Table of Contents

Abstract

Problems found in research domains which utilize remote video footage, such as environmental monitoring, can provide opportunities for the development of new computer vision-based approaches. Monitoring animal species in surveillance footage, for example, involves tasks which are time-consuming if done by hand. Manual counting of a bird known as the chimney swift is one such task, and is problematic due to the large amount of generated footage combined with difficult to interpret frames. Thus, this paper proposes an automated approach for tracking and counting the behaviors of chimney swifts in video datasets. It consists of multiple distinct stages, including swift localization, swift tracking, event detection, and event classification to generate estimated counts. Methods chosen for each stage include RPCA motion estimation, a Hungarian algorithm-based tracking framework, and classification of segments based on flight angle features. This approach was implemented using Python and tested on three video datasets from two distinct sources covering a range of visual conditions. Experimental evaluation is performed using manual annotation of these video datasets. Initial results show promise, with average F1 scores of 0.8761 across over 1700 instances of swifts entering chimneys. Notably, angle classification is shown to increase performance compared to event detection alone. Analysis of sources of error show several potential areas for improvement. Future work may include addressing swift occlusion, accounting for the presence of non-swift birds, and determining more robust classification boundaries.

1 Introduction

As capable hardware for recording and storing video data becomes increasingly accessible, new opportunities arise for interdisciplinary research which marries both computer vision and unrelated areas of study. Academic researchers in areas like environmental monitoring and conservation, for example, can observe their subjects remotely using video feeds. This way, disturbance to natural habitats is minimized [1]. Approaches from computer vision can then be applied to recorded video data in cases where large quantities of footage would otherwise be prohibitive for manual inspection. Thus, improvements to techniques such as object detection, localization, and tracking can help overcome barriers to building an understanding of the world and its inhabitants.

Bird species conservation research, in particular, is an attractive domain to apply frame-by-frame video analysis to due to the challenges associated with the height and speed of bird flight. Human vision is limited in what it can perceive, and human memory can be unreliable. Static camera systems installed at locations frequented by birds, then, can offer a way to expand past perceptual limitations when studying a species' behaviors. One example of conservation research which incorporates video analysis is the ongoing effort to address knowledge gaps in the behaviors of the chimney swift, a threatened migratory bird. Identifying priority roost and nest sites is a primary objective of the conservation of this species [2]. Swifts temporarily roost inside fixed chimneys during migration, making nesting site analysis ideal for the use of static video camera installations. Detecting and counting swifts which have entered chimneys is key to identifying active sites and is carried out by citizen scientists and academic researchers alike.

Manual counting of swifts entering chimneys is tedious work, as it requires navigation through video files which are lengthy and difficult to interpret. Nesting periods are variable from day-to-day, requiring observers to review hundreds of thousands of frames of video footage per 24-hour cycle to determine relevant periods of activity. Within video frames, the large distance between the chimney and camera lens often results in poorly resolved swifts, causing them to appear as small, blurred objects which lack detail. Also, swift flight patterns are erratic compared to typical bird flight, resulting in frequent rotations and non-rigid deformations [3]. Swifts can circle in flocks prior to and during flight into the chimney, too. This behavior causes clutter, as a scene may contain swifts flying at varying distances and directions at once. Distinguishing between chimney swifts flying into a chimney and those flying behind a chimney becomes a significant challenge under these conditions. Thus, it becomes required to examine many sequential frames per bird to develop a sense of their motion paths. This intensive labor acts as a bottleneck for time, preventing skilled individuals from applying their expertise elsewhere.

Figure 1: *Example of frame from May 18th Sault Ste. Marie video footage with notable swifts.* ***(A)*** *Full frame,* ***(B)*** *Distinct swifts near chimney entrance,* ***(C)*** *Isolated swift with wing motion blur,* ***(D)*** *Swifts at risk for occlusion,* ***(E)*** *Heavily blurred swifts,* ***(F)*** *Motion trail artifacts.*

To address difficulties in monitoring swift behavior, this paper proposes an automated method for detecting when a swift has entered an in-frame chimney. The remainder of the paper is organized as follows: Section 2 presents an overview of comparable research in swift monitoring, object localization, and object tracking. Section 3 describes in detail the proposed approach, with an experimental evaluation of the approach examined in Section 4. Finally, conclusions and recommendations for future research are provided in Sections 5 and 6 respectively.

2 Related Work

2.1 Swift Monitoring

For context, formalized procedures for the manual reporting of swift counts currently exist, and detail information relevant for accurate monitoring of chimney swifts [4]. Apart from this, there are several examples of research into automated tracking and counting for chimney swifts. Notably, the Mersey Tobeatic Research Institute deployed an Eco-Visio hardware-based data logger as part of counts conducted in 2013 [5]. Reports from the following year, however, state that “the logger consistently underestimated the number of birds” and that “the data logger was not deployed in 2014.” Thus, software-based methods are a potential alternative and will be examined further in forthcoming sections.

A separate research project conducted by students at the University of Victoria utilizes a MOSSE adaptive correlation filter to track swifts [6]. It also uses trajectory predictions and a rectangular chimney region-of-interest to trigger event detection. However, swift datasets contain a number of additional challenges compared to the test sequences used to demonstrate the effectiveness of MOSSE filters [7]. Unlike the faces of humans or animals found in those sequences, swift objects possess fewer characteristics (such as texture, shape, or color) which could distinguish them from other objects found within typical video scenes. Swift objects also undergo rotations and nonrigid deformations at rates faster than the objects demonstrated in MOSSE test sequences. These differences, along with the approach’s average recall rate of 0.597 demonstrated on selected swift video sequences, suggest that this approach in its current form is unsuitable for counting chimney swifts. However, the overall framework involving localizing and tracking swifts may still be useful if more appropriate methods are chosen for

each stage. Thus, the remaining related works will be grouped in accordance with this framework, with Section 2.2 covering object localization, and Section 2.3 covering object tracking.

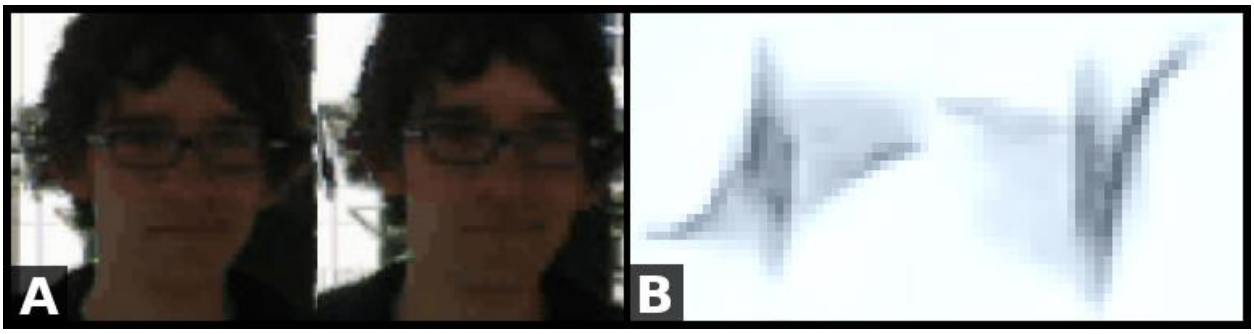

Figure 2: *Comparison between MOSSE test frames and June 13th Chalk River swift frames.* ***(A)*** *Two consecutive frames of trellis.avi dataset [7] demonstrating little change in shape or present features,* ***(B)*** *Two consecutive swift frames demonstrating abrupt changes in shape and present features.*

2.2 Object Localization

Tracking objects between video frames often requires initially identifying each object's location within a scene. For highly detailed objects with distinguishing features, a common approach is to utilize keypoint detection and feature descriptors. Yao and Jiebo [8], for example, use SIFT feature descriptors, combined with manually annotated training samples, to generate a density map for each pixel in video frames of migrating birds. No recommendations are provided, however, for converting this density map into individually labelled objects. T'Jampens et al. [9] test not just SIFT feature descriptors, but SURF and HOG descriptors as well. Their method is targeted at bird detection in marine video content, so a codebook is created for water motion keypoints as well as bird keypoints. They then test an SVM classifier with both linear and RBF kernels to distinguish between bird segments and non-bird segments. Their research demonstrates that the combination of HOG feature descriptors with an RBF kernel successfully filters out non-bird segments, with F1 scores of 0.95 for both bird and water segments. While

both methods are suitable for determining whether a portion of a scene is or isn't a bird, there is no guarantee that these methods alone could uniquely localize individual birds within a frame. The lack of features which could distinguish between multiple chimney swifts further suggests it may be difficult to segment individual swifts within a scene using this type of approach.

If contrast between foreground objects and their background can be assumed, intensity-based thresholding could be an alternative to feature description for object segmentation. Both Tou and Toh [10] as well as Laliberte and Ripple [11] use a manually-chosen intensity-based threshold to segment birds within aerial images. Dash and Albu [12] also make use of manual intensity-based thresholding in their work counting flocks of birds, but augment this approach with additional non-photorealistic edge enhancement and linear Gabor filtering stages. In research into flamingo counting, Groom et al. [13] depart from manual thresholding by first using quadtree segmentation to ensure regularity in segment size. They then apply a recursive intensity-based thresholding algorithm to isolate local maxima in the image, which are then treated as parts of flamingos. Despite the prevalence of intensity-based segmentation in the above approaches, the assumption of contrast does not necessarily hold for scenes containing chimney swifts. This is because typical scenes include not just a lighter background (sky) but also darker background components (including the chimney itself). In practice, swifts are similar enough in intensity to the chimney that contrast is not guaranteed. Thus, these approaches are unsuitable without additional modification.

If video files are an option as opposed to still images, then motion analysis techniques such as background subtraction can provide an alternative for bird segmentation. T'Jampens et al. [9] apply temporal median filtering to continuously update a background model for segmenting foreground birds from marine video content, then dilate the resulting segmented image. Shakeri and Zhang [14] choose instead to use Gaussian mixture modeling for the

segmentation stage of a real-time bird detection method. In both approaches, however, the datasets used for evaluation contain birds which are sparsely distributed throughout a scene. There is no guarantee that these methods would be able to segment chimney swifts from cluttered scenes. This would especially be a concern for the dilation stage used by T'Jampens et al., as it risks merging several swift segments together. For completeness, an overview and comparison of background subtraction techniques by Bouwmans [15] is provided for further reading. It compares several additional categories of background subtraction techniques which may prove more adept at segmenting tightly packed flocks of birds.

Multiple uses of shape-based modeling for localization can be found in recent literature, as well. Descamps et al. [16] detect individual flamingos by modeling them as ellipses contrasting against a darker boundary, with slight overlapping allowed. Their implementation adapts a marked point process model originally intended to detect trees in aerial images [17]. Ellipse-fitting can also be found in separate research involving the segmentation of non-bird species that share visual characteristics with swifts. Branson et al. [18] segment fruit flies by modeling them using a Hough ellipse-fitting algorithm, while Fukunaga et al. [19] model key-points belonging to medaka fish using mixtures of 2D Gaussian distributions. Ellipse-fitting shows promise, as ellipse-shaped bodies of chimney swifts are consistently visible despite any blur caused by wing motion. Shape descriptors can also be found in research aimed at distinguishing between bird and non-bird segments. Zhang et al. [20], for example, utilize both static, low-level features and dynamic, high-level features. These features are determined using shape context analysis and flying-bird Markovian models respectively. However, due to the erratic nature of chimney swift flight, typical bird-like features (such as a clearly distinguishable wing, beak, or tail) are rarely seen over multiple consecutive frames. This lack of detail renders complex shape descriptors as unsuitable. As an alternative to this approach, Tou and Toh [10] use the maximum and minimum dimensions of segment bounding boxes, as well as the ratio

between segment area and bounding box, to filter out non-bird segments. This approach is simpler, working independently of the specific visual features of the objects themselves, and may be general enough to be applicable towards chimney swifts.

Finally, modifications to segmentation stages have been proposed to address the issue of overlapping segments. Tidmerng et al. [21] demonstrate the use of the distance transform combined with erosion of binarized segments to separate overlapping bird segments. This approach, however, was tested only on high quality images containing birds with clearly defined shape, which is distinct from how chimney swifts typically appear. Alternatively, in their work with fruit flies Branson et al. [18] propose a modification to a Hough ellipse-fitting algorithm which tests multiple sets of ellipses per segment. Each configuration is penalized based on the relative area of each ellipse, and a configuration is chosen based on this penalty. The shape of a fruit fly segment is bloblike and nondescript, which makes this approach appealing for species which do not possess distinguishing characteristics. Separation by ellipse-fitting, then, is more likely to be applicable to swift localization.

2.3 Object Tracking

Once a chimney swift has been segmented from a frame's background, methods are needed to track changes in the swift's position from frame to frame. This task is challenging due to the erratic flight behaviors of chimney swifts, which cause rapid changes in shape and flight direction. Another challenge involves the fact that scenes involving chimney swifts can contain swifts at multiple depths flying in varied directions. Swifts typically lack features which could distinguish them from other present swift objects, too. Also, if a video's frame rate is sufficiently high, it may take several frames for a swift to fully enter a chimney. In this case, a swift will appear as partially obscured before disappearing, which may cause ambiguity in detectable

motion due to the aperture problem. Related works must be evaluated in the context of these challenges.

Shakeri and Zhang [14] approach bird tracking by minimizing the distance between bird segments in subsequent frames. This approach does not consider errors which stem from cluttered scenes; however, this is acceptable for the associated research due to the infrequency of occlusion from other birds, as stated by the authors. However, due to the cluttered nature of chimney swift flock behavior, there is a high risk that distance minimization may transfer the identity of a tracked object. This would occur if a second swift's resulting position is close to a first swift's initial position. Distance alone, then, is not enough to consistently track a flying swift.

Several existing works use a method which involves predicting an object's position in an upcoming frame using its last position and its motion vector, then comparing that prediction to actual present segments in that frame. Depending on the accuracy of the prediction, segments can be either linked together, or they can be considered new. T'Jampens et al. [9] successfully apply this method when tracking birds in marine video content. Of note, however, is the fact that the marine video datasets contain bird flight which is steady and predictable, with few abrupt changes in direction or segment shape. This contrasts with typical swift flight behavior due to frequent rotations and nonrigid deformations in swift flight. Branson et al. [18] also use this approach when tracking fruit flies. Fruit fly movement is comparably erratic to chimney swifts. In fact, erratic movements are explicitly mentioned as a cause of identity swaps by Branson et al. For this reason, it would be desirable to choose a method which allows for flexibility in comparing prior segment information with new segment information, so that erratic flight can be accommodated for.

Tou and Toh [10] utilize the Horn-Schunck method of estimating optical flow vectors to track the motion of bird segments. However, this method of estimating optical flow relies on a number of assumptions to ensure accurate motion vectors [22] which may not hold for chimney swift motion. Due to the high velocity associated with swift flight, swift objects undergo large displacements between frames. Therefore, the small motion assumption is not guaranteed unless frame rate is sufficiently high. However, pyramidal structures are reported to improve performance significantly with regards to this assumption [22], and should be considered when evaluating this method. A second assumption which may be violated with swift flight is that of spatial coherence. As typical swift scenes are cluttered with swift flocks containing birds with flying in various directions, there is no guarantee that neighboring points will move together. Furthermore, the presence of a chimney can partially obscure swifts, causing additional ambiguity in point motion. For these reasons, it is unlikely that optical flow can consistently describe the motion of chimney swifts.

Apart from bird tracking, additional object tracking methods exist in separate problem domains. Of note, Huang et al. [23] describe data association-based tracking methods which "consider associations beyond frame-by-frame basis, linking short trajectory fragmentations (i.e., tracklets) into longer trajectories by global optimization based on position, motion, and appearance similarities." As part of their take on this approach, the authors formulate tracklet association as a standard assignment problem and use the Hungarian algorithm to find a minimum cost solution across all possible tracklet combinations. This approach is tested on pedestrians in shopping malls and airport terminals but is stated to be generalizable to other objects in diverse circumstances, thus may be considered for the problem of swift tracking. Separately, Dash and Albu [12] hypothesize that high-density crowd-tracking methods originally designed for humans may be applicable towards flocks of birds. Examples of such an approach include the use of floor fields by Ali et al. [24] and the use of binary quadratic programming by

Dehghan and Shah [25]. However, as swift scenes contain flight with varying depths and directions at once, swift flocks do not move uniformly as crowds, thus dense crowd modeling is unlikely to be applicable here.

3 Proposed Approach

The proposed approach follows a structure similar to that of bird detection and tracking algorithms found in Section 2, particularly the work of T'Jampens et al. [9] as applied to marine video content. This entails a segmentation stage to separate chimney swifts from a static background, as well as a tracking stage to determine the motion paths of swifts across consecutive frames. The approach for each stage is chosen to address the unique challenges in chimney swift data sets, namely the erratic flight and lack of distinguishing characteristics in present chimney swifts. The proposed approach also extends the tracking algorithm by way of an event detection stage for chimney swifts which have entered an in-frame chimney. This is done through the extraction of features from segments and their motion paths which are then used to classify candidate chimney swifts. The remaining sections are organized to reflect the distinct stages of this framework. The overall structure of the approach is demonstrated in Figure 3.

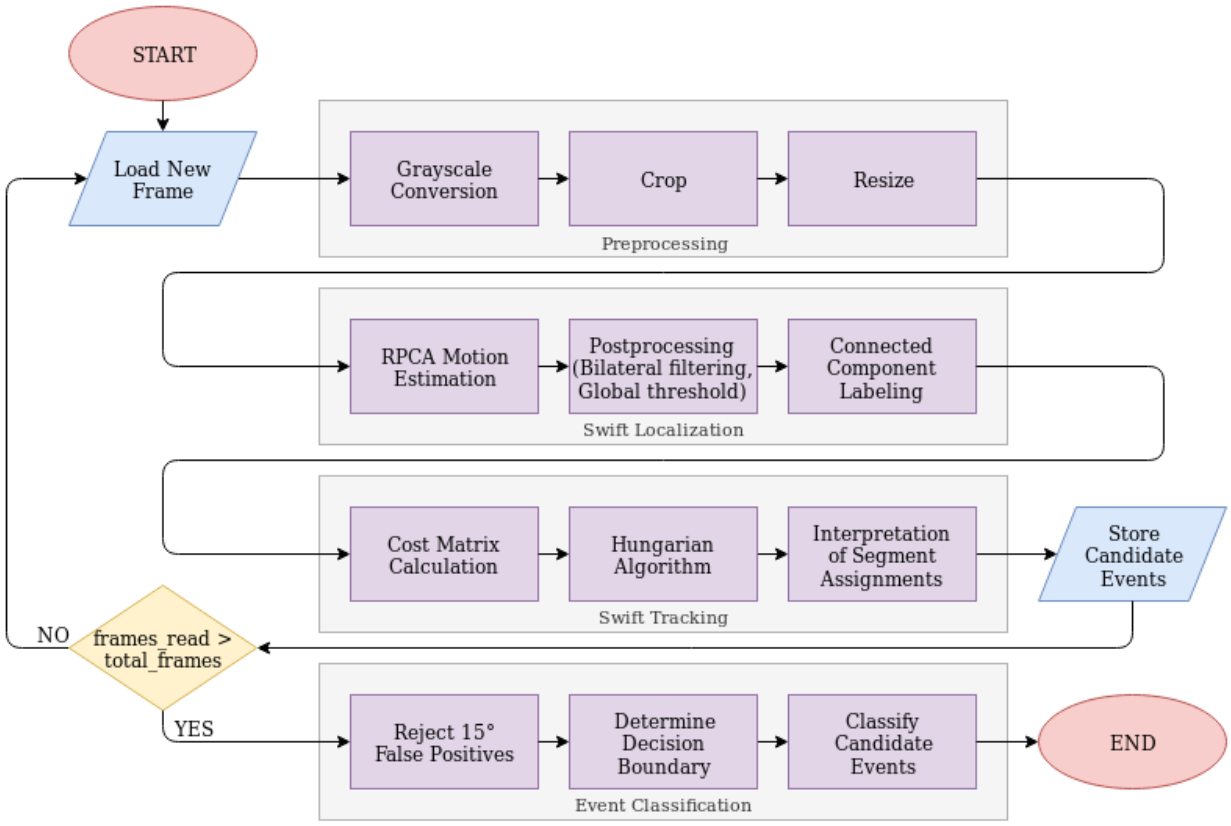

Figure 3: *Block diagram demonstrating individual stages of automatic swift-counting algorithm.*

3.1 Initialization

This algorithm is designed to run continuously on color video footage with durations upwards of one hour. However, prior to continuous processing an initialization stage is performed to determine regions of interest within the frame. As only flight into the chimney is relevant for detecting instances of entering swifts, much of the area within the frame is unnecessary for processing, such as patches of sky to the left or right of the chimney. Similarly, as relevant events only occur when a swift has immediately disappeared into the chimney, area outside of the top edge of the chimney is unnecessary for detecting this type of event. Thus, prior to processing, two smaller regions of interest within the frame are defined within this initialization stage. Because the chimney itself is the focus of these regions of interest, each region will be

defined as a function of the chimney's dimensions. An assumption is made that the camera is placed relative to the chimney such that two distinct chimney corners are present. Manual input is used to identify these corners, which are then used to determine the dimensions of the chimney, and thus the regions of interest. No additional manual input is needed for the remainder of the approach.

The first region of interest is rectangular and defines how the frame is cropped to eliminate areas of the frame which are not vital to swift tracking. The boundaries of the region of interest are determined as offsets from the two selected chimney corners, with offset values calculated as a function of the chimney width. The offsets are chosen such that the cropped region has a consistent height to width ratio of 1:2 regardless of the size of the chimney itself, and that the chimney is sufficiently centered within the cropped frame. This cropped frame is then resized to the fixed dimensions of 150 x 300 pixels to ensure some consistency in scale for the content of the frame.

Figure 4: *Color-coded demonstration of first region-of-interest determination.* ***White:*** *Chosen points representing corners of chimney.* ***Red:*** *Horizontal width between two points.* ***Yellow:*** *Offset from points determined as multiples of the chimney width.*

As the shape of the chimney perimeter is not guaranteed to be rectangular, a more complex approach is required to determine the region of interest that detects swifts entering chimneys. To minimize manual input, an automatic chimney segmentation approach is used. First, a rectangular area is defined using corner offsets to enclose the top edge of the chimney. Then, within this rectangular region, a median blur is applied to remove extraneous details found on the walls of the chimney, while still maintaining a contrasting edge between chimney and sky. The resulting cropped and blurred image is then automatically thresholded on its blue channel, which was found to be effective at separating a chimney from a background of sky across varied environmental conditions. Finally, the edge of this thresholded image is used in combination with an upwards-offset dilation operation to generate a mask which contains a detailed region of interest for the top edge of the chimney.

Figure 5: *Stages of automatic determination of second chimney region-of-interest.* ***(A)*** *Original cropped region isolating perimeter of chimney,* ***(B)*** *Bilateral filtering,* ***(C)*** *Thresholding on B channel,* ***(D)*** *Edge determination,* ***(E)*** *Upwards-offset dilation,* ***(F)*** *Overlay of region-of-interest on original frame.*

3.2 Swift Localization

Swifts are localized within a frame using a combination of preprocessing and motion-based segmentation, and grayscale filtering. The preprocessing stage is brief, consisting of grayscale conversion, as well as cropping and resizing to the dimensions determined in the initialization stage. Following preprocessing, frames are batch processed using the Inexact Augmented Lagrangian Multiplier (IALM) approximation of the Robust Principal Component Analysis (RPCA) approach to background subtraction. This approach was chosen because initial tests

for a second approach, Mixture of Gaussians (MOG), showed over-segmentation of swift objects. Additionally, available Python implementations of MOG background subtraction proved to be inflexible, as output consists solely of a binary mask, which restricts any additional grayscale processing. Available implementations of RPCA, on the other hand, provide grayscale output images which can be filtered further prior to binarization. Also of note are alternate approximations to the RPCA approach, however available implementations for these exist solely in MATLAB at the time of writing [26]. The chosen approach is applied in batches of 21 frames as to continually update the background model across an entire video. The resulting detected motion is then smoothed using a bilateral filter to preserve contrasting edges between swifts. A global threshold, manually set to a value of 15, is then applied to remove low-intensity motion. Once thresholded, a 3x3 opening operation is applied to the grayscale image to remove small non-swift segments. Finally, the image is segmented using a connected component labelling stage. Both the resulting segmented image and the previous frame's segmented image are then passed on to a segment tracking stage.

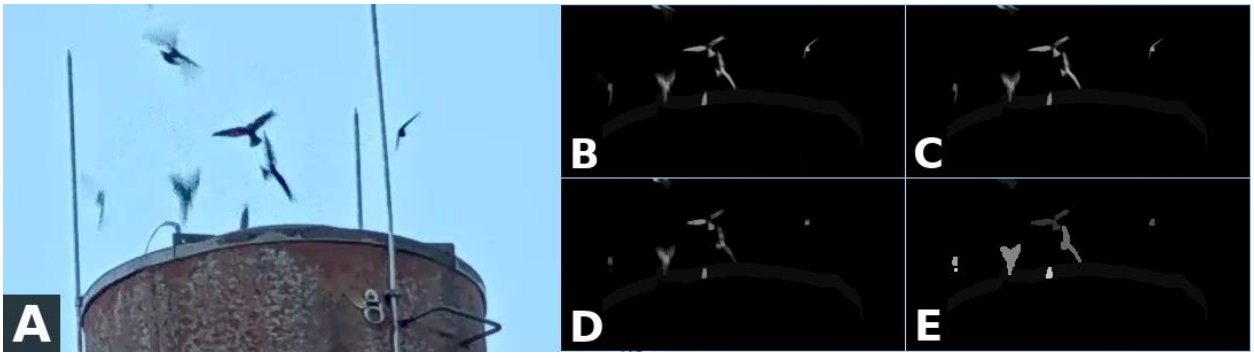

Figure 6: *Swift localization process.* ***(A)*** *Example of original frame,* ***(B)*** *Output of RPCA motion estimation stage,* ***(C)*** *Global threshold with intensity of 15,* ***(D)*** *3x3 opening operation,* ***(E)*** *Connected component labeling.*

3.3 Swift Tracking

For the chosen method, accurately linking segments between a pair of consecutive frames will be formulated as a standard assignment problem. For this formulation, however, the typical cost matrix is modified to allow for the possibility of leaving the frame (for previous frame segments) and entering the frame (for current frame segments). Thus, using the Hungarian algorithm, each segment across both frames will be simultaneously assigned either a matching segment from the other frame, or an outcome of having no match.

$$M_{cost} = \begin{bmatrix} C_{1a} & 1+e & C_{11} & C_{12} & C_{13} \\ 1+e & C_{2a} & C_{21} & C_{22} & C_{23} \\ 1+e & 1+e & C_{d1} & 1+e & 1+e \\ 1+e & 1+e & 1+e & C_{d2} & 1+e \\ 1+e & 1+e & 1+e & 1+e & C_{d3} \end{bmatrix} \qquad M_{cost} = \begin{bmatrix} 1 & 1+e & \mathbf{0.023} & 1067 & 5.6e7 \\ 1+e & 1 & 6.2e5 & 2.2e8 & \mathbf{0.005} \\ 1+e & 1+e & 1 & 1+e & 1+e \\ 1+e & 1+e & 1+e & \mathbf{1} & 1+e \\ 1+e & 1+e & 1+e & 1+e & 1 \end{bmatrix}$$

Figure 7: *Formulation of cost matrix for example of 2 segments in current frame matched with 3 segments in previous frame.* ***Left:*** *Elements are color-coded based on the outcome they represent, namely segments matching (blue), segments appearing (green), segments disappearing (red), and null outcomes (grey).* ***Right:*** *Sample matrix with typical values. Relevant elements of the minimum-cost solution have been bolded, signifying two matches and one segment having disappeared.*

Construction of the cost matrix is key for allowing for this kind of matching between segments. The cost matrix is initialized as an (M+N) by (M+N) matrix, where M corresponds to the number of segments detected in the current frame, and N corresponds to the number of segments detected in the frame immediately previous. Diagonal elements of this matrix correspond to the outcome of no match. Hence, the first M diagonal elements correspond to a current-frame segment appearing, and the last N diagonal elements correspond to a previous-frame segment disappearing. These elements will be set to a value of 1, chosen to simplify the cost functions for matches which are demonstrated further on. The top-right M by N grid of elements in the matrix correspond to the costs associated with matches between the M segments in the current frame and the N segments of the previous frame. These costs will be

determined as a function of features extracted from each pair of segments. From a semantic perspective, a cost with a value above 2 represents that the match between the pair of segments is less likely (has higher total cost) than each individual segment having no match. Conversely, a value below 2 represents that a match between the pair of segments is more likely (has lower total cost) than each individual segment having no match. Thus, any mappings between segment features and costs must be tailored to adhere to this semantic meaning. All other elements of the cost matrix represent assignments with no meaning and are initialized to values such that it is impossible for those assignments to be chosen. By choosing 1 + epsilon, it is ensured that “no match” outcomes will be chosen over any null elements when minimizing cost across all assignments.

$$C_{\alpha\beta} = 0.5 * (C_{dist,\,\alpha\beta} + C_{ang,\,\alpha\beta}) \tag{1}$$

Two features are extracted from each pair of segments and mapped to cost values, which are then averaged and used as a cost matrix element. This is demonstrated by Equation 1. The first feature is straightforward and involves calculating the distance between segment centroids. Experimentally, under the fixed dimensions to which frames are resized to in preprocessing, it was determined that true matches have typical distances between 0 and 25 pixels. The second feature is conditioned on whether the “previous frame” segment of the pair had itself been matched prior. If not, then only distance costs will factor into the cost calculation. But if so, then the angle of the previous match is compared to the angle of the potential match, and the deviation is computed. Typical angle deviations for true matches were found to be no greater than 90 degrees. Functions have been constructed to map these typical angle deviation and distance values to costs below a value of 2, and to map atypical values above a value of two. These functions are provided in Equation 2 and Equation 3.

$$C_{dist,\,\alpha\beta} = 2e^{(\Delta D - 25)} \tag{2}$$

$$C_{ang,\,\alpha\beta} = 2e^{(\Delta A - 90)} \tag{3}$$

These cost functions are used to fill the elements of the cost matrix, as displayed in Figure 7. By applying the Hungarian algorithm to the resulting cost matrix, a label of “match” or “no match” can be applied to each segment. Depending on the label applied to each segment, they will be processed differently in further steps. In the case of a current-frame segment marked as having no match (appearing), its centroid coordinates are stored for subsequent matching stages. In the case that a current-frame segment is determined to have a match, then its centroid coordinate is appended to a list containing any previous centroid coordinates and stored with that segment so that the centroid history can be retained. If a previous frame is marked as having no match, then it has disappeared, and it is further analyzed to determine whether this disappearing segment should be counted as a positive event. This is done through a two-stage approach: event detection using two simple criteria, and event classification using motion vector angle features.

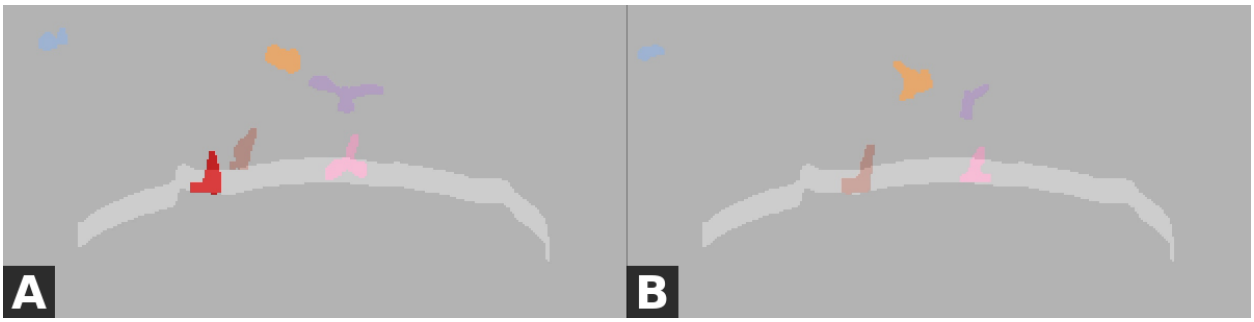

Figure 8: *Example of segment matching between consecutive frames.* ***(A)*** *Previous frame, containing one segment (colored red) which is not present in the current frame. This segment is within the region-of-interest, so it will be caught by the event detection stage.* ***(B)*** *Current frame, containing color-coded matches with segments in previous frame.*

3.4 Event Detection

Merely marking a segment as having disappeared from frame is not enough to accurately identify instances of swifts entering chimneys. With this approach, false positives would occur when swifts disappear by flying behind a foreground object, such as the chimney or another swift. Additionally, if a chimney swift happens to be over-segmented into distinct body and wing segments, but is correctly segmented in subsequent frames, then extraneous segments will be marked as having disappeared too. Because of the presence of these behaviors, additional measures are needed to exclude them from being considered as potential events. Thus, a segment which has disappeared will be considered a candidate event only if it meets two simple criteria. The first requires a segment's centroid to have been within the chimney's top edge region-of-interest when the segment disappears. The second requires that segment to have at least one instance of previous matching, which ensures that an angle feature can be calculated for the classification stage. Events which meet both criteria are stored until the entire video has been processed. Classification of these events will be performed once all candidate events are collected.

3.5 Event Classification

Intuitively, birds which fly into a vertical chimney must have motion paths that point downwards into that chimney. Thus, it was initially considered that typical flight angles of true detections would span a range symmetrical about -90 degrees. In practice, however, symmetry was found not to be guaranteed. Within the chosen experimental testing videos, swifts were found to fly into chimneys at a slight tilt such that the mode of the range of angles is offset from

-90 degrees by as much as ±20 degrees. Groups of swifts are found to maintain this offset throughout the duration of entire videos. Thus, a modification is proposed to instead dynamically determine classification boundaries to account for this offset. A mode calculation is performed by binning angle data and utilizing counts from the most populous bins as shown in Equation 4. If the calculated mode represents a realistic offset, the range of acceptable angles will be shifted according to this offset. Only modes within the range of -70 to -110 degrees are considered as realistic for this adjustment. Otherwise, the symmetrical decision boundaries of -120 degrees and -60 degrees are chosen by default, with angles in between marked as positive.

$$(\breve{x}) = x_l + \frac{f_0 - f_{-1}}{2f_0 - f_1 - f_{-1}} \times w \tag{4}$$

One final modification is made to this approach to address an easily detectable instance of false positives. Due to the nature of using 3x3 SE opening as the final step of the localization stage, orphaned 3-pixel-width segments may remain due to noise or over-segmentation. If a match is made between two close 3-pixel width segments, the angle of the vector connecting them will commonly be exactly a multiple of 15 degrees. Angles as precise as these are extremely rare for true swifts, as swifts will typically generate segments larger than width 3, resulting in more complex values for flight angles. Thus, these precise angles are excluded prior to mode calculation.

4 Experimental Evaluation

4.1 Dataset Descriptions

Datasets used to evaluate this approach come from two different sources, recorded over varying years of observation. The first source is from a camera system installed in Sault Ste. Marie, Ontario, directed at a post office chimney. The second source is from a camera system installed in Chalk River, Ontario, and is directed at a decommissioned nuclear power plant chimney stack. A direct comparison between the two video sources is provided in Table 1.

Table 1: Overview of source locations for swift datasets.

	Sault Ste. Marie, ON	**Chalk River, ON**
Frame Rate	18-30 FPS (Variable)	60 FPS (Constant)
Resolution	1080 x 1920 pixels	1080 x 1920 pixels
Chimney Width	172 pixels	340 pixels
Swift Width	10-15 pixels	30-50 pixels
Swift Behaviors	Entering and exiting chimney	Entering chimney only
Other Notes	• Higher incidence of non-swift birds (seagulls, crows) • Presence of video artifacts (motion trails)	• Rounded chimney shape

From these two video sources, ground truth annotations were created for three total videos. Annotations are written such that the total number of swifts entering the in-frame chimney are recorded for each individual frame of the video. The videos were chosen so that several different conditions in weather and lighting could be represented. Details relating to the specific videos are displayed in Table 2.

Table 2: Overview of chosen evaluation datasets.

	May 18, 2017 (Sault Ste. Marie)	**June 13th, 2019 (Chalk River)**	**June 14th, 2019 (Chalk River)**
Duration	00hr 05m (Partial Video)	01hr 00m (Full Video)	00hr 30m (Full Video)
Total Frames	7200	216,096	108,048
Total GT Count	726 swifts	460 swifts	541 swifts
Conditions	● Heavy cloud cover ● Rain on lens ● Motion trail artifacts	● Heavy fog, near-white sky ● High wind, chimney shaking motion	● Light clouds ● Noisy motion artifacts

4.2 Experimental Results

Results were determined during tests performed on a desktop computer running Linux Mint 19.1 containing 32GB of RAM and an Intel Core i7-4770 CPU. Ground truth annotations were written prior to testing; however, a modification is proposed during runtime. Due to the nature of high frame-rate video, it often takes several frames for a chimney swift to enter a chimney. Because of this, ambiguity arises for the precise frame at which a chimney swift enters a chimney. This can lead to “off-by-one frame” errors where the algorithm correctly identifies an event but registers that event at a different timestamp than the one identified during manual annotation. For 30FPS and 60FPS video files, this timestamp difference is merely 1/30th or 1/60th of a second off from the manual annotation. As these are not true errors, they are corrected prior to precision and recall calculations through automatic adjustments of the ground truth annotations.

Detailed descriptions of the performance of both detection and classification stages is provided in Table 3 and Table 4 respectively. Results across both stages are compared next,

with overall counts covered in Table 5, while performance metrics (precision, recall, F1 score) are covered in Table 6. Precision is calculated as the ratio between true positives and the sum of all positives. Recall is calculated as the ratio between true positives and the sum of true positives, false negatives from the classification stage, and missed events from the detection stage. The F1 score is calculated as the harmonic mean between precision and recall.

Table 3: Detailed performance of event detection stage.

Video	Total Swifts to Detect	Detected Events	True Positives	False Positives	Missed Detections
May 18 (Ste. Saint Marie)	726	1049	659 (of 726)	390	67 (of 726)
June 13 (Chalk River)	457	823	434 (of 457)	389	23 (of 457)
June 14 (Chalk River)	534	858	475 (of 534)	383	59 (of 858)

Table 4: Detailed performance of event classification stage.

Video	Detected Events	Labeled Positive	True Positive	False Positive	Labeled Negative	True Negative	False Negative
May 18	1049	715 (of 1049)	648	67	334 (of 1049)	323	11
June 13	823	500 (of 823)	415	85	323 (of 823)	304	19
June 14	858	485 (of 858)	439	46	373 (of 858)	337	36

Table 5: Comparison of total counts between stages.

		Detection Stage Only		Detection and Classification	
Video	**Actual Count**	**Predicted Count**	**Difference**	**Predicted Count**	**Difference**
May 18	726	1049	323	715	11
June 13	457	823	366	500	43
June 14	534	858	324	485	49

Table 6: Comparison of performance metrics between stages.

	Detection Stage Only			Detection and Classification		
Video	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
May 18	0.6282	0.9077	0.7425	0.9063	0.8926	0.8994
June 13	0.5273	0.9497	0.6781	0.8300	0.9081	0.8673
June 14	0.5536	0.8895	0.6825	0.9052	0.8221	0.8617
AVG	**0.5697**	**0.9156**	**0.7010**	**0.8805**	**0.8743**	**0.8761**

4.3 Discussion of Results

Results of the detection stage alone demonstrate strong ability to detect swifts entering chimneys, with recall rates averaging 0.9156. Despite strong recall rates, precision rates are poor for this stage, with an average of 0.5697. A high rate of false positives was expected, however, as described in the reasons justifying a classification stage found in Section 3.4. Those reasons, which include over-segmentation, noise, and swifts flying behind chimneys, are all present as sources of error in these experimental tests as well. Aside from false positives, missed swifts largely resulted from overlapping between bird segments. Both complete occlusions, as well as the joining of segments close in proximity, can cause motion paths of

individual segments to end prematurely. Hence, this can cause segments to fail to meet the two criteria of “segment which has disappeared within ROI” and “segment which has disappeared with a history of at least one previous match.” Of note, occlusion rates are variable between videos, with occlusions most present in the June 14th Chalk River video dataset. In that video, most present events are concentrated within two short periods of time spanning roughly 50 seconds each. In other videos, however, events are spread out over longer periods, resulting in less clutter and making occlusions less likely. A separate source of error, yet far less frequent than occlusions, came from swifts that were missed due to large between-frame displacement. A combination of high flight velocity and low frame rate can cause swifts to travel further in one frame than the chimney’s entire region of interest, thus evading detection. This source of error was found only in the Sault Ste. Marie video, which has a low frame rate that varies between 18 and 30 FPS. Finally, one other infrequent source of missed detection arises when swifts hover at the edge of the chimney prior to entering. This hovering behavior violates an assumption made in the detection stage: that swifts will travel with some displacement and with motion like that of previous motion. Without this assumption, the chosen cost function will prevent correct matches from being made, thus breaking up the hovering swift’s motion path prematurely.

When the angle classification is added following the event detection stage, results improve dramatically. Average precision rates increase from 0.5697 to 0.8805, with average recall rates only decreasing from 0.9156 to 0.8743. This results in an average F1 score improvement from 0.7010 to 0.8791 and demonstrates validity in the notion that an angle feature is discriminative between true and false events across a variety of video conditions. However, as with the detection stage alone, inspecting sources of error is key to understanding potential areas of improvement for future work. Examining sources of false positive events demonstrates differing primary causes between videos. The June 13th Chalk River video, for example, contained instances of sustained periods where swifts would circle and dive behind

the chimney. This resulted in birds which were correctly segmented and tracked, but which flew behind the chimney with an angle that would be typical for swifts entering the chimney. The May 18th Ste. Saint Marie video had a different primary source of error, since that video contains swifts which exit the chimney in addition to those entering the chimney. Swifts which exit commonly fly immediately back down the outside of the chimney. From the camera position chosen for these datasets, this behavior is indistinguishable from birds which exit the chimney then immediately reenter, hence this being a source of false positives. Other sources of error include over-segmentation of swifts, and swifts which hover near the edge of a chimney. Both cause multiple detections for a single bird but were infrequent relative to the previously mentioned sources of error. Finally, non-bird swifts such as crows or seagulls flying near the edge of the chimney were a source of false positives but were infrequent throughout each test video. If this approach were to be applied to videos with a higher prevalence of non-swift birds, false positive rates could become a concern.

Compared to false positives rates, false negative occurrences were relatively infrequent. When false negatives did occur, it was found to be largely due to the limitations of using the centroids of segments to represent swifts. If a segment is partially obscured or is segmented poorly, its centroid may shift dramatically from its body to one of its wings. With such a shift, the angle may be offset so greatly that it no longer falls within the typical range of swift angles. This was most common in the June 14th Chalk River dataset, which has a high frame rate of 60 FPS, as well as more occurrences of birds flying slowly into the chimney. These factors make it more likely for a frame to contain a swift partially obscured by the chimney itself as it enters, which could result in a centroid shift between frames. A second, less frequent source of error occurs when swifts which begin flying horizontally but switch directions mid-flight to enter the chimney. As angle features are calculated using the first and last centroids of the motion path, this change in direction is not adequately captured for the classification stage. Finally, identity

transfer may occur in cluttered scenes if birds flying in different directions cross paths. While this is unlikely due to the influence of flight angle on matching cost functions, it did occur on some occasions, but rarely so.

Moving on from error analysis, the hand-crafted decision boundary with variable offset performed well for each test video. Across all three datasets, there was a distinct mode representing the peak of the distribution of typical flight angles, and each mode was significantly offset from -90 degrees. Incorporating this offset allowed for decision boundaries that were more effective than a default range symmetrical about -90 degrees. However, each decision boundary was still handcrafted to some degree, and none stemmed from supervised learning techniques. This may cause issues for unseen datasets, as the proposed approach to statistical property estimation may be too inflexible to hold if ergodicity is not guaranteed. For example, there may be cases where the angle offset is not consistent throughout an entire video, or cases where the false positive rate is high enough to prevent there from being a distinct angle mode.

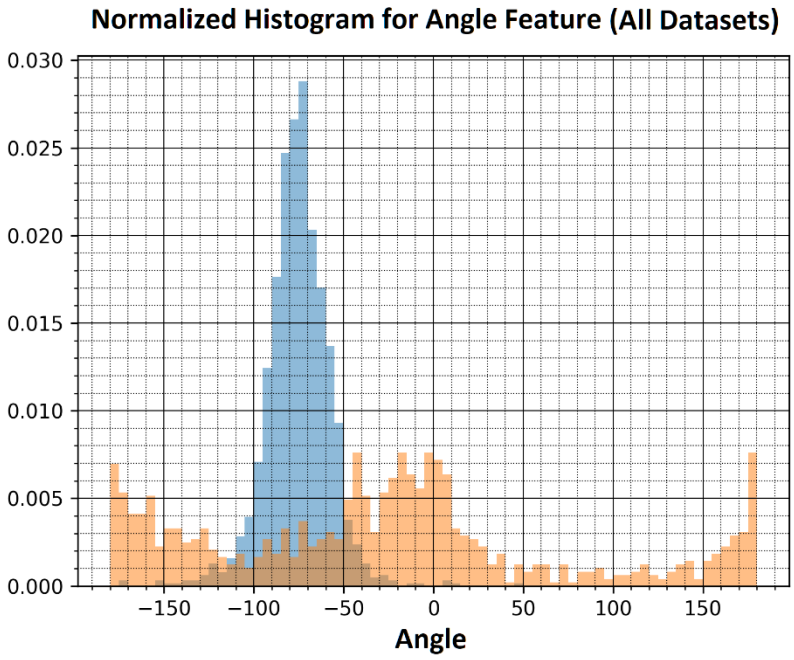

Figure 9: *Angle feature histogram comparing true detections versus false detections across all three video datasets.* ***Blue****: true detections,* ***Orange:*** *false detections.*

5 Conclusion

This paper proposes a method for automatically tracking chimney swifts and detecting instances of swifts entering chimneys. Experimentation evaluation of the approach showed promising results across video datasets with varied frame rates, swift behaviors, and visual qualities. This suggests validity in the following ideas: that swifts can be segmented from real-world surveillance video datasets using motion analysis; that segmented swifts can be effectively tracked using only distance and angle information; and that a region-of-interest combined with an angle feature can effectively detect instances of swifts entering chimneys. This approach provides clear improvements over previous efforts to monitor chimney swifts and is likely applicable to behavioral tracking of other animal species. It demonstrates that object feature descriptors which can distinguish between individual objects are not necessarily needed to sufficiently identify and track an object, even in high-clutter scenes. This, however, is under the assumption that each object can be consistently segmented from its background, which is not guaranteed with backgrounds more complex than sky.

6 Future Work

Considering the limited extent of experimental evaluation described within this paper, additional validation of the approach is strongly encouraged. Choosing and annotating datasets was limited due to time constraints, however exploring additional videos may provide a more complete coverage of varying swift behaviors and visual characteristics of scenes. Tests on videos which contain non-swift object motion (such as seagulls or crows) would be desirable, as would videos containing more complex background compositions. Additionally, ensuring the approach is robust to changes in illumination and scale would be ideal. This could be done by considering video datasets from additional chimney sources, as this may provide a more

complete range of conditions. Finally, a more thorough evaluation of classification features is desired, which could include modeling the motion path using polynomial fitting rather than a single angle feature. The way in which the classification decision boundary is determined could also be improved upon using supervised learning techniques.

Beyond further validating the proposed approach, potential modifications could address the sources of error outlined in Section 4.3. Numerous errors resulted from the fact that tracking segments is done on a frame-by-frame basis, with each segment assumed to represent a single chimney swift during tracking stages. This assumption breaks down in cases where each segment does not represent a single chimney swift, such as when overlapping of swifts occurs. Choosing a more sophisticated segmentation approach may address this problem, such as the recursive intensity-based thresholding technique employed by Groom et al. [13] In other cases, this could be addressed by modifying the segment approach to split apart overlapping swifts. In these scenarios, approaches from both Branson et al. [18] and Tidmerng et al. [21] could be of use. An alternative approach would be the departure from frame-by-frame linking, such as the tracklet-linking described by Huang et al. [23]. This may allow for tracking despite overlapping motion paths. Each of these approaches, however, may risk impacting the performance of segments that were correctly segmented and tracked by the proposed approach of this paper.

References

[1] J. Brown and S. D. Gehrt, “The Basics of Using Remote Cameras to Monitor Wildlife,” p. 8, 2009.

[2] L. Purves, “2016 Ontario SwiftWatch Report,” Bird Studies Canada, Feb. 2019 [Online]. Available: https://www.birdscanada.org/volunteer/ai/resources/2018%20Ontario%20Summary%20Report_EN.pdf. [Accessed: 16-Aug-2019]

[3] “Chimney swift | Ontario.ca.” [Online]. Available: https://www.ontario.ca/page/chimney-swift. [Accessed: 16-Aug-2019]

[4] “Protocol and Data Form for Volunteers.” Mar-2019 [Online]. Available: https://www.birdscanada.org/volunteer/ai/resources/Ontario_Swiftwatch_Protocol.pdf. [Accessed: 16-Aug-2019]

[5] “Mersey Tobeatic Research Institute (MTRI) : Institut de recherche du Mersey Tobeatic.” [Online]. Available: http://www.merseytobeatic.ca/projects-forest-chimney-swifts.php. [Accessed: 16-Aug-2019]

[6] C. Timm, *Detecting and counting swift birds who nest in chimneys in Ontario using OpenCV.: colbytimm/SwiftWatch*. 2018 [Online]. Available: https://github.com/colbytimm/SwiftWatch. [Accessed: 23-May-2019]

[7] “David Ross - Incremental Visual Tracking.” [Online]. Available: http://www.cs.toronto.edu/~dross/ivt/. [Accessed: 17-Aug-2019]

[8] Yao Zhou and Jiebo Luo, “A practical method for counting arbitrary target objects in arbitrary scenes,” in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.

[9] R. T’Jampens, F. Hernandez, F. Vandecasteele, and S. Verstockt, “Automatic detection, tracking and counting of birds in marine video content,” in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016, pp. 1–6.

[10] J. Y. Tou and C. C. Toh, “Optical Flow-Based Bird Tracking and Counting for Congregating Flocks,” in *Intelligent Information and Database Systems*, 2012, pp. 514–523.

[11] A. S. Laliberte and W. J. Ripple, “Automated Wildlife Counts from Remotely Sensed Imagery,” *Wildl. Soc. Bull. 1973-2006*, vol. 31, no. 2, pp. 362–371, 2003.

[12] A. Dash and A. B. Albu, “Counting Large Flocks of Birds Using Videos Acquired with Hand-Held Devices,” in *Advanced Concepts for Intelligent Vision Systems*, 2017, pp. 468–478.

[13] G. Groom, I. K. Petersen, M. D. Anderson, and A. D. Fox, “Using object-based analysis of image data to count birds: mapping of Lesser Flamingos at Kamfers Dam, Northern Cape, South Africa,” *Int. J. Remote Sens.*, vol. 32, no. 16, pp. 4611–4639, Aug. 2011.

[14] M. Shakeri and H. Zhang, “Real-time bird detection based on background subtraction,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 4507–4510.

[15] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Comput. Sci. Rev.*, vol. 11–12, pp. 31–66, May 2014.

[16] S. Descamps, A. Béchet, X. Descombes, A. Arnaud, and J. Zerubia, “An automatic counter for aerial images of aggregations of large birds,” *Bird Study*, vol. 58, no. 3, pp. 302–308, Aug. 2011.

[17] G. Perrin, X. Descombes, and J. Zerubia, “2D and 3D Vegetation Resource Parameters Assessment using Marked Point Processes,” in *18th International Conference on Pattern Recognition (ICPR’06)*, 2006, vol. 1, pp. 1–4.

[18] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson, “High-throughput ethomics in large groups of Drosophila,” *Nat. Methods*, vol. 6, no. 6, pp. 451–457, Jun.

2009.
[19] T. Fukunaga, S. Kubota, S. Oda, and W. Iwasaki, “GroupTracker: Video tracking system for multiple animals under severe occlusion,” *Comput. Biol. Chem.*, vol. 57, pp. 39–45, Aug. 2015.
[20] J. Zhang, Q. Xu, X. Cao, P. Yan, and X. Li, “Hierarchical incorporation of shape and shape dynamics for flying bird detection,” *Neurocomputing*, vol. 131, pp. 179–190, May 2014.
[21] N. Tidmerng, W. Songpan, and M. Wattana, “Solving bird image overlapping for automatic population counts of birds using image processing,” in *2016 Management and Innovation Technology International Conference (MITicon)*, 2016, p. MIT-84-MIT-87.
[22] A. Pinto, A. Moreira, P. Costa, and M. Correia, “Revisiting Lucas-Kanade and Horn-Schunck,” *J. Comput. Eng. Inform.*, vol. 1, pp. 23–29, Apr. 2013.
[23] C. Huang, Y. Li, and R. Nevatia, “Multiple Target Tracking by Learning-Based Hierarchical Association of Detection Responses,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 898–910, Apr. 2013.
[24] S. Ali and M. Shah, “Floor Fields for Tracking in High Density Crowd Scenes,” in *Computer Vision – ECCV 2008*, 2008, pp. 1–14.
[25] A. Dehghan and M. Shah, “Binary Quadratic Programing for Online Tracking of Hundreds of People in Extremely Crowded Scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 568–581, Mar. 2018.
[26] “Low-Rank Matrix Recovery and Completion via Convex Optimization.” [Online]. Available: https://people.eecs.berkeley.edu/~yima/matrix-rank/sample_code.html. [Accessed: 27-Aug-2019]