Article

Application of Digital Particle Image Velocimetry to Insect Motion: Measurement of Incoming, Outgoing, and Lateral Honeybee Traffic

Sarbjit Mukherjee * and Vladimir Kulyukin *

Department of Computer Science, Utah State University, 4205 Old Main Hill, Logan, UT 84322-4205, USA

* Correspondence: sarbjit.usu@gmail.com (S.M.); vladimir.kulyukin@usu.edu (V.K.)

Received: 23 January 2020; Accepted: 11 March 2020; Published: 18 March 2020

Abstract: The well-being of a honeybee (*Apis mellifera*) colony depends on forager traffic. Consistent discrepancies in forager traffic indicate that the hive may not be healthy and require human intervention. Honeybee traffic in the vicinity of a hive can be divided into three types: incoming, outgoing, and lateral. These types constitute directional traffic, and are juxtaposed with omnidirectional traffic where bee motions are considered regardless of direction. Accurate measurement of directional honeybee traffic is fundamental to electronic beehive monitoring systems that continuously monitor honeybee colonies to detect deviations from the norm. An algorithm based on digital particle image velocimetry is proposed to measure directional traffic. The algorithm uses digital particle image velocimetry to compute motion vectors, analytically classifies them as incoming, outgoing, or lateral, and returns the classified vector counts as measurements of directional traffic levels. Dynamic time warping is used to compare the algorithm's omnidirectional traffic curves to the curves produced by a previously proposed bee motion counting algorithm based on motion detection and deep learning and to the curves obtained from a human observer's counts on four honeybee traffic videos (2976 video frames). The currently proposed algorithm not only approximates the human ground truth on par with the previously proposed algorithm in terms of omnidirectional bee motion counts but also provides estimates of directional bee traffic and does not require extensive training. An analysis of correlation vectors of consecutive image pairs with single bee motions indicates that correlation maps follow Gaussian distribution and the three-point Gaussian sub-pixel accuracy method appears feasible. Experimental evidence indicates it is reasonable to treat whole bees as tracers, because whole bee bodies and not parts thereof cause maximum motion. To ensure the replicability of the reported findings, these videos and frame-by-frame bee motion counts have been made public. The proposed algorithm is also used to investigate the incoming and outgoing traffic curves in a healthy hive on the same day and on different days on a dataset of 292 videos (216,956 video frames).

Keywords: digital particle image velocimetry; video processing; image processing; electronic beehive monitoring; insect traffic; honeybee traffic; dynamic time warping

1. Introduction

A honeybee (*Apis mellifera*) colony consists of female worker bees, male bees (drones), and a single queen [1]. The number of worker bees varies with the season from 10,000 to 40,000. The number of drones also depends on the season and ranges from zero to several hundred. Each colony has a structured division of labor [2]. The queen is the only reproductive female in the colony. Her main duty is to propagate the species by laying eggs. When the queen is several days old, she takes flight to mate in the air with drones from other hives. When she returns to the hive, she remains fertilized for

three to four years. A prolific queen may lay up to 3000 eggs per day [1]. The drones are male bees. They have no sting and no means of gathering nectar, secreting wax, or bringing water. Their sole role is to mate with queens on their bridal trip and instantly die in the act of copulation. Only one in a thousand gets a chance to mate [1]. All day-to-day colony work is done by the female worker bees. When workers emerge from their cells, they start cleaning cells in the brood nest. After a week, they feed and nurse larvae. The nursing assignment lasts one or two weeks and is followed by other duties such as cell construction, receiving nectar from foragers, and guarding the hive's entrance. In their third or fourth week, the female bees become foragers. There are three kinds of foragers: nectar foragers, pollen foragers, and water foragers. Page [3] argued that pollen foraging is a function of the number of cells without pollen encountered on the boundary cells and the pollen cells. Thus, if a forager encounters more empty pollen cells than some value representative of her response threshold, she leaves the hive for another pollen load. If, on the other hand, she encounters fewer empty cells, she stops foraging for pollen and engages in nectar or water foraging. The foragers live 5–6 weeks and do not perform their previous duties within the hive.

The well-being of a honeybee colony in a hive critically depends on robust forager traffic. Repeated discrepancies or disruptions in forager traffic indicate that the hive may not be healthy. Thus, accurate measurement of honeybee traffic in the vicinity of the hive is fundamental for electronic beehive monitoring (EBM) systems that continuously monitor honeybee colonies to detect and report deviations from the norm. Our EBM research (e.g., [4–6]), during which we watched hundreds of field-captured honeybee traffic videos of three different bee races (Carniolan, Italian, and Buckfast), leads us to the conclusion that honeybee traffic in the vicinity of a Langstroth hive [7] can be divided into three distinct classes: *incoming*, *outgoing*, and *lateral*. Incoming traffic consists of bees entering the hive either through the landing pad or holes drilled in supers. Outgoing traffic consists of bees leaving the hive either from the landing pad or from holes drilled in supers. Lateral traffic consists of bees flying more or less in parallel to the front side of the hive (i.e., the side with the landing pad). We refer to these three types of traffic as *directional* traffic and juxtapose it with *omnidirectional* traffic [6] where bee motions are detected regardless of direction.

In this investigation, we contribute to the body of research on insect and animal flight by presenting a bee motion estimation algorithm based on digital particle image velocimetry (DPIV) [8] to measure levels of directional honeybee traffic in the vicinity of a Langstroth hive. The algorithm does not compute honeybee flight paths or trajectories, which, as our longitudinal field deployment convinced us [4,6], is feasible only at very low levels of honeybee traffic. Rather, the proposed method uses DPIV to compute motion vectors, analytically classifies them as incoming, outgoing, or lateral, and uses the classified vector counts to measure directional honeybee traffic.

The remainder of our article is organized as follows. In Section 2, we present related work and connect this investigation to our previous research. In Section 3, we briefly describe the hardware we built and deployed on Langstroth hives to capture bee traffic videos for this investigation and give a detailed description of our algorithm to measure directional bee traffic. In Section 4, we present the experiments we designed to evaluate the proposed algorithm. In Section 5, we give our conclusions.

2. Related Work

Flying insects and animals, as they move through the air, leave behind air motions which, if accurately captured, can characterize insect and animal flight patterns. The discovery of digital particle image velocimetry (DPIV) [8,9] enabled scientists and engineers to start investigating insect and animal flight topologies. Reliable quantitative descriptions of such flight topologies not only improve our understanding of how insects and animals fly and capture their flight patterns but may, in time, result in better designs of micro-air vehicles. For example, Spedding et al. [10] performed a series of experiments with DPIV to measure turbulence levels in a low turbulence wind tunnel and argued that DPIV can, in principle, be tuned to measure the aerodynamic performance of small-scale flying devices.

Several researchers and research groups have used DPIV to investigate various aspects of animal and insect flight. Henningsson et al. [11] used DPIV to investigate the vortex wake and flight kinematics of a swift in cruising flight in a wind tunnel. The researchers recorded the swift's wingbeat kinematics by a high-speed camera and used DPIV to visualize the birds' wake. Hedenström et al. [12] showed that the wakes of a small bat species are different from the wakes of birds by using DPIV to visualize and analyze wake images. The researchers discovered that in the investigated bat species each wing generated its own vortex loop and the circulation on the outer wing and the arm wing differed in sign during the upstroke, which resulted in negative lift on the hand wing and positive lift on the arm wing.

Spedding et al. [13] used DPIV to formulate a simple model to explain the correct, measured balance of forces in the downstroke- and upstroke-generated wake over the entire range of flight speeds in thrush nightingales. The researchers demonstrated the feasibility of tracking the momentum in the wake of a flying animal. DPIV measurements were used to quantify and sum disparate vortices and to find that the momentum is not entirely contained within the coarse vortex wake. Muijres et al. [14] used DPIV to demonstrate that a small nectar-feeding bat increases lift by as much as 40% using attached leading-edge vortices during slow forward flight. The researchers also showed that the use of unsteady aerodynamic mechanisms in flapping flight is not limited to insects but is also used by larger and heavier animals. Hubel et al. [15] investigated the kinematics and wake structure of lesser dog-faced fruit bats (*Cynopterus brachyotis*) flying in a wind tunnel. The flow structure in the spanwise plane perpendicular to the flow stream was visualized using time-resolved PIV. The flight of four bats was investigated to reveal patterns in kinematics and wake structure typical for lower and higher speeds.

Dickinson et al. [16] were the first to use DPIV to investigate insect flight. The researchers used DPIV to measure fluid velocities of a fruit fly and determine the contribution of the leading-edge vortex to overall force production. Bomphrey et al. [17] did the first DPIV analysis of the flow field around the wings of the tobacco hawkmoth (*Manduca sexta*) in a wind tunnel. During the late downstroke of *Manduca*, the flow was experimentally shown to separate at, or near, the leading edge of the wing. Flow separation was associated with a saddle between the wing bases on the surface of the hawkmoth's thorax. Michelsen [18] used DPIV to examine and analyze sound and air flows generated by dancing honeybees. DPIV was used to map the air flows around wagging bees. It was discovered that the movement of the bee body is followed by a thin (1–2 mm) boundary layer and other air flows lag behind the body motion and are insufficient to fill the volume left by the body or remove the air from the space occupied by the bee body. The DPIV analysis showed that flows collide and lead to the formation of short-lived eddies.

Several electronic beehive monitoring projects are related to our research. Rodriguez et al. [19] proposed a system for pose detection and tracking of multiple insects and animals, which they used to monitor the traffic of honeybees and mice. The system uses a deep neural network to detect and associate detected body parts into whole insects or animals. The network predicts a set of 2D confidence maps of present body parts and a set of vectors of part affinity fields that encode associations detected among the parts. Greedy inference is subsequently used to select the most likely predictions for the parts and compiles the parts into larger insects or animals. The system uses trajectory tracking to distinguish entering and leaving bees, which works reliably under smaller bee traffic levels. The dataset used by the researchers to evaluate their system consists of 100 fully annotated frames, where each frame contains 6–14 honeybees. Since the dataset does not appear to be public, the experimental results cannot be independently replicated.

Babic et al. [20] proposed a system for pollen bearing honeybee detection in surveillance videos obtained at the entrance of a hive. The system's hardware includes a specially designed wooden box with a raspberry pi camera module inside. The box is mounted on the front side of a standard hive above the hive entrance. There is a glass plate placed on the bottom side of the box, 2 cm above the flight board, which forces the bees entering or leaving the hive to crawl a distance of ≈ 11 cm. Consequently, the bees in the field of view of the camera cannot fly. The training dataset contains 50 images of pollen bearing honeybees and 50 images of honeybees without pollen. The test dataset consists of 354 honeybee images. Since neither dataset appears to be public, the experimental results cannot be independently replicated.

The DPIV-based bee motion estimation algorithm presented in this article improves our previously proposed two-tier method of bee motion counting based on motion detection and motion region classification [6] in three respects: (1) it can be used to measure not only omnidirectional but also directional honeybee traffic; (2) it does not require extensive training of motion region classifiers (e.g., deep neural networks); and (3) it provides insect-independent motion measurement in that it does not require training insect-specific recognition models. Our evaluation results are based on four 30-s videos captured by deployed BeePi monitors. Each video consists of 744 frames. Each frame is manually labeled by a human observer for the number of full bee motions. Thus, the evaluation dataset for this investigation consists of 2976 manually labeled video frames. To ensure the replicability of our findings, we have made the videos and frame-by-frame bee motion counts publicly available (see the Supplementary Materials).

3. Materials and Methods

3.1. Hardware and Data Acquisition

The video data for this investigation were captured by BeePi monitors, multi-sensor EBM systems we designed and built in 2014 [4], and have been iteratively modifying [6,21] since then. Each BeePi monitor consists of a raspberry pi 3 model B v1.2 computer, a pi T-Cobbler, a breadboard, a waterproof DS18B20 temperature sensor, a pi v2 8-megapixel camera board, a v2.1 ChronoDot clock, and a Neewer 3.5 mm mini lapel microphone placed above the landing pad. All hardware components fit in a single Langstroth super. BeePi units are powered either from the grid or rechargeable batteries.

BeePi monitors thus far have had six field deployments. The first deployment was in Logan, UT (September 2014) when a single BeePi monitor was placed into an empty hive and ran on solar power for two weeks. The second deployment was in Garland, UT (December 2014–January 2015), when a BeePi monitor was placed in a hive with overwintering honeybees and successfully operated for nine out of the fourteen days of deployment on solar power to capture ≈ 200 MB of data. The third deployment was in North Logan, UT (April 2016–November 2016) where four BeePi monitors were placed into four beehives at two small apiaries and captured ≈ 20 GB of data. The fourth deployment was in Logan and North Logan, UT (April 2017–September 2017), when four BeePi units were placed into four beehives at two small apiaries to collect ≈ 220 GB of audio, video, and temperature data. The fifth deployment started in April 2018, when four BeePi monitors were placed into four beehives at an apiary in Logan, UT. In September 2018, we decided to keep the monitors deployed through the winter to stress test the equipment in the harsh weather conditions of northern Utah. By May 2019, we had collected over 400 GB of video, audio, and temperature data. The sixth field deployment started in May 2019 with four freshly installed bee packages and is ongoing with ≈ 150 GB of data collected so far.

3.2. Directional Bee Traffic

The main logical steps of the proposed algorithm are given in Figure 1. Let F_t and F_{t+1} be two consecutive image frames in a video such that F_t is taken at time t and F_{t+1} at time $t + 1$. Let IA_1 be a $D \times D$ window, referred to as *interrogation area* in the PIV literature, selected from F_t and centered at

position (i, j) . Another $D' \times D'$ window, IA_2 , is selected in F_{t+1} so that $D \leq D'$. The position of IA_2 in F_{t+1} is the function of the position of IA_1 in F_t in that it changes relative to IA_1 to find the maximum correlation peak. In other words, for each possible position of IA_1 in F_t , a corresponding position IA_2 is computed in F_{t+1} . For example, IA_2 in F_{t+1} can be centered at the same position (i, j) as IA_1 in F_t . The 2D matrix correlation is computed between IA_1 and IA_2 with the correlation formula in Equation (1).

$$C(r, s) = \sum_{i=0}^{D-1} \sum_{j=0}^{D-1} IA_1(i, j) IA_2(i + r, j + s), \quad (1)$$

where $r, s \in [-\lfloor \frac{D+D'-1}{2} \rfloor, \dots, \lfloor \frac{D+D'-1}{2} \rfloor]$

In Equation (1), $IA_1(i, j)$ and $IA_2(i + r, j + s)$ are the pixel intensities at location (i, j) in F_t and $(i + r, j + s)$ in F_{t+1} , respectively (see Figure 2). For each possible position (r, s) of IA_1 inside IA_2 , the correlation value $C(r, s)$ is computed. If the size of IA_1 is $M \times N$ and the size of IA_2 is $P \times Q$, then the size of the C matrix is $(M + P - 1) \times (N + Q - 1)$. The C matrix records the correlation coefficient for each possible alignment of IA_1 with IA_2 . Let $C(r_m, s_m)$ be the maximum value in C . If (i_c, j_c) is the center of IA_1 , then the positions of (i_c, j_c) and (r_m, s_m) define a displacement vector $\vec{v}_{i_c, j_c, r_m, s_m}$ from location (i_c, j_c) in F_t to $(i_c + r_m, j_c + s_m)$ in F_{t+1} . This vector is a representation of how the particles may have moved from F_t to F_{t+1} . All displacement vectors form a vector field that can be used to estimate possible flow patterns. Figure 2 shows the steps of how the cross correlation coefficients are computed. A faster way to calculate correlation coefficients between two image frames is to use the Fast Fourier Transform (FFT) and its inverse, as shown in Equation (2). The reason the equation tends to be computationally faster is that IA_1 and IA_2 must be of the same size.

$$C(r, s) = \Re[FFT^{-1}(FFT^*(IA_1) \cdot FFT(IA_2))] \quad (2)$$

Figure 1. Directional DPIV-based bee motion estimation algorithm.

For many real world images, some displacement vectors are invariably noisy and must be eliminated from the vector field. Some vectors can be eliminated as outliers if their signal to noise ratio exceeds a certain threshold. Additional vectors may be computed, if necessary, as the averages of their immediate neighbors as functions of the second and third largest peaks in the correlation plane to achieve subpixel accuracy.

Figure 2. 2D Correlation Algorithm. The upper image with particles on the left is F_t . The lower image on the left is F_{t+1} . The region with the light orange borders in the upper image is IA_1 . The region with the pink borders in the lower image is IA_2 . The 3D plot on the right plots all 2D correlation values. The highest peak helps to estimate the general flow direction.

Figure 3 gives the directional vectors computed by the proposed algorithm from two consecutive 640×480 video frames with an interrogation window size of 90 and an overlap of 60%. Figure 3a,b presents two consecutive frames from a 30-s video captured by a deployed BeePi monitor. In Figure 3a, a single moving bee is indicated by a green rectangle. In Figure 3b, the same bee, again indicated by a green rectangle, has moved toward the white landing pad (the white strip in the middle of the image) of the hive. This movement of the bee is reflected in a vector field of eleven vectors shown in Figure 3e. Figure 3c,d presents two consecutive frames from a different 30-s video captured by the same deployed BeePi monitor. In Figure 3c, two moving bees are detected: the first bee is denoted by the upper green rectangle and the second one is denoted by the lower green rectangle. In Figure 3d, the first (i.e., upper) bee has moved toward the landing pad, while the second (i.e., lower) bee has moved laterally, i.e., more or less parallel to the landing pad. The two bee moves are shown in two vector fields in Figure 3f. The upper vector field that captures the motion of the upper bee contains nine vectors, while the lower vector field that captures the motion of the lower bee contains only one vector.

Once the vector fields are computed for a pair of consecutive frames, the directions of these vectors are used to estimate directional bee traffic levels. Specifically, each vector is classified as *lateral*, *incoming*, or *outgoing* according to the value ranges in Figure 4. A vector $\vec{v}$ is classified as *outgoing* if its direction is in the range $[11^\circ, 170^\circ]$, as *incoming* if its direction is in the range $[-11^\circ, -170^\circ]$, and as *lateral* if its direction is in the ranges $[-10^\circ, 10^\circ]$, $[171^\circ, 180^\circ]$, or $[-171^\circ, -180^\circ]$.

Let F_t and F_{t+1} be two consecutive frames from a video V . Let $I_f(F_t, F_{t+1})$, $O_f(F_t, F_{t+1})$, and $L_f(F_t, F_{t+1})$ be the counts of incoming, outgoing, and lateral vectors. For each pair of consecutive images F_t and F_{t+1} , the algorithm computes three non-negative integers: $I_f(F_t, F_{t+1})$, $O_f(F_t, F_{t+1})$, and $L_f(F_t, F_{t+1})$. If a video V is a sequence of n frames $(F_1, F_2, \dots, F_n)$, then we can use I_f , O_f , and L_f to define the functions $I_v(V)$, $O_v(V)$, and $L_v(V)$ that return the counts of incoming, outgoing, and lateral vector counts for V , as shown in Equation (3).

$$\begin{aligned}
I_v(V) &= \sum_{i=1}^{n-1} I_f(F_i, F_{i+1}) \\
O_v(V) &= \sum_{i=1}^{n-1} O_f(F_i, F_{i+1}) \\
L_v(V) &= \sum_{i=1}^{n-1} L_f(F_i, F_{i+1})
\end{aligned} \tag{3}$$

For example, let $V = (F_1, F_2, F_3)$ such that $I_f(F_1, F_2) = 10$, $O_f(F_1, F_2) = 4$, $L_f(F_1, F_2) = 3$ and $I_f(F_2, F_3) = 2$, $O_f(F_2, F_3) = 7$, $L_f(F_2, F_3) = 5$. Then, $I_v(V) = I_f(F_1, F_2) + I_f(F_2, F_3) = 10 + 2 = 12$, $O_v(V) = O_f(F_1, F_2) + O_f(F_2, F_3) = 4 + 7 = 11$, and $L_v(V) = L_f(F_1, F_2) + L_f(F_2, F_3) = 3 + 5 = 8$. For each pair of consecutive frames F_i and F_{i+1} , the omnidirectional motion vector count is defined as the sum of the values of I_f , and O_f , and L_f , as shown in Equation (4). Then, for each video V , the omnidirectional vector count $T_v(V)$ for the video is the sum of the three directional counts, as also shown in Equation (4).

$$\begin{aligned}
T_f(F_t, F_{t+1}) &= I_f(F_t, F_{t+1}) + O_f(F_t, F_{t+1}) + L_f(F_t, F_{t+1}) \\
T_v(V) &= I_v(V) + O_v(V) + L_v(V)
\end{aligned} \tag{4}$$

Figure 3. (a,b) Two consecutive frames selected from a 30-s video. (c,d) Two consecutive frames selected from a different 30-s video. The vector field in (e) is generated from the images in (a,b); the vector field in (f) is generated from the images in (c,d).

Figure 4. Degree ranges used to classify DPIV vectors as lateral, incoming, and outgoing.

Table 1 gives the list of the DPIV parameters we used to run our experiments described in the next section. In all experiments, only a single pass of DPIV was executed.

Table 1. DPIV parameters used in the proposed bee motion estimation algorithm.

Parameter	Value
Img. Size	160×120 , 320×240 , 640×480
Inter. Win. Size	$n \times n$, where $n \in \{6, 8, 12, 16, 20, 32, 64, 90\}$
Inter. Win. Overlap (%)	10, 30, 50
Inter. Win. Correlation	FFT
Time Delta (DT = 1/FPS)	0.04 (25 frames per second)
Signal to Noise Ratio	peak1/peak2 with a threshold of 0.05
Spurious Vector Replacement	local mean with kernel size of 2 and max. iter. limit of 15

4. Experiments

4.1. Omnidirectional Bee Traffic

In our first experiment, we compared the performance of the proposed DPIV-based bee motion estimation algorithm with our previously proposed two-tier method of omnidirectional bee motion counting based on motion detection and motion region classification to estimate levels of omnidirectional bee traffic [6]. In that earlier investigation, we also made a preliminary, proof-of-concept evaluation of the two-tier method's performance on four randomly selected bee traffic videos. Specifically, we took approximately 3500 timestamped 30-s bee traffic videos captured by two deployed BeePi monitors in June and July 2018. From this collection, we took a random sample of 30 videos from the early morning (06:00–08:00), a random sample of 30 videos from the early afternoon (13:00–15:00), a random sample of 30 videos from the late afternoon (16:00–18:00), and a random sample of 30 videos from the evening (19:00–21:00). Each video consisted of 744 640×480 frames. Then, we randomly selected one video from the 30 early morning videos, one from the 30 early afternoon videos, one from the 30 late afternoon videos, and one from the 30 evening videos. We chose these time periods to ensure that we had videos with different traffic levels.

For each of the four videos, we manually counted full bee motions, frame by frame, in each video. The number of bee motions in the first frame of each video (Frame 1) was taken to be 0. In each subsequent frame, we counted the number of full bees that made any motion when compared to their positions in the previous frame. In addition to counting motions of bees present in each pair of consecutive frames F_t and F_{t+1} , we also counted as full bee motions when bees appeared in the subsequent frame F_{t+1} but were not present in the previous frame F_t (e.g., when a bee flew into the camera's field of view when F_{t+1} was captured).

We classified the four videos on the basis of our bee motion counts as no traffic (NT) (NT_Vid.mp4), low traffic (LT) (LT_Vid.mp4), medium traffic (MT) (MT_Vid.mp4), and high traffic (HT) (HT_Vid.mp4). It took us approximately 2 h to count bee motions in NT_Vid.mp4, 4 h in LT_Vid.mp4, 5.5 h in MT_Vid.mp4, and 6.5 h in HT_Vid.mp4, for a total of ≈ 18 h for all four videos. Table 2 is reproduced from our article [6], where we compared the performance of our four best configurations of the two-tier method for omnidirectional bee counting: MOG2/VGG16, MOG2/ResNet32, MOG2/ConvNetGS3, and MOG2/ConvNetGS4. The first element in each configuration (e.g., MOG2 [22]) specifies a motion detection algorithm; the second element (e.g., VGG16 [23]) specifies a classifier that classifies each motion region detected by the algorithm specified in the first element.

We compared the omnidirectional bee motion counts in Table 2 with the omnidirectional bee motion estimates returned by the DPIV-based bee motion estimation algorithm (i.e., T_v values in Equation (4)). Since the configuration MOG2/VGG16 gave us the overall best results in our previous investigation [6], we compared the proposed algorithm's estimates with the outputs of this configuration (see the VGG16 column in Table 2). The video frame size was 640×480 , the interrogation window size was 90, and the interrogation window overlap was 60%. Let C_1 , C_2 , and C_3 be the counts returned by the proposed DPIV-based algorithm, the MOG2/VGG16 algorithm, and the human counter, respectively. Of course, these counts cannot be directly compared, because the former are motion vector counts while the two latter are actual bee motion counts. Several motion vectors returned by the proposed algorithm may correspond to a single bee motion. To make a meaningful comparison, we standardized these three variables to be on the same scale by calculating the mean μ and the standard deviation σ for each variable, subtracting each observed value v from μ , and dividing by σ . Thus, the three standardized variables (i.e., C_1^* , C_2^* , and C_3^*) had an approximate μ of 0 and a σ of 1. Figure 5 gives the plots of the three standardized variables, C_1^* , C_2^* , and C_3^* , for each of the four videos.

Table 2. Performance of the system on the four evaluation videos: the columns VGG16, ResNet32, ConvNetGS3, and ConvNetGS4 give the bee motion counts returned by the configurations MOG2/VGG16, MOG2/ResNet32, MOG2/ConvNetGS3, and MOG2/ConvNetGS4, respectively; and the last column gives the human bee motion counts for each video.

Video	Num. Frames	VGG16	ResNet32	ConvNetGS3	ConvNetGS4	Human Count
NT_Vid.mp4	742	151	75	182	127	73
LT_Vid.mp4	744	47	25	57	43	353
MT_Vid.mp4	743	1245	145	597	316	2924
HT_Vid.mp4	744	16,647	13,362	16,569	15,109	5738

The plots in Figure 5 of the three standardized variables C_1^* , C_2^* , and C_3^* can be construed as time series and compared as such. A classical approach for calculating similarity between two time series is by using Dynamic Time Warping (DTW) (e.g., [24,25]). DTW is a numerical similarity measure of how two time series can be optimally aligned (i.e., *warped*) in such a way that the accumulated alignment cost is minimized. DTW is based on dynamic programming in that the method finds all possible alignment paths and selects a path with a minimum cost.

$$D(i, j) = d_{i,j} + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} \quad (5)$$

Figure 5. Plots of the three standardized variables on four evaluation videos, no traffic (NT_Vid.mp4), low traffic (LT_Vid.mp4), medium traffic (MT_Vid.mp4), and high traffic (HT_Vid.mp4). The total number of frames examined were 216,956 video frames. The x-axis is the time (in seconds) when each video frame is captured. The y-axis is the values of the standardized variables C_1^* (DPIV), C_2^* (MOG2/VGG16), and C_3^* (Human Count). C_3^* is the ground truth. Let F_t and F_{t+1} be two consecutive frames captured at times t and $t+1$, $1 \leq t \leq 29$. $C_1^*(t+1)$ is the standardized value of $T_f(F_t, F_{t+1})$, $C_2^*(t)$ is the standardized value of the count returned by the MOG2/VGG16 algorithm for F_t , and C_3^* is the standardized value of the count returned by the human counter for F_t . $C_1^*(1) = C_2^*(1) = C_3^*(1) = 0$.

Let X and Y be two time series such that $X = (x_1, x_2, \dots, x_i, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_j, \dots, y_m)$, for some integers n and m . An $n \times m$ distance matrix D is created where each cell $D(i, j)$ is the cumulative cost of aligning $(x_1, \dots, x_i)$ with $(y_1, \dots, y_j)$ defined in Equation (5), where $d(i, j) = d(x_i, y_j) = (x_i - y_j)^2$ is the local distance (e.g., Euclidean) between the points x_i and y_j from X and Y , respectively. An optimal path is a path with a minimal cumulative cost of aligning the time series from $D(1, 1)$ to $D(n, m)$. $DTW(X, Y)$ is the sum of the pointwise distances along the optimal path.

Table 3 gives the DTW comparisons of C_1^* (DPIV) and C_2^* (MOG2/VGG16) with the ground truth of C_3^* (Human Count) for each of the four videos. The DTW coefficients between C_1^* and C_2^* are lower than the DTW coefficients between C_2^* and C_3^* on all videos except for the no traffic video NT_Vid.mp4. A closer look at the no traffic video revealed that when the video was taken a neighboring beehive was being inspected and one of the inspectors slightly pushed the monitored beehive, which caused the camera to shake. This push resulted in several frames in the video where the DPIV-based bee motion estimation algorithm detected motions. Overall, in terms of omnidirectional bee motion counts, the proposed algorithm approximates the human ground truth on par with the MOG2/VGG16 algorithm. Tables A1–A3 in Appendix A give the DTW similarity scores of the omnidirectional traffic curves of DPIV with MOG2/ConvGS3, MOG2/ConvGS4, MOG2/ResNet32, and the human ground truth that show similar results. These results indicate that the proposed algorithm estimates omnidirectional traffic levels on par with our previously proposed two-tier algorithm to count omnidirectional bee motions in videos [6].

Table 3. Omnidirectional traffic curve comparison: C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

DTW vs. Video	HT_Vid.mp4	MT_Vid.mp4	LT_Vid.mp4	NT_Vid.mp4
$DTW(C_1^*, C_3^*)$	20.75	21.71	15.36	10.53
$DTW(C_2^*, C_3^*)$	22.02	22.19	18.13	6.47

4.2. Subpixel Accuracy

We investigated the appropriateness of the traditional 3-point Gaussian sub-pixel accuracy in the context of estimating bee motions to determine whether the correlation matrices generated Gaussian curves. We randomly selected several consecutive image pairs from our dataset. For each selected image pair, the image size was 160×120 and the interrogation window size was 12 with a 50% overlap. To simplify visualization, we selected only the image pairs with a singly bee moving. Figure 6 shows a sample image pair with the moving bee selected by the green box and the corresponding vectors generated by DPIV.

We computed the correlation matrices for all motion vectors and visualized them with 3D plots, as shown in Figure 7. Each 3D plot in Figure 7 corresponds to a vector in the vector map of Figure 6. Specifically, the first 3D plot in the first row (Figure 7a) corresponds to the first (left) vector in the first row of the vector map in Figure 6c; the second 3D plot in the first row (Figure 7b) corresponds to the second (right) vector in the first row of the vector map in Figure 6c; the first 3D plot in the second row (Figure 7c) corresponds to the first vector in the second row of Figure 6c; the second 3D plot in the second row (Figure 7d) corresponds to the second vector in the second row of the vector map in Figure 6c; the first 3D plot in the third row (Figure 7e) corresponds to the first vector in the third row of the vector map in Figure 6c; and the second 3D plot in the third row (Figure 7f) corresponds to the second vector in the third row of the vector map in Figure 6c. Since the vectors in the first row are small, we scaled the plots to better visualize the corresponding distribution (see Figure 8). Another sample image pair with the corresponding DPIV vectors and plots are given in Figures A1 and A2 in Appendix A.

By analyzing such 3D plots of correlation vectors of two consecutive images with single bees moving, we concluded that all correlation maps follow Gaussian distribution. We also observed that not all plots have distinct peaks. For example, the first plot in the second row in Figure 7 has a distinct peak while the first plot of the third row has no distinct peaks. Therefore, the traditional three-point Gaussian sub-pixel accuracy method can be used to determine better peaks in such cases. Another observation we made is that the middle vectors tend to have larger magnitudes than the vectors in the higher and lower rows. This observation seems to indicate that the body of the bee, as a whole, causes maximum motion and, as a consequence, it is reasonable to treat whole bees as tracers. The bee body parts such as wings, thorax, and legs result in smaller motions or are not detected at all.

Figure 6. Two consecutive video frames, Frame 1 and Frame 2 with a single moving bee and corresponding DPIV motion vectors.

(a) Correlation map for Vector 1

(b) Correlation map for Vector 2

(c) Correlation map for Vector 3

(d) Correlation map for Vector 4

(e) Correlation map for Vector 5

(f) Correlation map for Vector 6

Figure 7. 3D plots of correlation matrices for the motion vectors in Figure 6c. Each row in the figure corresponds to each row in the vector plot of Figure 6c.

Figure 8. Scaled 3D map for the correlation matrices of Figure 7a,b respectively.

4.3. Interrogation Window Sizes

We used a single bee as a tracer. However, a single bee motion generates multiple motion vectors, because multiple bee parts (wings, thorax, legs, etc.) are moving simultaneously, which may be captured in interrogation windows. We executed additional experiments with our algorithm to evaluate its performance with different image and interrogation window sizes. Specifically, we experimented with three image sizes: 160×120 , 320×240 , and 640×480 . For each image size, we manually calculated the bee size in pixels. For 160×120 images, the average bee size was equal to 8×8 pixels; for 320×240 images, the average bee size was equal to 16×16 ; and for images 640×480 , the average bee size was equal to 32×32 . For each image size and bee size, we tested different sizes of interrogation windows (i.e., smaller than the bee size, equal to the bee size, and larger than the bee size) with different amounts of interrogation window overlap. We made no attempt to improve the quality of the data or reduce the uncertainty of particle displacement, because the system should work out-of-the-box in the field on inexpensive, off-the-shelf hardware for beekeepers and apiary scientists worldwide to afford it.

To remove the spurious vectors, we calculated the signal to noise (SNR) ratio for each generated correlation matrix. The SNR was calculated by finding the highest peak (p_1) and the second highest peak (p_2) in the correlation matrix and then computing SNR as their ratio (i.e., $SNR = p_1/p_2$). If the SNR values were less than a given threshold (0.05 in our case), the corresponding vectors were considered spurious and removed. Such vectors were replaced by the weighted average of the neighboring vectors. We used a kernel size (K) of 2 and a local mean to assign the weights with the formula $1/((2K+1)^2 - 1)$. We used a maximum iteration limit of 15 and a tolerance of 0.001 to replace the spurious vectors.

Table 4 gives the results of our experiments with different interrogation window sizes on 640×480 image frames in the high-traffic video *HT_Vid.mp4*. The last two columns give the dynamic time warping (DTW) results. Tables for the other image sizes and videos (Tables A5–A15) are given in the Appendix A. As these tables show, the best similarity with the ground truth was achieved when the interrogation window was twice the size of the bee or higher for all three image sizes and an interrogation window overlap of 30% or higher.

Table 4. DTW values for *HT_Vid.mp4* with image size of 640×480 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
640×480	32×32	28	10	13	19.1884	22.02607
		28	30	8	18.4507	22.02607
		28	50	14	18.6779	22.02607
		32	10	3	19.3991	22.02607
		32	30	10	19.8832	22.02607
		32	50	16	17.7650	22.02607
		36	10	4	19.4907	22.02607
		36	30	11	17.4369	22.02607
		36	50	18	20.0698	22.02607
		64	10	6	22.0901	22.02607
		64	30	19	21.1846	22.02607
		64	50	32	20.1906	22.02607
		90	10	9	23.4524	22.02607
		90	30	27	20.8363	22.02607
		90	50	45	21.3194	22.02607

In the course of our investigation, we discovered that in some videos there were bees which flew very fast. Figure 9 gives the first and last frames (Frames 1 and 7, respectively) in a seven-frame sequence from a high bee traffic video. Table A3 in the Appendix A gives all the frames in the sequence. In Figure 9, a fast moving bee is delineated in each frame by a green polygon. The bee is moving right to left. The coordinates of the green dot (a single pixel) in each polygon are $x = 507.0$, $y = 1782.0$ in Frame 1 and $x = 660.0$, $y = 151.5$ in Frame 7. The pixel displacements of the green dot for this sequence are as follows: 220.9 pixels between Frames 1 and 2 (first displacement); 218.5 pixels between Frames 2 and 3 (second displacement); 247.6 pixels between Frames 3 and 4 (third displacement); 329.7 pixels between Frames 4 and 5 (fourth displacement); 347.6 pixels between Frames 5 and 6 (fifth displacement); and 303.1 pixels between Frames 6 and 7 (sixth displacement). The total pixel displacement of the green dot between Frames 1 and 7 is ≈ 1667.4 pixels. The average pixel displacement per frame is ≈ 277.9 pixels. The pi camera used in our investigation is not sufficiently powerful to capture all such displacements. In our system, the maximum pixel displacement per frame is set to 40 pixels, which represents the maximum velocity that our system can currently handle. As the pi hardware improves, we hope to raise this parameter to 60 or 80 pixels.

Figure 9. The first and last frames (Frames 1 and 7, respectively) from a sequence of seven consecutive frames taken from a high bee traffic video; the entire seven-frame sequence is given in Table A3.

4.4. Directional Bee Traffic

The two-tier method of bee motion counting proposed in our previous investigation [6] is omnidirectional in that it counts bee motions regardless of direction. Therefore, this method cannot be used to estimate incoming, outgoing, or lateral bee traffic. The algorithm proposed in this article, on the other hand, can be used to estimate these types of bee traffic.

In another series of experiments, we sought to investigate whether the proposed algorithm can be used to estimate how closely incoming traffic matches outgoing traffic in healthy beehives. In principle, in healthy beehives, levels of incoming bee traffic should closely match levels of outgoing bee traffic, because all foragers that leave the hive should eventually come back. Of course, if the levels of two traffic types are treated as time series, local mismatches are inevitable, because some foragers that leave the hive at the same time may not return at the same time because they fly to different foraging sources. However, the overall matching tendency should be apparent. Departures from this tendency can be treated as deviations from normalcy and may lead to manual inspection. For example, a large number of foragers leaving the hive at a given time and never coming back may signal that a swarm event has taken place. On the other hand, if the level of incoming traffic greatly exceeds the corresponding level of outgoing traffic over a given time segment, a beehive robbing event may be underway.

To estimate how closely the levels of outgoing and incoming bee traffic match, we took 292 30-s videos (i.e., all the videos captured by a BeePi monitor deployed on a healthy beehive from 1–5 July 2018 and 8–14 July 2018 (≈ 24 videos per day)) and computed the values of I_v and O_v (see Equation (3)) for each video with the proposed algorithm. The video frame size was 640×480 , the interrogation window size was 90, and the interrogation window overlap was 60%. Figure 10 shows the plots of the values of I_v and O_v for all videos on four days from the specified period. As can be seen from these plots, on many days in a healthy beehive the estimated levels of incoming and outgoing traffic are likely to be very close. Spikes in outgoing curves tend to follow spikes in incoming curves and vice versa.

We computed the DTW similarity scores between each outgoing bee traffic curve and each incoming bee traffic curve in Figure 10. Table 5 gives these DTW scores. These scores indicate that the outgoing and incoming curves tend to match closely on the same day and tend to be dissimilar on different days. In particular, the DTW scores between the incoming and outgoing bee traffic curves on the same days, shown in the diagonal, are all below 5. The values are almost twice as high when the incoming and outgoing curves are from different days. Table A4 in Appendix A gives the DTW similarity scores for all twelve days in the sampled period. The blue diagonal values in the table show the DTW similarity scores between the incoming and outgoing traffic curves on the same day. The statistics for the diagonal values are: median = 4.75, $\mu = 6.1$, and $\sigma = 2.85$. The statistics for the other 132 non-diagonal values (i.e., DTW similarities between traffic curves on different days) are: median = 13.145, $\mu = 13.28$, $\sigma = 2.99$. These statistics suggest possible ranges for monitoring DTW similarity scores between incoming and outgoing traffic curves for healthy beehives.

On two days in the sample, 3 July 2018 and 5 July 2018, as can be seen in Table A4, the DTW similarity scores between the outgoing and incoming curves were above 10 on the same day. Figure 11a,b shows traffic curves for these two days. Our manual analysis of the videos on these two days revealed that the incoming bees were flying into the beehive too fast for DPIV to detect motion in the videos captured by a raspberry pi camera, which resulted in the mismatch between the incoming and outgoing traffic estimates.

(a) 4 July 2018

(b) 8 July 2018

(c) 9 July 2018

(d) 11 July 2018

Figure 10. Plots of I_v and O_v values computed by the proposed algorithm for 4, 8, 9, and 11 July 2018.**Table 5.** DTW similarity scores between outgoing and incoming traffic curves in Figure 10.

	Outgoing	2018-07-04	2018-07-08	2018-07-09	2018-07-11
Incoming					
4 July 2018		4.78	9.58	10.43	11.49
8 July 2018		12.36	4.32	11.42	10.39
9 July 2018		10.90	11.94	4.69	9.59
11 July 2018		14.08	10.18	9.37	4.43

(a) 3 July 2018

(b) 5 July 2018

Figure 11. Plots of I_v and O_v values for 3 July 2018 and 5 July 2018. DTW similarity of the curves on 3 July is 12.436; DTW similarity of the curves on 5 July is 10.179.

4.5. DPIV on Raspberry Pi

DPIV is computationally expensive on the raspberry pi hardware even when cross correlation is implemented with FFT (see Equation (2)). In implementing our algorithm, we used the OpenPIV library (<http://www.openpiv.net>), a community driven initiative to develop free and open-source particle image velocimetry software. Since in our BeePi monitors all data capture and computation is done on raspberry pi computers, it is imperative that DPIV run on the raspberry pi hardware.

We tested our algorithm on a single raspberry pi 3 model B v1.2 computer on several 30-s videos from our dataset. The video frame size was 640×480 , the interrogation window size was 90, and the interrogation window overlap was 60%. On average, it took ≈ 2.5 h for the algorithm to process a single 30-s video. This result is unacceptable to us, because each BeePi monitor records a 30-s video every 15 min from 8:00 to 21:00 every day. An obvious solution is cloud computing where the captured videos are processed in the cloud by a collection of GPUs. We rejected this solution, because from the beginning of the BeePi project in 2014 we made it a priority to avoid external dependencies such as continuous Internet and cloud computing. In remote areas, BeePi monitors should function as self-configurable and self-contained networks that rely exclusively on local area networking mechanisms (e.g., as an ad hoc network).

Since the raspberry pi 3 model B v1.2 computer has four CPUs, we decided to introduce multi-threading into our algorithm to reduce video processing time. A BeePi monitor records videos at ≈ 25 frames per second. Thus, a 30-s video generates 743–745 frames. Each thread ran on a separate CPU to process a subset of frames. Specifically, Thread 1 processed Frames 1–185, Thread 2 processed Frames 185–370, Thread 3 processed Frames 370–555, and Thread 4 processed Frames 555–743/745. It should be noted that, to handle boundary frames, there is a one frame overlap between the ranges because DPIV works on pairs of consecutive frames. Multi-threading reduced the average video processing time from 2.5 h to 1 h 12 min.

After multi-threading, we introduced parallelism to our system by distributing the algorithm's computation over six raspberry pi computers. We chose to experiment with six raspberry pi computers, because they easily fit into a single deep Langstroth super (i.e., a wooden box) placed on top of a beehive [26,27]. The current BeePi hardware also fits into a single deep super. In this six-node network, one node acts as a dispatcher in that it captures video via a raspberry pi camera and dispatches video frames to the other five nodes that process them and return the results back to the dispatcher. In this architecture, the dispatcher sends Frames 1–148 to Node 1, Frames 148–296 to Node 2, Frames 296–444 to Node 3, Frames 444–592 to Node 4, and Frames 592–743/745 to Node 5. Each processing node uses multi-threading, described above, to compute directional vectors, classifies them as incoming, outgoing, and lateral, and returns these vector counts to the dispatcher. We timed each task in the system on ten 30-s videos. Table 6 details our measurements. The sum total of the third column is equal to 1169.462 s ≈ 19.49 min, which is the total time our six-node network takes to process a 30-s video.

Table 6. Average video processing performance of six-node ad hoc network.

Task	Executive	Time (in sec)
Frame Generation	Dispatcher Node	70.826
Frame Dispatch	Dispatcher Node	211.116
Frame Processing	5 Processing Nodes	887.52

Through multi-threading and parallelism, we managed to reduce the processing time of a single video from 2.5 h to below 20 min, which makes it feasible to process all videos captured in one 24-h period. In the current version of the BeePi monitor, video recording starts at 8:00 and ends at 21:00 (i.e., 13 h or 780 min) and records a 30-s video every 15 min. If there are no hardware disruptions, there are 52 videos recorded every 24 h. The period from 21:00 of the current day to 8:00 of the next day (i.e., 11 h or 660 min) is the video recording downtime during which the BeePi monitor does no video recording and records only temperature and audio every 15 min, neither of which is computationally taxing. During the video recording period, the monitor spends 2 min per hour (four 30-s videos) on recording a video for a total of 26 min per day (13-h video capturing period with 2 min of actual video recording per hour), which makes a total of 754 min ($780 - 26$) available for video processing during the video recording period. During the video recording downtime (660 min), the monitor can also process videos. Therefore, every 24 h period, 1414 ($754 + 660$) min are available for video processing. However, to process all 52 videos captured every 24 h, the six-node network requires only 1040 min. Thus, the six-node ad hoc network can process all the videos recorded in the current 24-h period before the next 24-h data capture period begins.

5. Conclusions

Our experiments indicate that the proposed DPIV-based bee motion estimation algorithm performs on par with the two-tier bee motion counting algorithm proposed in our previous research [6] in terms of measuring omnidirectional honeybee traffic. However, unlike the two-tier algorithm that can estimate only omnidirectional honeybee traffic, the proposed algorithm can measure both omnidirectional and directional honeybee traffic. The proposed algorithm shows that DPIV can be used not only for quantifying particle velocities but also for analyzing instantaneous and longitudinal flow structures.

Another advantage of the current algorithm is that it does not rely on image classification and, consequently, can be used to measure traffic of other insects such as bumble bees and ants. Inasmuch as DPIV is a purely analytical technique, it requires no machine learning and no large curated datasets. By contrast, our two-tier algorithm uses machine learning methods (deep neural networks and random forests) that took us 1292.77 GPU h (53.87 days) to train, test, and validate on our curated datasets of 167,261 honeybee images [6,28–30]. The network of six raspberry pi computers can process all captured videos within every 24-h data capturing period. The fact that the proposed DPIV-based bee motion estimation algorithm provides us directionality and insect-independent motion measurement is an acceptable price to pay for its relative slowness. As the raspberry pi hardware improves, we expect this processing time to decrease.

A fundamental limitation of DPIV is its dynamic velocity range (DVR) [31], which is the ratio of the maximum measurable velocity and the minimum measurable velocity. The maximum measurable velocity is the maximum allowable particle displacement within each interrogation area on the paired images (see Figure 2). The minimum measurable velocity is the smallest measurable particle displacement between pixels in the paired images. To overcome the DPIV limitation, traditional DPIV research has required significant capital and maintenance investments, which confined DPIV applications mostly to academic or industrial research environments [32]. By distributing the algorithm's computation over six raspberry pi computers, we showed that useful DPIV-based results can be obtained on a hardware base that costs less than 1000 USD. This result bodes well for citizen science in that the hardware base can be replicated without significant capital and maintenance costs. We are also currently investigating several techniques of combining multiple neighboring vectors into a single bee. While our preliminary experiments indicate that such techniques may work on low-traffic and medium-traffic videos, they may have issues with high-traffic videos.

We have learned from this investigation that obtaining ground truth videos to estimate the accuracy of bee motion counts is very labor intensive and time consuming. It took us ≈ 18 h to obtain human bee motion counts on four videos of live honeybee traffic. We are currently working on curating more live honeybee traffic videos and hope to make them public in our future publications.

One curious observation we have made about lateral traffic is that the orientation of the bees engaged in it typically remains perpendicular to the hive (i.e., the bees' heads face the hive). Another curious observation we have made about lateral traffic is that sometimes a bee engages in lateral traffic before entering the hive and sometimes, after doing several sidewise maneuvers, she leaves the vicinity of the hive captured by the camera. The laterally moving bees that eventually enter the hive may use lateral moves to better home in on the entry point, whereas the laterally moving bees that fly away might be stray bees looking for a new home or scouts from other hives checking if the hive is appropriate for robbing. As computer scientists, however, we have no explanation of these observations and leave them for entomologists to investigate.

We made no attempt in this investigation to estimate the uncertainty of a DPIV displacement field. In our future work, we plan to consider applying the methods proposed by Wieneke [33] and evaluated by Sciacchitano et al. [34] to estimate the uncertainty of generated displacement fields. However, the application of such methods will have to undergo a detailed cost-benefit analysis in view of our ultimate objective to obtain reliable and consistent bee traffic measurements from digital videos on inexpensive hardware that can be replicated by third parties. We are not concerned with flight trajectories or flow patterns as such. While the latter may turn out to be a significant research by-product and a contribution, they are not the primary focus of video-based electronic beehive monitoring. We plan to focus our future work on effective, computationally feasible methods to match the numbers of bees leaving the hive with the numbers of bees entering the hive and using significant mismatches as indicators that the hive is not healthy or is under stress. Another possible improvement that we plan to introduce into our system is the application of fast background elimination techniques to the video frames prior to DPIV. More uniform backgrounds will enable us to experiment with larger interrogation windows and multi-pass DPIV to reduce noise and uncertainty.

Our long-term objective has been, and will continue to be, the development of an open design and open data electronic beehive monitoring (EBM) platform for researchers, practitioners, and citizen scientists worldwide who want to monitor their beehives or use their beehives as environmental monitoring stations [35]. Our approach is based on the following four principles. First, the EBM hardware should be not only open design but also 100% replicable, which we ensure by using exclusively readily available off-the-shelf hardware components [26,27]. Second, the EBM software should also be 100% replicable, which we achieve by relying on open source resources. Third, the project's hardware should be compatible with standard beehive models used by many beekeepers worldwide (e.g., the Langstroth beehive [7] or the Dadant beehive [36]) without any required structural modifications of beehive woodenware. Fourth, the sacredness of the bee space should be preserved in that the deployment of EBM sensors should not be disruptive to natural beehive cycles.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2076-3417/10/6/2042/s1>, The supplementary materials include four CSV files with frame-by-frame full bee motion counts and the corresponding four honeybee traffic videos. Due to space limitations, we could not include the 292 honeybee traffic videos we used in our directional honeybee traffic experiments. Interested parties should contact the second author directly if they want to obtain the videos.

Author Contributions: Supervision, Project Administration, and Resources, V.K.; Conceptualization and Software, S.M. and V.K.; Data Curation, V.K. and S.M.; Writing—Original Draft Preparation, V.K. and S.M. Writing—Review and Editing, V.K. and S.M.; Investigation and Analysis, S.M. and V.K.; and Validation, S.M. and V.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded, in part, by our Kickstarter fundraisers in 2017 [26] and 2019 [27].

Acknowledgments: We would like to thank all our Kickstarter backers and especially our BeePi Angel Backers (in alphabetical order): Prakhar Amlathe, Ashwani Chahal, Trevor Landeen, Felipe Queiroz, Dinis Quelhas, Tharun Tej Tammineni, and Tanwir Zaman. We express our gratitude to Gregg Lind, who backed both fundraisers and donated hardware to the BeePi project. We are grateful to Richard Waggstaff, Craig Huntzinger, and Richard

Mueller for letting us use their property in northern Utah for longitudinal EBM tests. We would like to thank Jacquelyn Mukherjee for helping us with beehive inspections and beekeeper log entries and for proofreading this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EBM	Electronic Beehive Monitoring
FFT	Fast Fourier Transform
DL	Deep Learning
ConvNet	Convolutional Neural Network
PIV	Particle Image Velocimetry
DPIV	Digital Particle Image Velocimetry
DVR	Dynamic Velocity Range
DTW	Dynamic Time Warping
NT	No Traffic
LT	Low Traffic
MT	Medium Traffic
HT	High Traffic

Appendix A

Figure A1. Two consecutive video frames A1 and A2, with two bees moving and corresponding DPIV motion vectors.

Table A1. Omnidirectional Traffic Curve Comparison: C_1^* is DPIV; C_2^* is MOG2/ConvGS3; C_3^* is Human Count (ground truth); the video frame size was 640×480 , the interrogation window size was 90; and the interrogation window overlap was 60%.

DTW vs. Video	HT_Vid.mp4	MT_Vid.mp4	LT_Vid.mp4	NT_Vid.mp4
$DTW(C_1^*, C_3^*)$	20.75	21.71	15.36	10.53
$DTW(C_2^*, C_3^*)$	18.23	18.78	19.26	6.47

(a) Correlation Map 1

(b) Correlation Map 2

(c) Correlation Map 3

(d) Correlation Map 4

(e) Correlation Map 5

(f) Correlation Map 6

Figure A2. 3D plots of correlation matrices of the vectors in Figure A1c. Each row in the figure corresponds to each row in the vector plot of Figure A1c.

Table A2. Omnidirectional Traffic Curve Comparison: C_1^* is DPIV; C_2^* is MOG2/ConvGS4; C_3^* is Human Count (ground truth); the video frame size was 640×480 , the interrogation window size was 90; and the interrogation window overlap was 60%

DTW vs. Video	HT_Vid.mp4	MT_Vid.mp4	LT_Vid.mp4	NT_Vid.mp4
$DTW(C_1^*, C_3^*)$	20.75	21.71	15.36	10.53
$DTW(C_2^*, C_3^*)$	19.29	18.46	22.28	6.47

Table A3. Omnidirectional Traffic Curve Comparison: C_1^* is DPIV; C_2^* is MOG2/ResNet32; C_3^* is Human Count (ground truth); the video frame size was 640×480 , the interrogation window size was 90; and the interrogation window overlap was 60%.

DTW vs. Video	HT_Vid.mp4	MT_Vid.mp4	LT_Vid.mp4	NT_Vid.mp4
$DTW(C_1^*, C_3^*)$	20.75	21.71	15.36	10.53
$DTW(C_2^*, C_3^*)$	21.76	20.17	19.58	6.47

Table A4. DTW similarity scores between outgoing and incoming traffic curves for 1–5 July 2018 and 8–14 July 2018; the diagonal values are marked in blue to signify that that particular DTW value is the lowest in the investigating row. The statistics of the blue diagonal values (same day) are: median = 4.75, $\mu = 6.1$, and $\sigma = 2.85$; and the statistics for the other 132 non-diagonal values (different days) are: median = 13.145, $\mu = 13.28$, $\sigma = 2.99$.

Incoming \ Outgoing	01	02	03	04	05	08	09	10	11	12	13	14
1 July 2018	5.96	10.44	14.31	12.70	16.06	11.13	11.48	10.99	12.75	14.66	8.15	16.44
2 July 2018	8.86	7.25	15.65	11.61	16.26	12.17	11.08	9.67	8.99	13.42	8.56	14.97
3 July 2018	11.79	13.15	12.43	15.43	14.85	17.29	14.74	13.30	15.03	12.67	12.78	17.69
4 July 2018	13.46	15.66	18.14	4.78	15.70	9.58	10.43	12.80	11.49	13.64	11.19	17.25
5 July 2018	11.48	11.11	15.26	12.90	10.17	16.42	13.72	13.89	14.01	15.33	13.14	18.12
8 July 2018	11.54	13.70	16.62	12.36	18.39	4.32	11.42	8.06	10.39	11.81	9.67	13.94
9 July 2018	12.01	12.40	15.04	10.90	17.51	11.94	4.69	11.86	9.59	9.49	8.94	12.04
10 July 2018	12.36	13.57	15.87	11.70	17.72	9.21	10.80	3.59	10.16	15.18	8.76	15.67
11 July 2018	11.98	13.67	14.57	14.08	17.19	10.18	9.37	9.20	4.43	11.31	7.83	13.16
12 July 2018	16.46	14.19	17.60	16.08	20.79	15.62	12.11	14.27	10.63	4.72	13.78	11.75
13 July 2018	8.63	12.17	14.39	13.60	17.03	11.52	9.59	7.58	7.65	12.44	2.27	16.39
14 July 2018	19.27	17.52	22.13	16.87	20.80	17.02	12.72	15.84	14.14	11.32	14.45	8.75

Table A5. DTW values for *MT_Vid.mp4* with image size of 640×480 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pixel Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
640×480	32×32	28	10	13	17.0894	22.1954
		28	30	8	21.9474	22.1954
		28	50	14	21.9149	22.1954
		32	10	3	18.7923	22.1954
		32	30	10	21.6549	22.1954
		32	50	16	22.8640	22.1954
		36	10	4	23.0033	22.1954
		36	30	11	22.2269	22.1954
		36	50	18	23.3584	22.1954
		64	10	6	23.7384	22.1954
		64	30	19	16.8875	22.1954
		64	50	32	20.4414	22.1954
		90	10	9	24.4479	22.1954
		90	30	27	23.9692	22.1954
		90	50	45	23.0591	22.1954

Table A6. DTW values for *LT_Vid.mp4* with image size of 640×480 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
640×480	32×32	28	10	13	22.5870	18.1374
		28	30	8	16.6490	18.1374
		28	50	14	19.7898	18.1374
		32	10	3	19.9289	18.1374
		32	30	10	22.1706	18.1374
		32	50	16	21.5204	18.1374
		36	10	4	19.9186	18.1374
		36	30	11	20.2431	18.1374
		36	50	18	19.0067	18.1374
		64	10	6	18.7126	18.1374
		64	30	19	17.3478	18.1374
		64	50	32	17.7892	18.1374
		90	10	9	3.4591	18.1374
		90	30	27	14.7361	18.1374
		90	50	45	14.7819	18.1374

Table A7. DTW values for *NT_Vid.mp4* with image size of 640×480 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
640×480	32×32	28	10	13	20.7712	6.4764
		28	30	8	23.3059	6.4764
		28	50	14	21.2160	6.4764
		32	10	3	19.2069	6.4764
		32	30	10	19.3079	6.4764
		32	50	16	18.7725	6.4764
		36	10	4	18.1162	6.4764
		36	30	11	18.9071	6.4764
		36	50	18	18.6362	6.4764
		64	10	6	17.9736	6.4764
		64	30	19	15.1433	6.4764
		64	50	32	5.4638	6.4764
		90	10	9	12.0973	6.4764
		90	30	27	11.9666	6.4764
		90	50	45	12.0287	6.4764

Table A8. DTW values for *HT_Vid.mp4* with image size of 320×240 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
320×240	16×16	12	10	1	18.8078	22.0260
		12	30	4	18.5066	22.0260
		12	50	6	17.3962	22.0260
		16	10	2	18.1778	22.0260
		16	30	5	18.9622	22.0260
		16	50	8	17.9464	22.0260
		20	10	2	22.2556	22.0260
		20	30	6	22.8514	22.0260
		20	50	10	19.3598	22.0260
		32	10	3	19.8358	22.0260
		32	30	10	21.5200	22.0260
		32	50	16	21.9050	22.0260

Table A9. DTW values for *MT_Vid.mp4* with image size of 320×240 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
320×240	16×16	12	10	1	17.9630	22.1954
		12	30	4	22.4978	22.1954
		12	50	6	23.8835	22.1954
		16	10	2	17.9911	22.1954
		16	30	5	18.7131	22.1954
		16	50	8	20.2220	22.1954
		20	10	2	24.0475	22.1954
		20	30	6	21.0510	22.1954
		20	50	10	22.1260	22.1954
		32	10	3	24.4598	22.1954
		32	30	10	20.8316	22.1954
		32	50	16	21.1558	22.1954

Table A10. DTW values for *LT_Vid.mp4* with image size of 320×240 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
320×240	16×16	12	10	1	23.7113	18.1374
		12	30	4	23.3807	18.1374
		12	50	6	24.4808	18.1374
		16	10	2	24.3014	18.1374
		16	30	5	23.4989	18.1374
		16	50	8	25.0730	18.1374
		20	10	2	21.0612	18.1374
		20	30	6	22.6352	18.1374
		20	50	10	23.5116	18.1374
		32	10	3	22.4214	18.1374
		32	30	10	17.4053	18.1374
		32	50	16	17.7859	18.1374

Table A11. DTW values for *NT_Vid.mp4* with image size of 320×240 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
320×240	16×16	12	10	1	20.9882	6.4764
		12	30	4	20.2531	6.4764
		12	50	6	19.9547	6.4764
		16	10	2	14.5662	6.4764
		16	30	5	14.8737	6.4764
		16	50	8	13.5293	6.4764
		20	10	2	11.2902	6.4764
		20	30	6	11.9446	6.4764
		20	50	10	11.6527	6.4764
		32	10	3	11.9800	6.4764
		32	30	10	12.0924	6.4764
		32	50	16	10.5347	6.4764

Table A12. DTW values for *HT_Vid.mp4* with image size of 160×120 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
160×120	8×8	6	30	2	17.0827	22.0260
		6	50	3	16.4780	22.0260
		8	30	2	18.4156	22.0260
		8	50	4	18.8498	22.0260
		12	30	4	18.8323	22.0260
		12	50	6	17.1548	22.0260
		16	30	5	21.3475	22.0260
		16	50	8	17.6036	22.0260

Table A13. DTW values for *MT_Vid.mp4* with image size of 160×120 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
160×120	8×8	6	30	2	18.2114	22.1954
		6	50	3	21.3350	22.1954
		8	30	2	24.3428	22.1954
		8	50	4	22.9037	22.1954
		12	30	4	18.6853	22.1954
		12	50	6	20.3662	22.1954
		16	30	5	22.1244	22.1954
		16	50	8	21.0507	22.1954

(a) Frame 1

(b) Frame 2

(c) Frame 3

(d) Frame 4

(e) Frame 5

(f) Frame 6

(g) Frame 7

Figure A3. Seven consecutive frames from a high bee traffic video; the frames are numbered row by row from top left down as Frame 1–7; a fast moving bee is delineated in each frame by a green polygon; the bee is moving right to left; the x, y coordinates of the green dot in each polygon in the corresponding frames are: $x = 507.0$, $y = 1782.0$ in Frame 1; $x = 448.5$, $y = 1569.0$ in Frame 2; $x = 427.5$, $y = 1351.5$ in Frame 3; $x = 433.5$, $y = 1104.0$ in Frame 4; $x = 486.0$, $y = 778.5$ in Frame 5; $x = 580.5$, $y = 444.0$ in Frame 6; $x = 660.0$, $y = 151.5$ in Frame 7.

Table A14. DTW values for *LT_Vid.mp4* with image size of 160×120 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
160×120	8×8	6	30	2	21.5966	18.1374
		6	50	3	21.6198	18.1374
		8	30	2	23.3670	18.1374
		8	50	4	24.0971	18.1374
		12	30	4	22.8642	18.1374
		12	50	6	25.0676	18.1374
		16	30	5	22.9499	18.1374
		16	50	8	22.1025	18.1374

Table A15. DTW values for *NT_Vid.mp4* with image size of 160×120 ; C_1^* is DPIV; C_2^* is MOG2/VGG16; and C_3^* is Human Count (ground truth).

Img. Size	Bee Size	Inter. Win. Size	% Overlap	Pix. Overlap	DTW(C_1^*, C_3^*)	DTW(C_2^*, C_3^*)
160×120	8×8	6	30	2	18.4938	6.4764
		6	50	3	17.3970	6.4764
		8	30	2	16.2854	6.4764
		8	50	4	17.7443	6.4764
		12	30	4	12.2254	6.4764
		12	50	6	11.8504	6.4764
		16	30	5	12.0754	6.4764
		16	50	8	11.7858	6.4764

References

1. Winston, M. *The Biology of the Honey Bee*; Harvard University Press: Cambridge, MA, USA, 1987.
2. Dadant, C. *First Lessons in Beekeeping*; Charles Scribner's Sons: New York, NY, USA, 1980.
3. Page, R., Jr. The spirit of the hive and how a superorganism evolves. In *Honeybee Neurobiology and Behavior: A Tribute to Randolph Menzel*; Galizia, C.G., Eisenhardt, D., Giurfa, M., Eds.; Springer: Berlin, Germany, 2012; pp. 3–16. [\[CrossRef\]](#)
4. Kulyukin, V.; Putnam, M.; Reka, S. Digitizing buzzing signals into A440 piano note sequences and estimating forager traffic levels from images in solar-powered, electronic beehive monitoring. In *Lecture Notes in Engineering and Computer Science, Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, China, 16–18 March 2016; Newswood Limited: Hong Kong, China, 2016; Volume 1, pp. 82–87.
5. Kulyukin, V.; Reka, S. Toward sustainable electronic beehive monitoring: Algorithms for omnidirectional bee counting from images and harmonic analysis of buzzing signals. *Eng. Lett.* **2016**, *24*, 317–327.
6. Kulyukin, V.; Mukherjee, S. On video analysis of omnidirectional bee traffic: Counting bee motions with motion detection and image classification. *Appl. Sci.* **2019**, *9*, 3743. [\[CrossRef\]](#)
7. Langstroth Beehive. Available online: https://en.wikipedia.org/wiki/Langstroth_hive (accessed on 1 November 2019).
8. Adrian, R.J. Particle-imaging techniques for experimental fluid mechanics. *Annu. Rev. Fluid Mech.* **1991**, *23*, 261–304. [\[CrossRef\]](#)
9. Willert, C.E.; Gharib, M. Digital particle image velocimetry. *Exp. Fluids* **1991**, *10*, 181–193. [\[CrossRef\]](#)
10. Spedding, G.R.; Hedenström, A.; Johansson, L.C. A note on wind-tunnel turbulence measurements with DPIV. *Exp. Fluids* **2009**, *46*, 527–537. [\[CrossRef\]](#)

11. Henningsson, P.; Spedding, G.R.; Hedenström, A. Vortex wake and flight kinematics of a swift in cruising flight in a wind tunnel. *J. Exp. Biol.* **2008**, *211*, 717–730. [CrossRef] [PubMed]
12. Hedenström, A.; Johansson, L.C.; Wolf, M.; von Busse, R.; Winter, Y.; Spedding, G.R. Bat flight generates complex aerodynamic tracks. *Science* **2007**, *316*, 894–897. [CrossRef]
13. Spedding, G.R.; Rosén, M.; Hedenström, A. A family of vortex wakes generated by a thrush nightingale in free flight in a wind tunnel over its entire natural range of flight speeds. *J. Exp. Biol.* **2003**, *206*, 2313–2344. [CrossRef]
14. Muijres, F.T.; Johansson, L.C.; Barfield, R.; Wolf, M.; Spedding, G.R.; Hedenström, A. Leading-edge vortex improves lift in slow-flying bats. *Science* **2008**, *319*, 1250–1253. [CrossRef]
15. Hubel, T.Y.; Riskin, D.K.; Swartz, S.M.; Breuer, K.S. Wake structure and wing kinematics: The flight of the lesser dog-faced fruit bat, *Cynopterus brachyotis*. *J. Exp. Biol.* **2010**, *213*, 3427–3440. [CrossRef]
16. Dickinson, M.; Lehmann, F.; Sane, S. Wing rotation and the aerodynamic basis of insect flight. *Science* **1999**, *284*, 1954–1960. [CrossRef] [PubMed]
17. Bomphrey, R.J.; Lawson, N.J.; Taylor, G.K.; Thomas, A.L.R. Application of digital particle image velocimetry to insect aerodynamics: Measurement of the leading-edge vortex and near wake of a hawkmoth. *Exp. Fluids* **2006**, *40*, 546–554. [CrossRef]
18. Michelsen, A. How do honey bees obtain information about direction by following dances? In *Honeybee Neurobiology and Behavior: A Tribute to Randolph Menzel*; Galizia, C.G., Eisenhardt, D., Giurfa, M., Eds.; Springer: Berlin, Germany, 2012; pp. 65–76. [CrossRef]
19. Rodriguez, I.F.; Megret, R.; Egnor, R.; Branson, K.; Agosto, J.L.; Giray, T.; Acuna, E. Multiple insect and animal tracking in video using part affinity fields. In Proceedings of the Workshop Visual Observation and Analysis of Vertebrate and Insect Behavior (VAIB) at International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018.
20. Babic, Z.; Pilipovic, R.; Risojevic, V.; Mirjanic, G. Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 51–57. [CrossRef]
21. Kulyukin, V.; Mukherjee, S.; Amlathe, P. Toward audio beehive monitoring: Deep learning vs. standard machine learning in classifying beehive audio samples. *Appl. Sci.* **2018**, *8*, 1573. [CrossRef]
22. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780. [CrossRef]
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
24. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (AAAIWS'94), Seattle, WA, USA, 31 July–4 August 1994; pp. 359–370.
25. Salvador, S.; Chan, P. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* **2007**, *11*, 561–580. [CrossRef]
26. Kulyukin, V. BeePi: A Multisensor Electronic Beehive Monitor. Available online: <https://www.kickstarter.com/projects/970162847/beepi-a-multisensor-electronic-beehive-monitor> (accessed on 2 January 2020).
27. Kulyukin, V. BeePi: Honeybees Meet AI: Stage 2. Available online: <https://www.kickstarter.com/projects/beepihoneybeesmeetai/beepi-honeybees-meet-ai-stage-2> (accessed on 2 January 2020).
28. Tiwari, A.; Kulyukin, V. BEE1: A Dataset of 54,383 Labeled Images of Honeybees Obtained from BeePi, a Multi-Sensor Electronic Beehive Monitor. Available online: <https://usu.box.com/s/p7y8v95ot9u9jvbbayci61no3lzbqx3> (accessed on 27 December 2019).
29. Vats, P.; Kulyukin, V. BEE2_1S: A Dataset of 54,201 Labeled Images Of Honeybees Obtained from BeePi, a Multi-Sensor Electronic Beehive Monitor, Mounted on Top of the First Super of a Langstroth Beehive. Available online: <https://usu.box.com/s/p7y8v95ot9u9jvbbayci61no3lzbqx3> (accessed on 27 December 2019).
30. Vats, P.; Kulyukin, V. BEE2_1S: A Dataset of 54,678 Labeled Images of Honeybees Obtained from BeePi, a Multi-Sensor Electronic Beehive Monitor, Mounted on Top of the Second Super of a Langstroth Beehive. Available online: <https://usu.box.com/s/3ccizd5b1qzcqcs4t0ivawmrxbgva7ym> (accessed on 27 December 2019).
31. Adrian, R. Twenty years of particle image velocimetry. *Exp. Fluids* **2005**, *39*, 159–169. [CrossRef]

32. Klewicki, J. Measurement considerations in wall-bounded turbulent flows: Wall shear stress. In *Handbook of Experimental Fluid Mechanics*; Tropea, C., Yarin, A.L., Foss, J.F., Eds.; Springer: Berlin, Germany, 2007.
33. Wieneke, B. PIV uncertainty quantification from correlation statistics. *Meas. Sci. Technol.* **2015**, *26*. [[CrossRef](#)]
34. Siacchitano, A.; Neal, D.; Smith, B.; Warner, S.O.; Vlachos, P.; Wieneke, B.; Scarano, F. Collaborative framework for PIV uncertainty quantification: Comparative assessment of methods. *Meas. Sci. Technol.* **2015**, *26*. [[CrossRef](#)]
35. McAfee, A. Honey, let me tell you about this city. *Am. Bee J.* **2019**, *159*, 521–524.
36. Dadant Beehive. Available online: https://en.wikipedia.org/wiki/Charles_Dadant (accessed on 1 November 2019).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).