BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ÚSTAV AUTOMATIZACE A MĚŘÍCÍ TECHNIKY

SEMESTRIAL PROJECT – RECOGNITION

SEMESTRÁLNÍ PRÁCE – ROZPOZNÁVÁNÍ

AUTHORS

AUTOŘI PRÁCE

Bc. Dominik Ficek
Bc. David Makówka

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Šimon Bilík

BRNO 2022

Introduction

The aim of this work is to explore solutions for tracking bees for the purposes of counting the number of bees arriving to and leaving a hive through a monitored environment. The final solution should be applicable on an embedded device allowing real-time processing at a 5 Hz sampling frequency.

In this work we discuss the usage of motion segmentation and other motion tracking techniques for our task. Our proposed solution stands on motion signal analysis and is discussed in the second chapter of this document.

1 Motion tracking techniques

Our task stands on analyzing the movement of individual objects, bees traversing through a predefined environment. There are numerous existing techniques tackling this problem, each with its own sets of pros and cons. In this chapter we briefly discuss practical applicability of several techniques on our task.

With the uprising of convolutional neural networks and specifically object detectors many works focus on tracking objects with direct access to detector's predictions on each frame, for example the Deep SORT [1] algorithm. This approach is not suitable for our cause as convolutional neural networks are computationally expensive.

Another motion analysis technique is to use initial state of objects in their first appearance in analyzed sequence to track them though the sequence using just visual information. A common approach for this problem is to use Siamese neural networks [2] that formulate the problem as convolutional feature cross-correlation between objects and region of interest. Again this approach requires the use of computationally expensive neural networks and further more due to the large bee traffic though the observed alley the initialization step would've had to be ran on every time step, increasing the computational cost even more.

Some works focus on analyzing temporal information though a low-dimensional subspace of ambient space with clustering techniques, these techniques are known as subspace clustering techniques [3] and are used to separate low-dimensional data to their parent subspaces. This approach seems to be feasible for our needs at the first glance but is not applicable due to numerous reasons. The most common problem with subspace clustering variants is that they set requirements on the low-dimensional data that are not realistic for us to meet, for example they require prior knowledge of number of subspaces [4, 5] or prior knowledge of number of data points each of the subspaces contain [6]. Other approaches generally rely on building a similarity matrix from input data matrix [7, 8, 9], this approach is also not suitable for our needs as our low-dimensional data points would be extracted with SIFT-like algorithm, resulting in inconsistencies in subsequent data points between time steps, both in terms of their location and their quantity. Also, it's worth to mention that as the objects of interest are bees, which are visually difficult to distinguish from each other even for a human, relying on local features does not seem like the correct approach, especially when considering our low sampling frequency which further lowers the temporal information each feature's geometric properties carry.

2 Proposed solution

As none of the previously mentioned motion tracking techniques proved to be suitable for our needs we deviate from analyzing a movement of individual objects of interest and utilize our prior knowledge of environment and analyze only an overall movement in region of interest.

In order to keep a reliable representation of our environment even after a longer period of time we utilize dynamic background model. We initialize our dynamic model $m(x, y, k)$ optionally with the first frame of the sequence or with a previously captured frame of the environment $f(x, y, k = 0)$.

$$m(x, y, 0) = f(x, y, 0) \tag{2.1}$$

In each time step of the sequence we analyze the overall dynamic properties of the frame with thresholded subtraction

$$d_1(x, y, k) = \begin{cases} 1 & \text{if } |f(x, y, k) - f(x, y, k-1)| > T_1 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where T_1 is empirically selected threshold. And based on the overall number of dynamic pixels we flag the scene as either dynamic or non-dynamic

$$D_1(k) = \sum_{x,y} d_1(x, y, k) > T_2 \tag{2.3}$$

where T_2 is again empirically selected threshold. Our dynamic model is updated only when the scene is flagged as static in sequence, meaning D_1 is flagged as *false*, and the scene is flagged as static with respect to the dynamic model. This flag is calculated in similar fashion as the D_1 flag.

$$d_2(x, y, k) = \begin{cases} 1 & \text{if } |f(x, y, k) - m(x, y, k-1)| > T_1 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

$$D_2(k) = \sum_{x,y} d_2(x, y, k) > T_2 \tag{2.5}$$

The dynamic model is then updated with simple adaptive filter

$$m(x, y, k) = \begin{cases} \alpha \cdot f(x, y, k) + (1 - \alpha) \cdot m(x, y, k-1) & \text{if } D1(k) \text{ and } D2(k) \\ m(x, y, k-1) & \text{otherwise} \end{cases} \tag{2.6}$$

where α is the learning rate of the dynamic model.

To track the level of dynamic activity in our region of interest, we inspect the ratio of dynamically flagged pixels in subtracted frame from the current time step with our dynamic environment model to the number of pixels in the region. However, this metric alone does not carry any information in terms of the movement's direction. To compensate this we split the inspected region into N sections along the y axis.

$$\begin{gathered} f(x,y) \to g(x,y,n) \\ \mathbb{R}^{\mathrm{X}\times\mathrm{Y}} \to \mathbb{R}^{\mathrm{X}\times\lfloor\frac{\mathrm{Y}}{\mathrm{N}}\rfloor\times\mathrm{N}} \end{gathered} \tag{2.7}$$

In each of these N sections we calculate the already mentioned metric of dynamic activity

$$d(x,y,n,k) = \begin{cases} 1 & \text{if } |f(x,y,n,k) - m(x,y,n,k)| > T_1 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$r(n,k) = \frac{1}{X \cdot Y} \sum_{x,y} d(x,y,n,k) \tag{2.9}$$

this $r(n,k)$ signal now carries enough information to determine the level and direction of movement in observed region. To further ease this signal's processing we approximate first order partial derivative with respect to the time step dimension with differentiation

$$dr(n,k) = r(n,k) - r(n,k-1) \tag{2.10}$$

In the resulting signal $dr(n,k)$ we threshold its peaks to classify the current timestep with one of three classes: bee arrival ($class = 1$), idle state ($class = 0$) and bee departure ($class = -1$).

$$class(n,k) = \begin{cases} 1 & \text{if } dr(n,k) > T_3 \\ -1 & \text{if } dr(n,k) < -T_3 \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

where $T_3 \in \langle 0, 1 \rangle$ is empirically selected threshold value.

To implement the actual counter of bees that traverse the region in one way or the other we keep track of classes from the last K_{max} time steps. On each bee arrival we add a track to a list and with each bee departure we flag an unflagged track as valid. Once a valid track is present in all N sections, we increment a counter. The direction of the bee's movement is based on the age of the valid tracks on the edges of the region.

As this is quite a simple approach, it's bound to have limitations, its main disadvantage is that if the bees do not travel independently but in packs, $r(n,k)$ will remain close to constant, $dr(n,k)$ will be close to zero and the bees won't be accounted for as $dr(n,k)$ does not cross T_3 threshold value. Other limitation that we have observed in experiments is that once a bee slows down at some point of its traverse through the observed area or the lightning in the observed area lowers its intensity, the $dr(n,k)$ values reduce sometimes even to a noise level. This leads to a missing track entry in one or more sections of a tunnel, the bee is not accounted for and there may be hanging tracks left in sections where the bee was registered. This effect can have detrimental effect on next registered bees as hanging tracks on the edges of the observed region define estimated traverse direction. This effect can be suppressed by lowering the maximum track age K_{max}, but lowering this value also effectively reduces sensitivity when the bee's velocity lowers. On the bright side this approach runs under 20ms on Raspberry Pi 4B, sequentially processing 12 bee tunnels in each frame.

Conclusion

In this work we tackle a problem of counting bees leaving and arriving to a hive. After a brief summary of motion tracking techniques in the first chapter we decided not to use any of them as they are not very suited for our needs, we describe our solution in the second chapter. The solution lies in analyzing overall movement in multiple sections of region of interest to detect general movement and determine its direction. The algorithm's limitations are discussed at the end of the second chapter, and it can be run on an embedded device as a part of online processing pipeline.

Bibliography

[1] N. Wojke, A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645-3649, doi: 10.1109/ICIP.2017.8296962.

[2] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing and J. Yan, "SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4277-4286, doi: 10.1109/CVPR.2019.00441.

[3] E. Elhamifar and R. Vidal, "Sparse Subspace Clustering: Algorithm, Theory, and Applications," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2765-2781, Nov. 2013, doi: 10.1109/TPAMI.2013.57.

[4] J. Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., 2003, pp. I-I, doi: 10.1109/CVPR.2003.1211332.

[5] Teng Zhang, A. Szlam and G. Lerman, "Median K-Flats for hybrid linear modeling with many outliers," 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009, pp. 234-241, doi: 10.1109/ICCVW.2009.5457695.

[6] Sugaya, Yasuyuki and Kenichi Kanatani. "Geometric structure of degeneracy for multi-body motion segmentation." International Workshop on Statistical Methods in Video Processing. Springer, Berlin, Heidelberg, 2004.

[7] Yan, Jingyu and Marc Pollefeys. "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.

[8] A. Goh and R. Vidal, "Segmenting Motions of Different Types by Unsupervised Manifold Clustering," 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1-6, doi: 10.1109/CVPR.2007.383235.

[9] Ng, Andrew, Michael Jordan and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems 14 (2001).