Abstract

Tracking multiple near-identical objects can be accomplished by finding the locations of each object in each video frame and then establishing object correspondences across all frames. In this work, we present a two-step approach to track multiple near-identical objects that move freely in a controlled environment. We first identify object locations in each frame using a robust background subtraction algorithm. After the locations of the objects have been identified, we find object correspondences across frames using the Hungarian algorithm. This tracking system demonstrates outstanding performance when tracking flies that move freely in a transparent container.

Introduction

In computer vision, object tracking techniques have been studied for years. Effective and efficient algorithms have been created to track single and multiple objects under different environmental conditions [1,2]. Although such techniques have shown to be robust, they usually require the use of expensive equipment or assume that the objects of interest have significant morphological differences. These requirements are very limiting for potential adopters of these techniques.

Scientists that are interested in tracking animals often perform this task in controlled laboratory environments. For example, biologists conducting experiments on flies or ants are in control of aspects such as the camera location, illumination, and background. We propose to take advantage of these characteristics to create a robust and efficient tracking algorithm that minimizes the overall computational cost of the task.

Methods

We divide the fly tracking process into the following two steps:

1. Finding the locations of each object in each video frame
2. Establishing object correspondences across all frames
1. The following steps describe our approach to effectively and efficiently identify flies in a given video frame.

![Fig 1. The video is divided into n segments.](image)

2. Once the objects of interest have been detected in all frames, we use the Hungarian algorithm [3] to link such detections. The following steps show how we tackle this problem.

![Fig 5. All flies in consecutive pairs of frames are identified.](image)

3. Flies from both frames are associated. Each entry of a matrix is filled with the probability of each association.

![Fig 6. Flies from both frames are associated.](image)

4. The Hungarian algorithm [3] is used to find the permutation with the highest probability.

![Fig 7. A basic model is created by tracking one fly.](image)

Experimental Data

The Neuroscience and Metabolic Disorders research group at UTEP provided us with a set of recordings of different experiments performed on flies.

- Resolution: 320x288
- Frames per second: 15
- Number of flies: 10
- Number of frames: 3765

![Fig 9. Experimental data properties](image)

Results

We parallelized both steps of our approach and used a combination of Matlab and C for efficiency purposes. This allowed us to process the data in conventional computer at a speed of 40 frames per second. The number of frames where the 10 flies were correctly identified were 3241 (86.08%). Additionally, we were able to process more than 2000 matrices per second using the Hungarian algorithm [3] with a correspondence accuracy of 100%.

![Fig 10. Tracking results where color-coded boxes are used to illustrate fly associations.](image)

Conclusion and Future Work

We have shown that tracking multiple near-identical objects can be performed effectively and efficiently using a conventional computer. We showed how this approach can be used to successfully track flies in conventional videos that scientists record when conducting experiments on these insects.

To further improve the effectiveness of this approach, we need to refine our techniques to account for collisions and interactions among the objects being tracked. In addition, incorporating more informative metrics that measure the performance of our approach from different perspectives, such as the ones proposed by Bernardin et al. [4], would allow potential adopters of this approach to better evaluate it.

References


Acknowledgments

This work was supported by the Research Experiences for Undergraduates (REU) Program of the National Science Foundation under Award Number IIS-0852066 and the Computing Alliance of Hispanic Serving Institutions (CAHIS) under Award Number CNS-1042341. Thanks to Dr. Kyung-An Han and Dr. Young-Chung Kim for introducing us to the problem, providing videos used for this program, and providing valuable guidance.