Cornell University, 1988 and 1989, respectively, and MFS and PhD in Wildlife Ecology, Yale University, 1995 and 1999, respectively. From 1999-2001, she was a David H. Smith Post-Doctoral Fellow at Princeton University in the Department of Ecology and Evolutionary Biology. From 2001-2007, she was an Assistant Professor at Simon Fraser University in the School of Resource and Environmental Management. She is now an Assistant Professor at Kutztown University of Pennsylvania in the Biology Department.