Multiple Mice Tracking Using Microsoft Kinect

by

Chun-Kai Wang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author ....
Department of Electrical Engineering and Computer Science
May 28, 2013

Certified by ....
Tomaso Poggio
Eugene McDermott Professor
Thesis Supervisor

Accepted by ....
Prof. Dennis M. Freeman
Chairman, Masters of Engineering Thesis Committee

Multiple Mice Tracking Using Microsoft Kinect

by

Chun-Kai Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 28, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Mouse tracking is integral to any attempt to automate mouse behavioral analysis in neuroscience. Systems that rely on vision have successfully tracked a single mouse in one cage[10], but when attempting to track multiple mice, video-based systems often struggle when the mice interact physically.

In this thesis, I develop a novel vision-based tracking system that addresses the challenge of tracking multiple deformable mice with identical appearance, especially during complex occlusions. The system integrates both image and depth modalities to identify the boundary of two occluding mice, and then performs pose estimation to locate nose and tail locations of each mouse.

Detailed performance evaluation shows that the system is robust and reliable, with low rate of identity swap after each occlusion event and accurate pose estimation during occlusion. To evaluate the tracking system, I introduce a dataset containing two 30-minute videos recorded with Microsoft's Kinect from the top view. Each video records the social reciprocal experiment of a pair of mice.

I also explore applying the new tracking system to automated social behavior analysis, by detecting social interactions defined with position- and orientation-based features from tracking data. The preliminary results enable us to characterize lowered social activity of the Shank3 knockout mouse, and demonstrate the potential of this system for quantitaive study of mice social behavior.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Acknowledgments

I am grateful for all the help and support I received along the way, without which I would not have been able to complete this thesis.

First of all, I have to thank Professor Tomaso Poggio for giving me the opportunity to join CBCL. I have been enjoying the experience as a master student in this lab, and without a doubt I have learned a great deal in the meantime.

I am grateful for the great times and memories that my CBCL colleagues brought me. I enjoy the dinner times with Allen, Guille, and Youseff, which often cheered me up from a long exhausting day. I remember that Andrea offered to help annotate data when I wished time could stop ticking before the paper submission. Especially, I cannot thank Charlie Frogner enough, who is a great mentor throughout my MEng life. In spite of his busy schedule, he still made time to discuss with me about the project. I always received invaluable feedback from him, and whenever I felt stuck with my work, he was the one who inspried a different direction.

I appreciate the great help from people in collaboration with the mouse phenotyping project, including Kim Maguschak, Dongqing Wang, and Holly Robertson from the Guoping Feng lab, Jorge Castro Garcia and Ikue Nagakura from the Mriganka Sur lab, and Swarna Pandian from the Constantine-Paton lab. I could continue my research thanks to their kindly assistance with experiments and equipment.

Last but not the least, I would like to thank my family in Taiwan for their unconditional support and encouragement that have been driving me towards my dream.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Social Behavior Analysis

As an essential step to treating a variety of social disorders, such as autism, neuroscientists have been studying social interactions of mice in order to gain more insight into social disorders such as autism [5]. For such purpose, mouse behavioral analysis becomes a widely used tool to investigate the connection between a gene and the symptoms. By observing behaviors of a line of mutants, researchers study the cause of certain diseases, which facilitates the development of new medications that mitigate the disease by changing the activity of the gene. For instance, Shank3-mutant mice are found to display autism-like behaviors, such as excessive grooming and deficits in social behaviors [15].

Recently, automated behavioral analysis has received much attention. Traditionally, behavioral analysis involves human monitoring mouse behaviors over long periods of time. However, manual behavioral assessment consumes a considerable amount of human hours, and the time scales of data are limited to minutes instead of days due to the tedious nature of human monitoring. This becomes the bottleneck of research in this field [17], and automated behavioral analysis is recognized as a more time- and cost-effective alternative for study. An accurate system targeting mice social behavior analysis is therefore desirable.

1.2 Multiple Mice Tracking

Multiple mice tracking is an integral part of social behavior analysis. The tracking system computes relative positions and poses of the tracked mice, which are used as features to discriminate among social behaviors. For example, a small distance between the noses of two mice might imply a nose-to-nose sniffing behavior. Since the quality of behavior analysis depends on the tracking output, developing a reliable tracking system is the first step to automating behavior analysis.

Accurate mouse tracking during occlusions is especially important for action recognition and behavioral analysis. Many types of social interactions between two mice occur when they are in contact with or even occluding each other, from nose-to-nose and nose-to-anogenital encounters to social grooming. If the body of each interacting mouse cannot be located precisely, it would be difficult to capture these actions. For instance, precise head location is necesarry to distinguish between nose-to-nose sniffing and nose-to-head sniffing.

In the meantime, tracking through occlusions is the most challening part of multiple mice tracking. As the mice are nearly identical in appearance and their bodies are almost textureless, the tracking system can easily swap mouse identities during complex social interactions. In such events, existing social behavior analysis systems still have high rate of identity swaps or inaccurate poses during occlusions. Human interventions are thus required to fix these errors, defeating the purpose of minimal human labor.

1.3 Contributions

Owing to the motivations above, I devote most of this thesis to solving challenges that arise from tracking multiple mice in one cage. I present a computer vision system that tracks the contours and head/tail locations of two mice identical in appearance given the intensity and depth image sequences from the Kinect. For each frame, the system first computes the foreground, and then segments the foreground into two mouse regions by utilizing the Kinect's depth information.

The system can track reliably during both occlusion and non-occlusion frames, with minimal human assistance. I provide detailed performance comparison of my approach with human-labeled ground-truth and other existing systems. Figure 1-1 shows two examples of the tracking results.

The following summarizes the main contributions of this thesis.

1. Tracker for multiple mice with identical appearance.
    - Developed a tracking system that tracks through complex occlusions by integrating intensity and depth modality.
    - Developed a pose estimator based on a trainable statistical shape model.
    - Evaluated the performance of the system by comparing the tracking results with other existing methods and ground-truth.
2. Multiple Mice tracking dataset.
    - Recorded two 30-minute videos that include image frames and depth frames using Microsoft's Kinect.
    - Collected human labels of the head/tail locations and the mouse contours of 1000 sampled frames in the dataset.
3. Preliminary automated behavior analysis.
    - Proposed 15 behavior rules based on 13 position- and orientation-based features.
    - Demonstrated behavior analysis on my dataset based on the results of the multiple mice tracker.

1.4 Outline

The thesis is organized as follows: In Chapter 2 I briefly introduce related work and existing methods on mouse tracking, and explain the challenges of multiple mice tracking. In Chapter 3 I present and elaborate on my tracking system. In Chapter 4 I show performance assessment on the system as well as comparison with other methods. In Chapter 5 I demonstrate using my tracking system to detect social interactions. Finally, in Chapter 6 I provide discussions on future research directions.

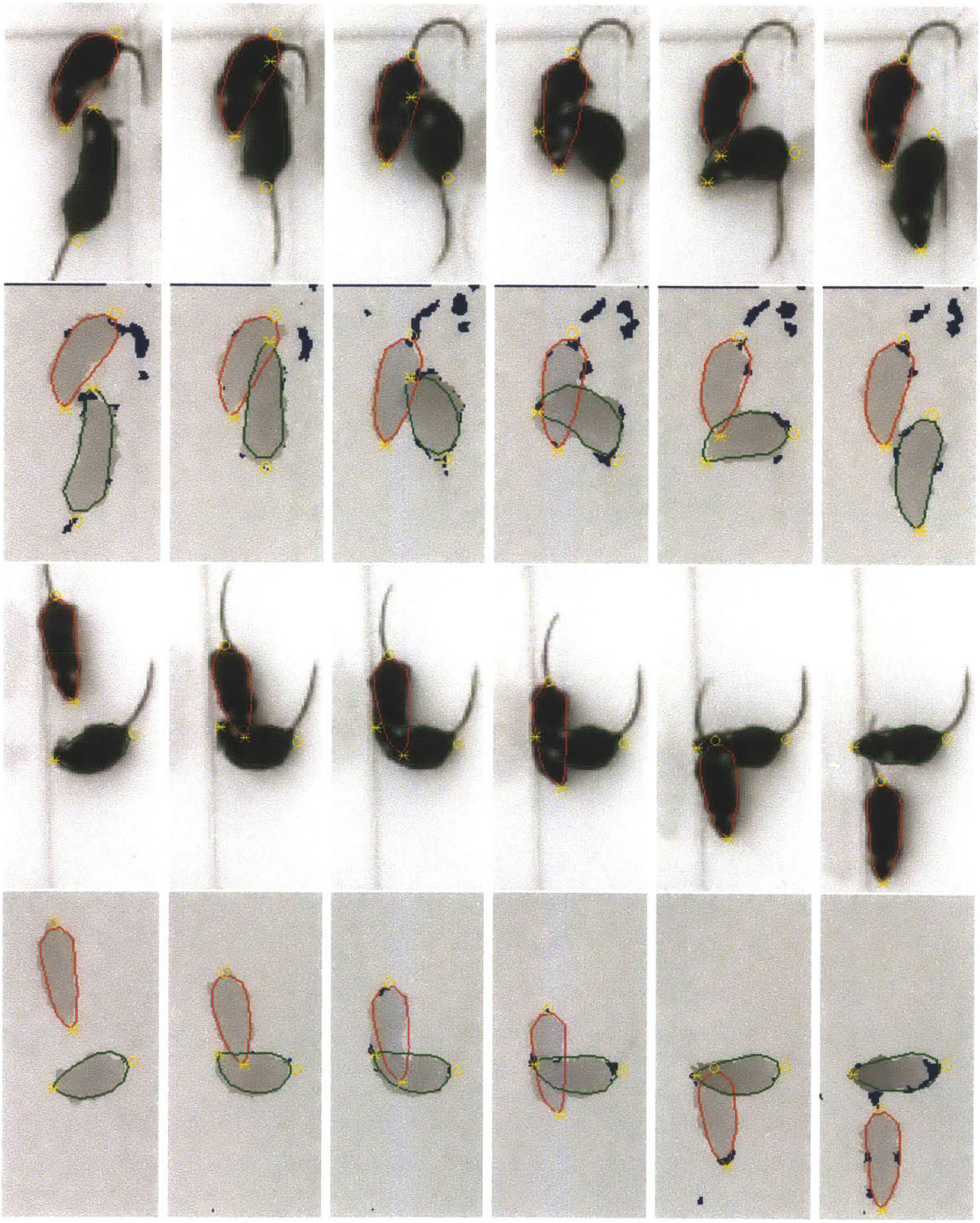


Figure 1-1: This figure illustrates the results of my tracking system during occlusion frames. The contour of each mouse is traced in red or green, and the noses and tails are marked by yellow stars and circles.

Chapter 2

Background and Related Work

In this chapter, I begin by introducing previous work related to mouse tracking. Then I explain the challenges of mouse tracking, the conditions for a good tracker, and how the additional depth information addresses these challenges.

2.1 Work Related to Mouse Tracking

Vision-based systems have been developed to track a single mouse in various settings. Jhuang et al. [10] locate and track the mouse using simple static background subtraction, operating under fixed lighting and with black or brown mice. Farah et al. achieve more robustness to variations in illumination, cage setup and mouse color by using a number of computer vision features, including edglets for contour refinement [8].

For tracking multiple mice, a number of vision-based methods have been proposed. Goncalves et al. [9] and Pistori et al. [16] combine k-means and particle filters to track bounding ellipses; Edelman similarly relies on fitting an ellipse model [7]. Braun's method matches the foreground shape to pre-labeled templates, to locate body parts like nose-tip and tail-base [2]. These methods work well when the degree of occlusion of one mouse by another is small, but they are highly unreliable whenever there is significant physical interaction.

Branson and Belongie use a particle filter to track mouse "blob" locations and contours, for multiple mice. Their method relies on training data and is computationally more expensive than those described above [1]. A system proposed by de Chaumont et al. uses a physics-based model to track the skeleton of each mouse with physical constraints imposed by the skeleton's joints [6]. Their system, however, is not robust under prolonged interactions between the mice and requires human assistance to correct the tracking.

Commercial systems from CleverSys and Noldus also address this problem. However, Noldus' EthoVision only tracks non-occlusion frames, and requires color marking visible to the camera on each mouse to maintain correct mouse identities. Other details of their methods are not available.

There exist previous attempts to evaluate the use of depth cameras for detecting mice behavioral patterns. Monteiro et al. show preliminary results of depth-based segmentation [12], and OuYang et al. develop a system that reconstructs continuous high-resolution 3D location and measurements of a mouse using Kinect [13]. However, these works only address tracking a single mouse. This thesis, on the other hand, addresses tracking through complex occlusions among multiple mice by integrating depth modality.

2.2 Challenges of Tracking Multiple Mice

Compared with tracking single mouse in one cage, tracking multiple mice through a sequence of intensity images is much more challenging for a number of reasons, as I describe in the following. Figure 2-1 shows some frames that are difficult for tracking.

The mice are textureless. The body of a mouse is almost textureless, so approaches that rely solely on local appearances models are unreliable in our problem. The upper row in Figure 2-2 demonstrates such challenge. In each example, the occlusion edge between the two mice is faint in the intensity image, and most computer vision segmentation techniques cannot identify the contour of the occluding mouse.

The mice are often nearly identical in appearance. When performing social interaction tests, scientists usually use genetically similar strains, and thus they have little difference in appearance and color.

The mice can have complex occlusions. As mice may roll around or crawl over/under each other, segregating their body contours from a top view is very challenging.

The mice are highly deformable. The apparent shape, size, orientation of each mouse can change significantly over time. In addition, the often sudden shifts in direction and velocity of motion further complicate the task.

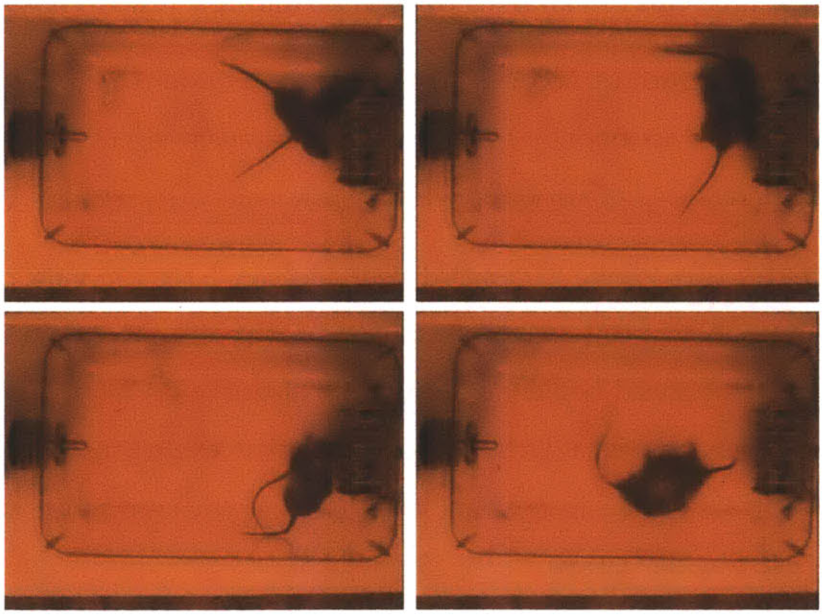


Figure 2-1: Some challenging frames for tracking. (Figure obtained from [7].)

2.3 Design Considerations

As the tracking system is designed specifically to automate social behavior analysis, there are several requirements to consider, which I elaborate as follows.

Assumes no modification to the animals. As we want to observe animal behaviors in their most nature state, the tracking system assumes no paint or mark is applied to the mice. Also, the markers can be hidden from the camera when animals are in contact, making them unreliable for resolving mouse identities, especially during complex physical interactions.

Preserves mouse identity. The system has to maintain a track for each mouse, and furthermore, each track should correctly corresponds to the same mouse identity throughout the video.

Provides precise head and tail tracking. Head and tail locations serve as important features to recognize social behaviors, and precise estimates of these body parts can help distinguish many social behaviors. For instance, we would expect the nose tips of two mice to be close during a nose-to-nose encounter.

Requires minimal human intervention. The system should operate using as little human labor as possible; this means tracking needs to be robust against events that require human intervention, such as identity swaps.

Tracks from the top-view. Compared with side-views, the top-view provides most information about positions, orientations, relative distances, and speeds of the mice. Since the social behaviors we wish to characterize rely on these features, top-view is the most ideal view-angle. In addition, occlusions are more frequent and complicated from side-views, thus making tracking more susceptible to failures like identity swaps.

2.4 Depth Modality

Although the intensity image alone is not sufficient for mice tracking, if we are provided with the depth map of the frame, it is possible to resolve segmentation during occlusion. When one mouse lies partly on top of another mouse, there is usually an observable height difference between the two mice. The lower row in Figure 2-2 shows the depth maps corresponding to the intensity images in the upper row. As the occlusion edges are much easier to identify with the depth images, mice tracking becomes feasible even using simple segmentation algorithms.

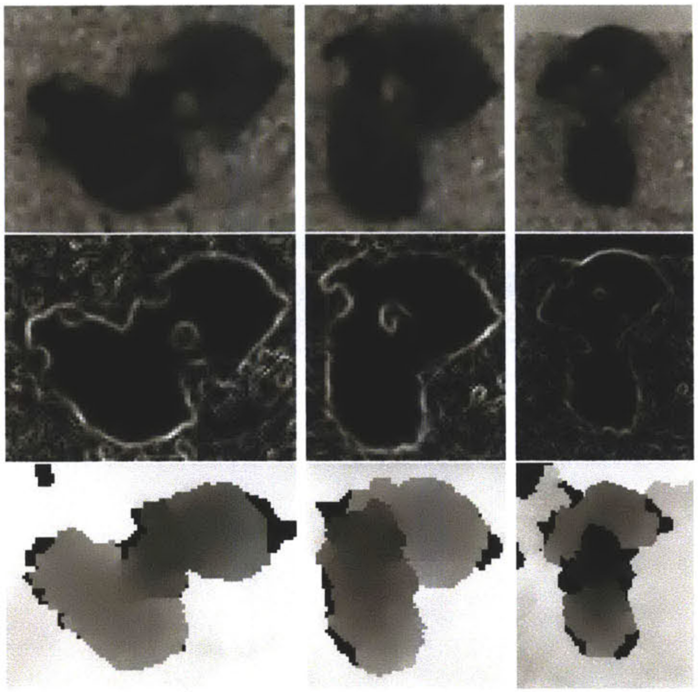


Figure 2-2: This figure illustrates the fact that intensity images alone is not sufficient to resolve occlusions, and the depth maps can often help locate occlusion edges in such cases. Shown here are the intensity images of 3 representative occlusion frames (upper row), their gradient magnitudes (middle row), and the depth images (lower row).

Microsoft's Kinect acquires both video and depth data at a resolution of 640x480 pixels and frame rate of 30 fps. It provides both an intensity image sequence and a depth image sequence, which are synchronized at the frame level and registered at the pixel level. Alternatives to the Kinect are available at similar cost: the Asus Xtion PRO is one example.

The Kinect's depth image typically contains regions where no depth information could be acquired – this is a source of noise. In our setting, this loss occurs most consistently due to reflections off of the glass on the sides of the cage, but often also includes regions of the mouse's body. This is one motivation for integrating both depth and image intensity signals for accurate foreground extraction and segmentation.

Chapter 3

Tracking System

In this chapter, I describe my method for tracking the contours and the head/tail locations of multiple mice, which utilizes both image and depth modalities of videos taken from the top-view.

3.1 System Overview

As illustrated Figure 3-1, my tracking system comprises three major stages. First, the system extracts foreground regions, using a static background model that combines the depth- and image intensity-based foregrounds for robustness. Secondly, for frames in which an occlusion is detected, the system further divides the foreground into mouse regions; this procedure starts with oversegmenting the foreground into many subregions, then the system applys a graphical model to join subregions into a full mouse contour, and then recovers occluded regions using tracking results from the previous frame. Finally, using the segmented mouse regions from the previous step, the system tracks the pose and significant body parts with a statistical shape model.

I tested the system using two mice in a single cage, but the algorithm can potentailly be extended to three or more mice.

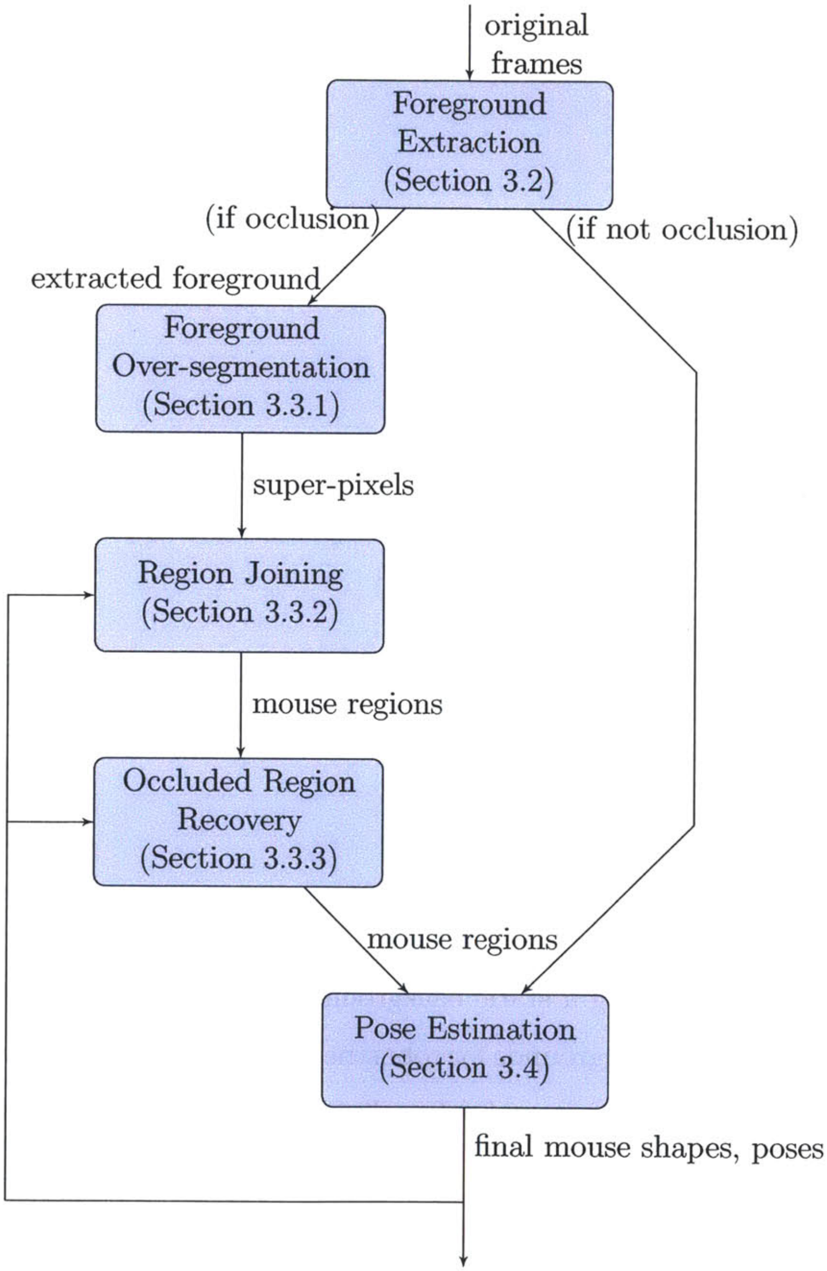

original frames
Foreground Extraction (Section 3.2)
(if occlusion)
(if not occlusion)
extracted foreground
Foreground Over-segmentation (Section 3.3.1)
super-pixels
Region Joining (Section 3.3.2)
mouse regions
Occluded Region Recovery (Section 3.3.3)
mouse regions
Pose Estimation (Section 3.4)
final mouse shapes, poses

Figure 3-1: The flow diagram of the tracking system.

3.2 Foreground Extraction

I assume a static background model – videos are acquired in a controlled setting, with camera angle, cage position and lighting fixed throughout recording. The background is modeled for each pixel by its median value over a uniform sample of video frames (Figure 3-2 (a)). Background images are computed independently for both the intensity and depth sequences.

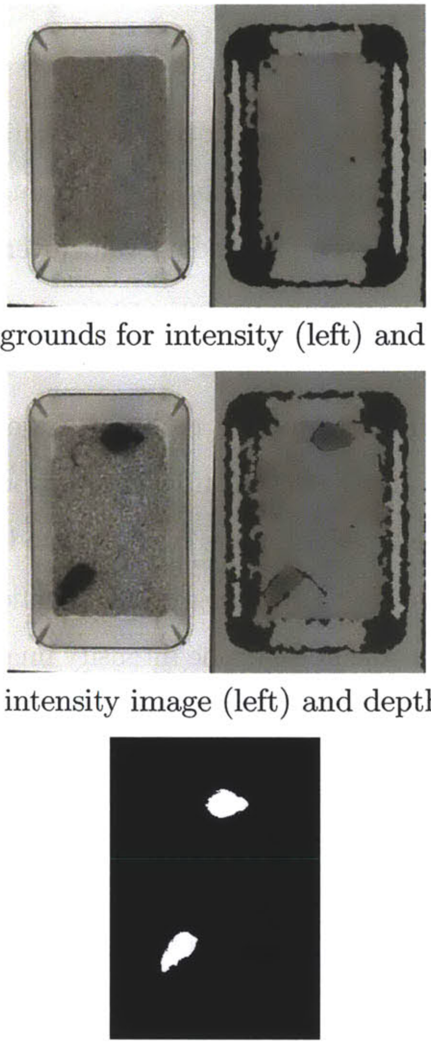
By combining the depth and intensity modalities, we eliminate the artifacts entailed by each individually. Background subtraction produces an initial estimate of the binary foreground image for each frame, computed independently for the intensity and depth images. In the depth-based foreground, significant artifacts occurs due to loss of signal. We eliminate some noise in the intensity image – arising from reflections off of the cage glass and from moving cage bedding, for example – by restricting to pixels of the same (dark) color as the mouse's body, but significant artifacts remain, as parts of the mouse's body (such as the ears) are of similar color to the background. By combining (via pixel-wise OR) the two foreground images – depth and intensity – however, we obtain a robust foreground image that eliminates the artifacts present in each individually.

The resulting foreground regions accurately capture the contours of the mice (Figure 3-2 (b-c)). Absent occlusions, then, we can choose the joint assignment of mouse identities that maximizes the summed overlaps of the mouse regions with their counterparts in the previous frame.

3.3 Segmenting Mouse During Occlusions

When one mouse occludes another, their foreground regions are connected and accurate contour tracking requires a method to find the boundary between them.

To detect occlusions, we compare the sizes of the two largest connected foreground components. If their sizes are sufficiently different, then we assume there is an occlusion. Given an occlusion, the system proceeds in 3 steps.



(a) Static backgrounds for intensity (left) and depth (right).

(b) One frame's intensity image (left) and depth image (right).

(c) Final extracted foreground.

Figure 3-2: Foreground extraction.

1. Over-segment the foreground into many subregions, each belonging to exactly one mouse.

2. Identify and join the subregions corresponding to each mouse.

3. Recover the covered portion of any occluded mouse.

3.3.1 Over-segmentation

We apply the Watershed algorithm for initial segmentation, both for computational efficiency and to preserve sharp corners often lost by contour-based algorithms [18].

The results are shown in Figure 3-3 (a-c).

Applying Watershed to the depth image alone is not sufficient: when one mouse crawls on top of the other, their combined 3D surface frequently forms a single catchment, and the algorithm fails to separate the mice. Instead the Watershed is applied to a linear combination of intensity gradient magnitudes and depth gradient magnitudes. Although the intensity gradients between the mice are typically weak, when combined with the differences in depth, the two are sufficient to resolve the boundary between the mice.

Note that, in computing depth gradients, we subtract out false gradients generated by pixels lacking valid depth measurements.

3.3.2 Region joining

The Watershed algorithm yields a (potentially large) set of foreground subregions, as shown in Figure 3-3 (c). Each should be contained within exactly one mouse and we can obtain the final segmentation by identifying each subregion with the mouse within which it's contained.

We use a probabilistic graphical model to infer the identity for each subregion. We construct a region adjacency graph, defined as an undirected graph $G = (V, E)$ in which each node $v \in V$ represents a subregion and an edge $(i, j) \in E$ exists if the subregions i and j are adjacent in the image.

We associate each node v with a random variable x_v taking on values $l \in L$, representing the mouse to which the subregion belongs. We construct node potentials ϕ_v and edge potentials ψ_{ij} so that the probability of a subregion v being assigned an identity $l \in L$ is

$$\Pr(x_v = l) \propto \\ \sum_{X \backslash \{x_v\}} \exp\left(\sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \psi_{ij}(x_i, x_j)\right)\Bigg|_{x_v = l},$$

where X is the set of all node random variables.

The node and edge potentials are defined heuristically and they are the key to accurate identity assignments. We encode in the unary node potentials the tendency of a region of the image to possess the same identity in this frame as in the previous one:

$$\phi_i(l) = w_\phi \cdot \sum_{(x,y)\in R_i^{(t)}} D((x,y); S_l^{(t-1)}),$$

where $R_i^{(t)}$ is the set of pixels in subregion i in frame t, $S_l^{(t-1)}$ is the tracked shape from the previous shape that belongs to the mouse with identity l, D is the Euclidean distance transform function, and w_ϕ is a universal weight for unary node potentials.

We encode in the edge potentials the tendency of two adjacent subregions to have different identities if the difference of the average depth values near their common border is high. Define B_{ij} to be the set of pixels lying around the common border between regions i and j. The edge potential for edge (i, j) is

$$\psi_{ij}(l_i, l_j) = \begin{cases} 0 & \text{if } l_i = l_j \\ w_\psi \cdot \left| \dfrac{1}{|B_{ij} \cap R_i|} \displaystyle\sum_{B_{ij}\cap R_i} H - \dfrac{1}{|B_{ij} \cap R_j|} \displaystyle\sum_{B_{ij}\cap R_j} H \right| & \text{if } l_i \neq l_j, \end{cases}$$

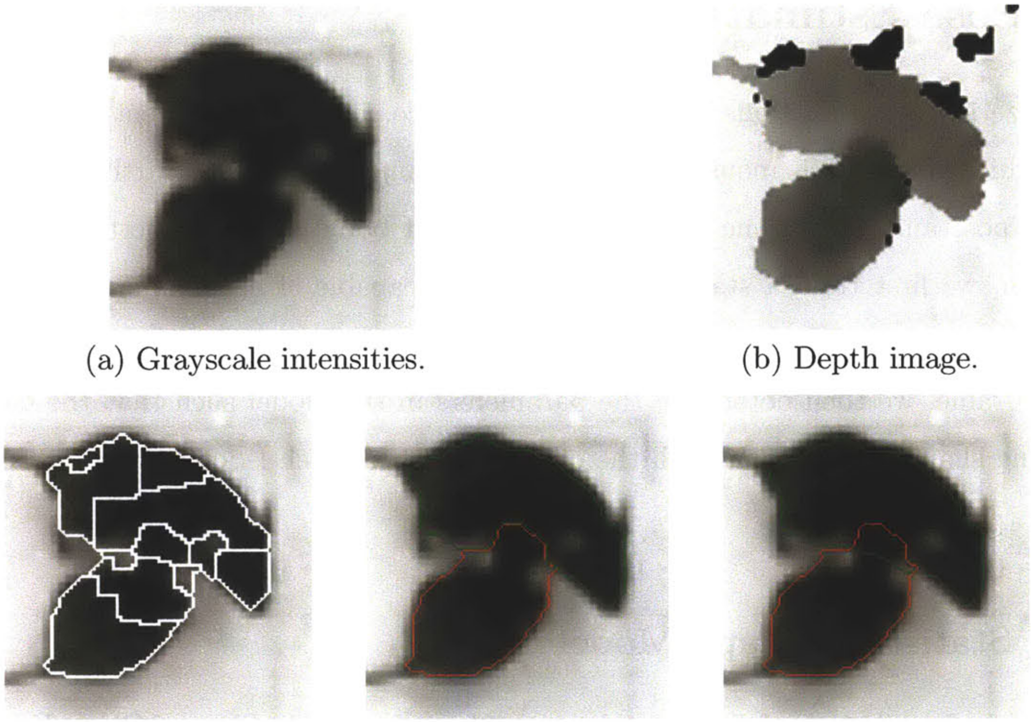
where H is the depth image and w_ψ is a universal weight for pairwise edge potentials.

Given this graph structure, we use loopy belief propagation to compute the marginal probability distribution for each subregion's identity variable x_i, and assign it the most probable identity: $\hat{l} = \text{argmax}_{l\in L} \Pr(x_i = l)$. Figure 3-3 (d) shows the result.

Having assembled the subregions into two regions representing the interacting mice, we choose the joint assignment of mouse identities that maximizes the summed overlaps of the mouse regions with their counterparts in the previous frame.

3.3.3 Occluded Shape Recovery

After each pixel seen by camera is assigned to a mouse identity, the system then recovers the total mouse shape for the occluded mouse, including parts that are hiding underneath the top mouse, so that we have a complete shape estimate for each



(a) Grayscale intensities.

(b) Depth image.

(c) Over-segmented.

(d) Final result.

(e) Shape recovered.

Figure 3-3: Segmentation during occlusion.

mouse.

First, the system has to determine which mouse is being occluded. A potential occlusion event between two mice A and B can be divided into 3 major cases: 1) A occludes B, 2) B occludes A, and 3) they are close, but neither of them is occluding the other. These cases can be roughly differentiated by comparing the mean height difference between the two mice. More precisely, if A's body near the occlusion boundary between the two mice is higher than B's, then the frame is categorized as case 1, and vice versa. The frame is considered as case 3 if the height difference does not exceed a certain threshold.

Once the system identifies a mouse as occluded, the system complements the occluded portion with the tracked shape from the previous frame, as shown in Figure 3-3 (e).

3.4 Pose Estimation

To analyze social interactions between multiple mice, it is not sufficient to know merely which pixels each mouse occupy in every frame; we need to further determine the pose and extract the locations of important body parts. To perform pose estimation, we first train a statistical shape model capable of approximating most mouse shape variations with just a few parameters, which I describe in Section 3.4.1. For each frame, we then determine the parameters in the model such that the corresponding shape is consistent with the observed shape, which I describe in Section 3.4.2.

3.4.1 Statistical Shape Model

This section describes the fundamental approach of the Active Shape Model developed by Tim Cootes [4], in which a statistical model is trained by analyzing the principal components of a set of labelled examples. This model provides a compact representation of allowable variations, but is specific enough not to allow arbitrary shapes different from the training examples.

Vector Representation of Shape

In this model, a planar shape is described by a fixed number n of landmarks along its contour, $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, represented as a $2n$-dimensional vector

$$\mathbf{x} = (x_1, ...x_n, y_1, ...y_n)^T. \tag{3.1}$$

Suitable landmarks are essential for an effective shape representation, and landmarks are usually assigned to points of high curvature or biologically meaningful locations. For our application, the landmarks consist of 4 major points corresponding to the nose tip, the tail base, and the two ears of a mouse, and between 2 consecutive major points are a fixed number of equally spaced intermediate points along the boundary (Figure 3-4).

Prior to the statistical analysis, the labeled shapes are transformed them into a common reference frame. The approach is to translate and rotate each shape such that the mean point and orientation between its head and tail landmarks are centered at the origin and aligned to a fixed orientation.

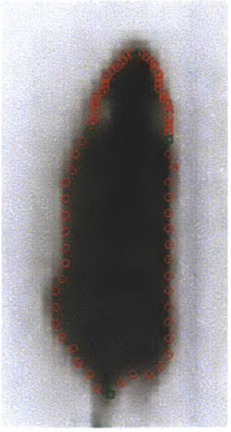


Figure 3-4: An example annotated mouse shape. The 4 major landmarks are marked with green squares, while the remaining points are marked with red circles.

Statistical Analysis of Shapes

Among different instances of mouse shapes, the movement of landmarks are observed to be highly correlated provided that the landmarks are well-chosen. Therefore, we can exploit this inter-point correlation and approximate each shape with fewer parameters while ensuring most variations are still captured.

Principal Component Analysis (PCA) is a common and effective method for dimension reduction. After applying PCA to the training shape data, we can then approximate any mouse shape in vector representation $\mathbf{x}$ as

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}, \tag{3.2}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P} = (\mathbf{p}_1|\mathbf{p}_2|...|\mathbf{p}_m)$ contains the first m eigenvectors of the covariance matrix, and $\mathbf{b}$ is an m-dimensional vector given by

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}). \tag{3.3}$$

We chose m such that these eigenvectors account for most (98%) of the total variations.

The vector $\mathbf{b}$ now defines a deformable shape model with only m parameters. By constraining each parameter b_i within certain range with respect to its variance across the training set λ_i, we ensure that any shape generated by this model is similar to those in the original training set.

The shape model training set consists of 211 manually annotated mouse shapes. $m = 14$

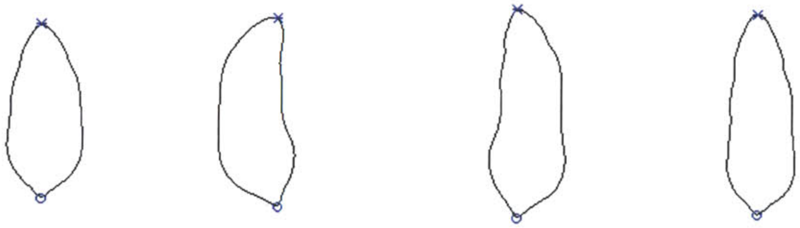


Figure 3-5: From left to right: the mean shape and the first 3 PCA components. The head and tail locations are marked with blue stars and circles.

3.4.2 Shape Tracking

Given an input shape S_i, a target shape S_t, and a dissimiliarity measure, the goal of shape registration is to find a proper transformation such that the dissimilarity between the transformed input shape $\hat{S}_i$ and S_t is minimized. Once the two shapes are correctly aligned, we can locate head and tail locations from the transformation determined.

The approach to be presented here is based on non-rigid registration using distance functions developed by Paragios et al. [14].

Input Shape

The input shape S_i is generated from the statistical shape model developed in Section 3.4.1 with the parameters $\mathbf{b}$:

$$S_i(\mathbf{b}) = \text{Polygon}(\bar{\mathbf{x}} + \mathbf{Pb}). \tag{3.4}$$

The transformations allowed in our problem are translation and rotation. We denote a linear transformation A with rotation by angle θ and translation by a vector $\mathbf{T} = (T_x, T_y)$ as follows

$$A((x,y),\mathbf{T},\theta) = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_x \\ T_y. \end{pmatrix} \tag{3.5}$$

Note that scale transformation is not considered in our application. I assume that the mouse shape being registered is roughly of the same size as those in the training set.

Implicit Representation

If both shapes are represented with landmarks, we can easily register the input shape by minimizing the sum of squared distances between matched points, but this approach fails if some landmarks from the two shapes are not sampled at corresponding locations. As target shapes in our problem are in image domain, their vector representations are not readily available to us.

We can avoid such issue by using the implicit representation, which is essentially the Euclidean distance map of the shape contour. Implicit representation can be easily combined with gradient descent to perform image registration. For a given shape S inside the image domain Ω, its implicit representation is a 2D function $\Phi : \Omega \rightarrow \mathbb{R}$ defined by

$$\Phi((x,y);S) = \begin{cases} 0 & (x,y) \in S \\ +D((x,y);S) > 0 & (x,y) \in R_S \\ -D((x,y);S) < 0 & (x,y) \in \Omega - R_S, \end{cases} \tag{3.6}$$

where R_S is the region enclosed by the shape, and $D((x,y);S)$ refers to the minimum Euclidean distance between the grid location (x,y) and the shape S. Figure 3-6 illustrates the implicity representation of a shape.

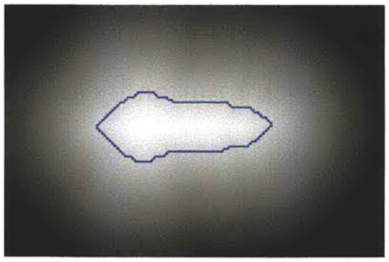


Figure 3-6: The implicit representation of the shape traced in blue.

Energy Minimization

There are 3 main considerations when we register the two shapes: 1) the input shape needs to align up with the target shape, 2) the input shape should confine to the foreground region, and 3) the input shape should look similar to the shapes in the training set. These criteria can be translated into an energy minimization setup.

Given the input shape $S_i(\mathbf{b})$ under transformation A and the target shape S_t, denote their implicit representations as $\Phi_i \equiv \Phi(A((x,y);\mathbf{T},\theta);S_i(\mathbf{b}))$ and $\Phi_t \equiv \Phi((x,y);S_t)$, respectively. Also, denote $R_{\text{bg}} = \Omega - R_{\text{fg}}$ as the background region, which is area not enclosed by the foreground.

Our goal is to determine the parameters T_x, T_y, θ, and $\mathbf{b}$ such that the following energy function is minimized:

$$E = \left(\sum_{\Omega} C\left(\Phi_t - \Phi_i\right) \right) + \alpha_1 \cdot \left(\sum_{R_{\text{bg}} \cap R_i} {\Phi_i}^2 \right) + \alpha_2 \cdot \mathbf{b}^T \mathbf{b}, \tag{3.7}$$

where $C(\cdot)$ is the cost function, and α_1 and α_2 are constant weights. To reduce the impact due to outliers, I chose the cost function as an $L1$-norm approximation $C(x^2) = \sqrt{x^2 + \varepsilon^2}$.

The second term of the right-hand side assigns cost to pixels that spill out of the foreground boundary, and the third term penalizes shapes far away from the original training set, and thus implicitly enforcing the shape constraint.

This minimization problem can be solved using the gradient descent method. Denote $\Delta\Phi \equiv \Phi_t - \Phi_i$. Since A is a linear transformation, the partial derivatives with respect to T_x, T_y and θ can be derived straightforwardly as the following

$$\frac{\partial E}{\partial T_x} = \frac{\partial E}{\partial \Phi_i} \cdot \frac{\partial \Phi_i}{\partial x} \tag{3.8}$$

$$\frac{\partial E}{\partial T_y} = \frac{\partial E}{\partial \Phi_i} \cdot \frac{\partial \Phi_i}{\partial y} \tag{3.9}$$

$$\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial \Phi_i} \cdot \left[(-x \sin\theta + y \cos\theta) \frac{\partial \Phi_i}{\partial x} + (-x \cos\theta - y \sin\theta) \frac{\partial \Phi_i}{\partial y} \right], \tag{3.10}$$

where

$$\frac{\partial E}{\partial \Phi_i} = -\sum_{\Omega} \frac{\Delta \Phi}{\sqrt{\Delta \Phi^2 + \varepsilon^2}} + 2\alpha_1 \cdot \sum_{R_{\text{bg}} \cap R_i} \Phi_i. \tag{3.11}$$

On the other hand, it is easier to approximate the partial derivative with respect to b_i by computing the finite difference.

$$\frac{\partial E}{\partial b_i} = \left(\frac{\partial E}{\partial \Phi_i} \cdot \frac{\Phi(A(p; \mathbf{T}, \theta); S_i(\mathbf{b} + \delta \hat{\mathbf{b_i}})) - \Phi_i}{\delta} \right) + 2\alpha b_i, \tag{3.12}$$

where $\hat{\mathbf{b_i}}$ is the unit vector along the i-th dimension, and δ is a small constant spacing.

Figure 3-7 shows several shape registration results. Observe that as all input shapes are based on the statisitcal model, the aligned results still look similar to mouse shapes from the training set even when the target shapes are highly unsmooth.

As the input shape is inherently in vector representation, the head and tail of each mouse can be located by simply extracting the corresponding landmarks from the vector.

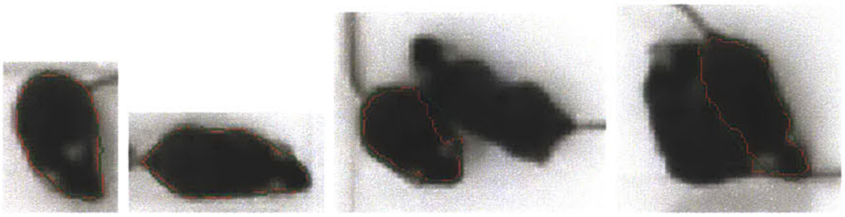


Figure 3-7: Shape registration examples. The target shapes are in red, while the fitted input shapes are in green. Even if the target shapes are nosiy, such as the two examples on the right, the algorithm still gives good results.

Kalman Filtering

For smoother pose estimations over time, the parameters $\mathbf{T}$ and θ from the linear transformation is tracked using the Kalman filter [11]. I use the constant velocity model, in which the state space also maintains the first-order derivative of each parameter. Although mice hardly move with constant velocity, this model provides good shape estimates for shape recovery. Furthermore, as the shape registration is initialized with the predicted state from the Kalman filter, the optimization can converge to a local minimum within fewer steps.

Chapter 4

Performance Evaluation

In this chapter, I provide detailed performance evalution of the tracking system on the dataset I collected. I show that my system can reliably track through occlusions, with significantly low number of identity swaps and accurate pose estimation. I also discuss the primary causes of error I observed from my system.

4.1 Mouse Tracking Data Set

For evaluation, I collected two 30-minute top-view videos of two-mice in one cage. I need to collect my own dataset because my system requires depth maps on top of the image frames, but there exists no such dataset. I tuned the parameters of my system using this dataset prior to the performance assessment. Figure 4-1 shows a frame from the videos.

In both videos, the occlusion frames account for approximately 50% to 60% during the first 15 minutes.

4.1.1 Recording Setup

To collect the videos, the Kinect device was placed approximately 75 cm above the top of a 30cm-by-30cm cage, facing downward.

Illumination is a particularly crucial component in the recording setup. On one

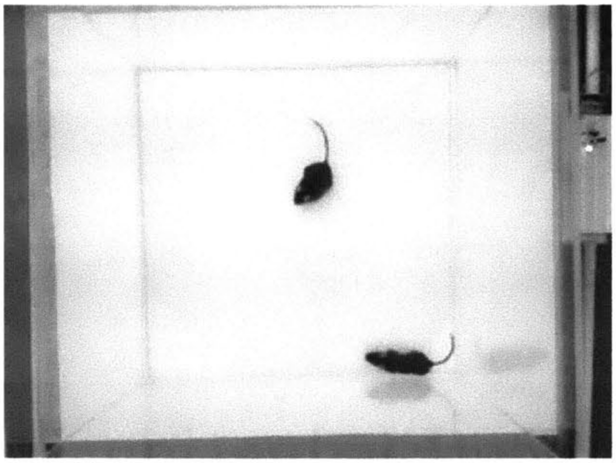


Figure 4-1: An example frame from the mouse tracking data set.

hand, the cage needs to be bright enough for video recording, and on the other hand, it should remain dark to prevent the mice from being anxious. To ensure uniform lighting and minimize impact from shadows, I placed 4 white light bulbs over the 4 corners of the cage, each of which softened by a sheet of diffuser. The bulbs are dimmed such that the white bulbs contributed light intensity below 10 lux. As mouse visual system cannot perceive red, I used red LEDs to bring up the illumination for recording.

The Kinect records with 640x480 resolution at 30 fps. I captured RGB and depth frames from the Kinect device with OpenNI APIs, and used FFmpeg library to write these frames as video files. For each recording, the RGB channel and depth channel are synchronized to frame level.

4.1.2 Animals

For accessing the performance of tracking as well as behavior analysis, I recorded 2 videos, each of which involves two mice in the cage: a wild type mouse and a Shank3 knockout mouse in the first video (referred to as the wild-shank3 video), and the two wild type mice in the second video (referred to as the wild-wild video). In each experiment, I recorded the first 30 minutes of interaction for pairs of mice.

4.1.3 Training the Shape Model

To train the statistical shape model introduced in Section 3.4.1, I randomly sampled 211 non-occlusion frames from the wild-shank3 video as the training set. For each frame, I first manually annotated the 4 major landmarks (head, tail, and two ears) on the contour of an arbitrary mouse, and then programmatically added 18 equally spaced intermediate points along the contour between each two consecutive major landmarks. Thus, each shape consists of 76 points in total, as shown in Figure 4-2.

The first $m = 14$ PCA components of the trained model covers 98% of variance within the training set. However, to save computation, I only used the first 5 PCA components for tracking.

The mean mouse shape spans approximately 60 pixels in length and 30 pixels in width.

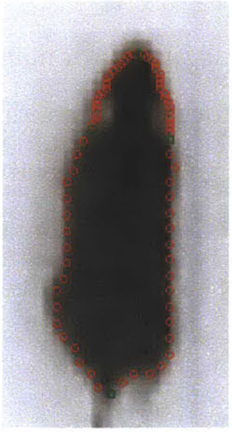


Figure 4-2: An example annotated mouse shape. The 4 major landmarks are marked with green squares, while the remaining points are marked with red circles.

4.2 Evaluating the Tracking System

This section details the evaluation on my tracking system. The evaluation process consists of two parts:

- Measuring how well the system tracks mouse identity
- Measuring how well the system tracks pose

4.2.1 Methods to Compare

To thoroughly assess the performance of my system, I ran 3 versions of algorithm:

- In the first version, I supervised the tracking process of the first 15 minutes (27000 frames) from both videos using my system. Evey time I observed an identity swap or head-tail swap, I corrected the track before resuming the tracking, and logged the correction. The outcome of this tracking is referred to as MY-S.

- The setting for the second version was the same as the previous one, except that this time I did not make any correction throughout the tracking. This version is referred to as MY-NS.

- I ran the third version without the depth maps, in order to determine how much depth information assists tracking. I modified the system such that during over-segmentation stage it used only intensity gradient magnitudes, and during region joining stage it treated all foreground subregions as if they are at the same depth level. Similar to MY-NS, this version received no supervision, and is referred to as MY-NSND.

Here I compare the performance with other mice tracking systems: MiceProfiler by de Chaumont [6], Braun's work [2], and the ellipse tracking system by Edelman [7]. There was no human intervention throughout the tracking for each of them.

I also attempted to compare with the commercial software EthoVision XT by Noldus and SocialScan by CleverSys, but I'm not able to quantify their performance. When tracking multiple mice in one cage, EthoVision is designed to use color markings to distinguish the mouse identities; the system does not maintain a track for every mouse during occlusions. I also observed identity swaps even during non-occlusion frames. CleverSys kindly offered to run their system on my videos, but I did not receive the results in time to include them in this thesis.

4.2.2 Parameter Tuning

To establish fair comparison among different methods, I need to ensure each program operates under its best condition by tuning the parameters as best as I can.

The primary parameters of my system include: foreground segmentation threshold, Watershed algorithm threshold, various weights in region joining, and the step sizes of gradient descent. As these paramters belong to different stages of the system, they were tuned stage-by-stage; I examined the results from each stage to determine the most appropriate parameters of that stage, before moving on to the next one. For example, I would set the Watershed algorithm threshold such that different mice's body do not share the same subregion.

On the other hand, there are relatively fewer parameters in other systems, and they are more straightforward to tune. Two of the most common parameters are the foreground segmentation threshold and the size of mouse.

4.2.3 Accuracy of Identity Tracking

Maintaining correct identity is critical in multiple mice tracking. In this part, I evaluate the rate of identity swaps for each system, using the tracking outcome of MY-S as the ground-truth of identity assignment. Figure 4-3 illustrates the progression of an identity swap.

Identity swaps are quantified by their rate per occlusion event. For each occlusion segment, defined as a consecutive occlusion frame sequence, I compare the identity assignments at the beginning and the end of the segment with the ground-truth. If the assignments are not consistent, then I regard such event as one identity swap.

Table 4.1 summarizes the number of identity swaps from different methods. There are 235 occlusion segments during the first 27000 frames in the wild-wild video, 146 of them in the wild-shank3 video.

Note that identity swaps may occur several times within one occlusion segment and thus cancel out each other, so the numbers presented here are slightly lower than their actual swap rate. This explains why the number of swaps in MY-S is higher

	Wild-wild video (235 occlusions)	Wild-shank3 video (146 occlusions)	Avg. swap rate per occlusion event
MY-S	5	2	1.8%
MY-NS	4	2	1.6%
MY-NSND	2	2	1.0%
Edelman	35	14	12.9%
Braun	19	6	6.6%
de Chaumont	15	9	6.3%

Table 4.1: Number of identity swaps over the first 15 minutes (27000 frames) of each video. The results from my system under supervision (MY-S) represent the number of corrections for each video.

than that in MY-NS.

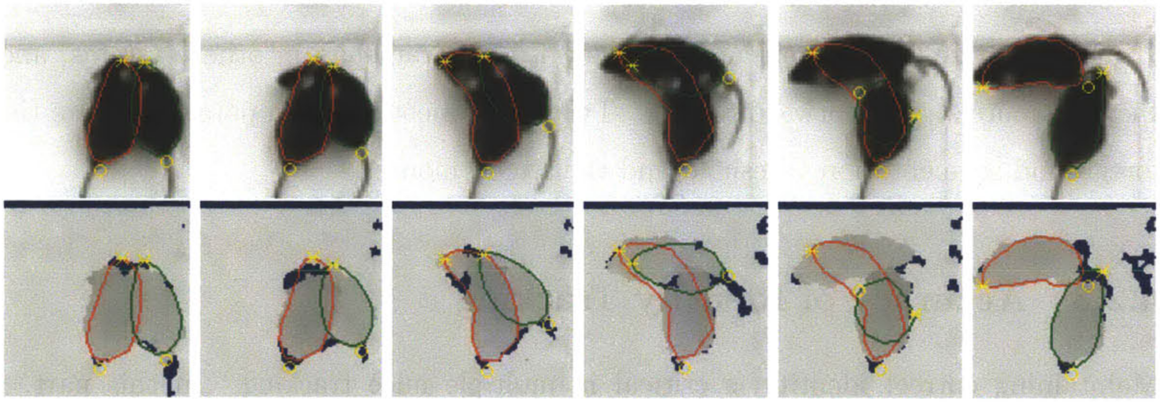


Figure 4-3: This figure illustrates the progression of an identity swap. Note that at the second column, the green mouse shape does not include its actual nose. Due to missing depth values at the nose area, the segmentation module assigns that region to the red mouse, causing the identity swap.

4.2.4 Accuracy of Pose Estimation

In this part, I measure the accuracy of pose estimation from two aspects: the rate of head-tail swaps, and the accuracy of various pose parameters.

- Head-tail swap is a common problem for mouse trackers, and low occurences of such event requires reliable tracking through occlusion. Because a mouse shape is nearly symmetrical, the shape model can still fit well to the observed mouse contour even when the orientation is off by 180 degrees. Figure shows an instance of a head-tail swap process.

- Aside from maintainng correct identity and orientation, a tracker needs to accuracty locate the head/tail locations, and the accuracy of pose parameters directly reflect the quality of pose estimation. Here I approach with 4 representative indicators: nose locations, tail base locations, mouse orientations, and mouse shape. The mouse orientation is defined as direction from the tail base to the nose.

Head-tail Swaps

I define the flip state of a mouse as true if its orientation is off by more than 90 degrees from the ground-truth, and vice versa. A head-tail swap occurs when the flip state of a mouse changes. Figure 4-4 shows an example of a head-tail swap.

I quantify head-tail swaps with two ways: rate per occlusion event, and rate per non-occlusion frame.

An occlusion event triggers a head-tail swap if the flip state before and after the occlusion segment is different, and rate per occlusion event tallies such occurances. Since head-tail swaps often occur as the result of occlusions, rate per occlusion event can reflect the quality of tracking through occlusions.

The rate per non-occlusion frame is the rate of swaps between consecutive non-occlusion frames. This number shows the stability of tracking during non-occlusion frames.

Table 4.2 summarizes the head-tail swap rates for each system. In this assessment, the outcome of MY-S also serves as the ground-truth, and all identity swaps are removed.

Pose Parameters

In order to assess the performance of my tracking system, I need ground-truth for each indicator. Evaluating frame-by-frame performance would be impratctical, so instead, I randomly sampled 1000 frames among the 2 videos. These frames comprises 250

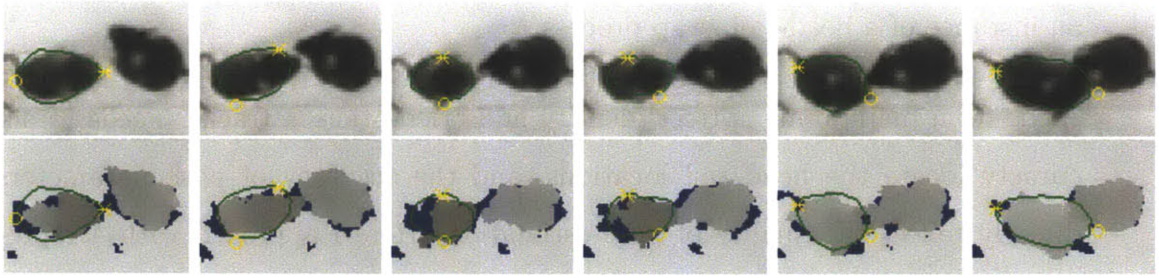


Figure 4-4: This figure shows how mouse rearing can induce a head-tail flip.

	Wild-wild video		Wild-shank3 video		Avg. swap rate per mouse	
	Non-occ	Occ	Non-occ	Occ	Non-occ	Occ
MY-S	2	4	2	5	0.1‰	1.2%
MY-NS	3	5	6	11	0.2‰	2.1%
MY-NSND	2	18	4	19	0.1‰	4.9%
Edelman	298	89	309	60	12.4‰	19.6%
Braun	517	16	728	18	25.4‰	4.5%
de Chaumont	89	82	83	48	3.5‰	17.1%

Table 4.2: Number of head-tail swaps over the first 15 minutes (27000 frames) of each video. The results from my system under supervision (MY-S) represent the number of corrections for each video. All identity swaps are already removed.

occlusion frames and 250 non-occlusion frames from each of the two videos. These human labels are regarded as ground-truth for evaluating accuracy.

With the help of Amazon's Mechanical Turk, I collected human annotations of these frames from online workers. The set of frames are divided into many small labeling tasks, each of which consists of 5 or 10 frames. The online workers are instructed to label the nose locations, the tail base locations, and trace the contours of the presented mice. They had to label based on the true locations and shapes even when some parts are hidden.

To ensure the quality of the labels, the annotation process took several rounds. At the end of each round, I selected frames with defective labels and posted a new task to re-label them.

As validation of the quality of these human annotations, I randomly selected and labeled 40 occlusion and 40 non-occlusion frames from the original sampled set, and compare my labels with labels from online workers. This comparison is established

Non-occlusion:

	Nose (pixel)	Tail (pixel)	Orientation (degree)	Contour
Baseline	1.22	3.17	2.89	0.91 ± 0.02
MY-S	3.76	7.97	8.39	0.85 ± 0.04
MY-NS	5.06	6.86	8.13	0.84 ± 0.05
MY-NSND	5.85	6.36	9.11	0.82 ± 0.05
Braun	4.35	6.03	7.38	0.86 ± 0.04
Edelman	10.37	11.03	11.17	0.85 ± 0.10
de Chaumont	20.76	12.88	23.29	N/A*

Occlusion:

	Nose (pixel)	Tail (pixel)	Orientation (degree)	Contour
Baseline	3.95	4.95	4.22	0.89 ± 0.05
MY-S	7.53	8.07	9.81	0.79 ± 0.09
MY-NS	7.93	7.71	10.41	0.78 ± 0.10
MY-NSND	11.72	9.12	15.93	0.74 ± 0.12
Braun	15.57	9.91	18.80	0.75 ± 0.15
Edelman	20.00	16.19	25.73	0.61 ± 0.18
de Chaumont	26.99	17.16	41.15	N/A*

Table 4.3: Accuracy of the tracking system. These numbers are computed with all head-tail swaps and identity swaps removed. On average the mouse in the videos is 60 pixels long and 30 pixels wide. (* The tracking outputs of de Chaumont's system do not include shapes.)

as the baseline, reflecting how difficult these frames are for human.

Table 4.3 summarizes the evaluation outcomes. As the performance measure, I compute the root mean square error with respect to the ground-truth for nose/tail locations and orientations, and I compute the Jaccard coefficient (intersection-over-union) for the mouse shape. Appendix A contains the cumulative distribution of deviation from the ground-truth for each indicator.

Here I assume the main failure modes are identity swaps and head-tail swaps, and in order to exclude their impact on performance, for each frame I choose the identity and orientation assignment that gives the best result. Therefore, the numbers here reflects how well the tracking is in the absense of these failures.

4.3 Discussion

My tracking system outperforms all other methods in every aspect of performance assessment, especially during occlusion frames. In this section I summarizes the evaluation results, then discuss in-depth about cases that make the system fail, and finally explain any modification required to track more than 2 mice.

4.3.1 Summary of Evaluation

My system has low rate of identity swaps. Without any marking on the mouse, the tracking system presented in this thesis can maintain mouse identites very well. The identity swap rate is 1.6% per occlusion event (MY-NS), while the rates of other systems are at least 4 times higher. This allows minimal human intervention for correcting identity errors.

My system achieves accurate pose estimation. The head-tail swap rate of the system is 2.1% per mouse per occlusion event (MY-NS). After removing identity and head-tail swaps, it can locate the nose and the tail base of a mouse within 8 pixels from their ground-truth for 80% of the occlusion frames, while the human agreement is around 3 pixels. On the other hand, Braun's approach, which yields the best post estimation results among other methods I compare against, can only locate the head within 17 pixels at the same percentage of occlusion frames.

Depth modality improves tracking through occlusions. The performance of MY-NS is higher than MY-NSND in terms of pose estimation during occlusion frames. This proves that depth modality can indeed improve tracking through occlusions. Although MY-NSND reports fewer identity swaps than MY-NS, this comparison is less statistically meaningful due to the low swap rate.

4.3.2 Failure Cases

Here I discuss some reasons that result in unreliable tracking.

The depth map from Kinect is noisy and does not perfectly align to the color image, as illustrated in Figure 4-5. The system can fail to segment the two mice when the depth values around their boundary are missing. The identity swap in Figure 4-3 happens partly because of the missing depth values.

The system is more prone to identity or head-tail swaps when a mouse is rearing during an occlusion frame. Because the body of a rearing mouse spans across a large depth range, sometimes during region joining the system assigns part of its body to another mouse. In addition, when rearing is in action, the mouse shape resembles a circle, and thus the mouse can easily rotates to the opposite orientation. Figure 4-4 shows an instance of such head-tail swap.

In addition, as Kinect is designed to operate at a longer range: from 80 cm to several meters, the image and depth resolutions are not high, making it more difficult to identify the boundary between the mice when they are roughly at the same height. With a better depth-sensing device, the tracking system can potentially perform better.

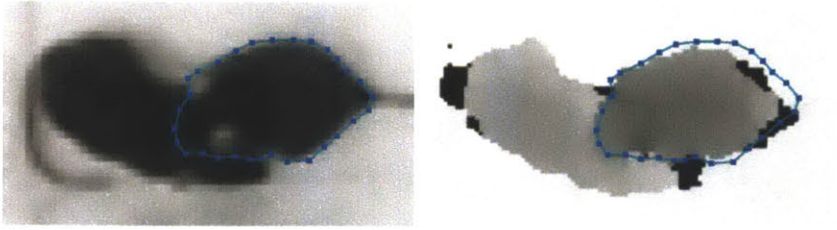


Figure 4-5: This figure shows that the grayscaled image and depth map do not align perfectly despite they correspond to the same frame. The contour of the mouse is traced in blue.

4.3.3 Extension to More Than Two Mice

Although the performance evaluation here is based on two mice in one cage, it is possible to extend the system to $m > 2$ mice. With m mice, the region joining module should maintain m identities instead. Also, during occluded shape recovery, the overlapping cases among all mice have to be enumerated. Although this number grows exponentially, if we assume most occlusion frames involves only 2 or 3 mice, we may still obtain reasonable recovered shapes.

Chapter 5

Social Behavior Analysis

In this chapter, I explore automating social behavior analysis using tracking data produced from my tracking system. I first generate simple position- and orientation-based features, and then define behavior rules based on them. Finally, I compare the statistics of these behaviors from the two videos in my dataset.

The purpose of this chapter is to demonstrate quantification of behaviors using the tracking results. Note that the results presented in this chapter are preliminary, and more careful validations are required.

5.1 Feature Representation

First of all, I convert the tracking data into a set of frame-by-frame features that I will use to define behaviors.

Table 5.1 details the feature set. In the table, I denote N_M as the position of mouse M's nose, T_M as the position of its tail base, C_M as the position of its body center, and S_M as its shape. These features mainly consists of distances and relative orientations of body parts, and can be directly computed with the results from the tracking system. In this thesis, the features are based on two mice in one cage, referred to as mouse A and mouse B. However, the list can easily be extended to represent more than 2 mice.

As the features should evolve smoothly in time, I apply a temporal Gaussian filter

Feature Expression	Feature Description
$f_1 = \lvert\overrightarrow{N_A N_B}\rvert$	distance between noses
$f_2 = \lvert\overrightarrow{T_A T_B}\rvert$	distance between tail-bases
$f_3 = \lvert\overrightarrow{N_A T_B}\rvert$	distance between A's nose and B's tail base
$f_4 = \lvert\overrightarrow{T_A N_B}\rvert$	distance between A's tail base and B's nose
$f_5 = \lvert\overrightarrow{C_A C_B}\rvert$	distance between the body centers
$f_6 = \angle\left(\overrightarrow{N_A C_B}, \overrightarrow{T_A N_A}\right)$	direction of B's body relative to A's orientation
$f_7 = \angle\left(\overrightarrow{N_B C_A}, \overrightarrow{T_B N_B}\right)$	direction of A's body relative to B's orientation
$f_8 = \min\left(S_A, S_B\right)$	minimum distance between the bodies of A and B
$f_9 = \lvert S_A \cap S_B\rvert$	area of the overlapping region between the bodies of A and B
$f_{10} = \left\lvert\overrightarrow{C_A C_A^{(-)}}\right\rvert$	change in A's body center between adjacent frames
$f_{11} = \left\lvert\overrightarrow{C_B C_B^{(-)}}\right\rvert$	change in B's body center between adjacent frames
$f_{12} = \left\lvert\overrightarrow{C_A C_B}\right\rvert - \left\lvert\overrightarrow{C_A^{(-)} C_B^{(-)}}\right\rvert$	change of A and B's distance between adjacent frames
$f_{13} \in \{A, B, 0\}$	the mouse at bottom during occlusion

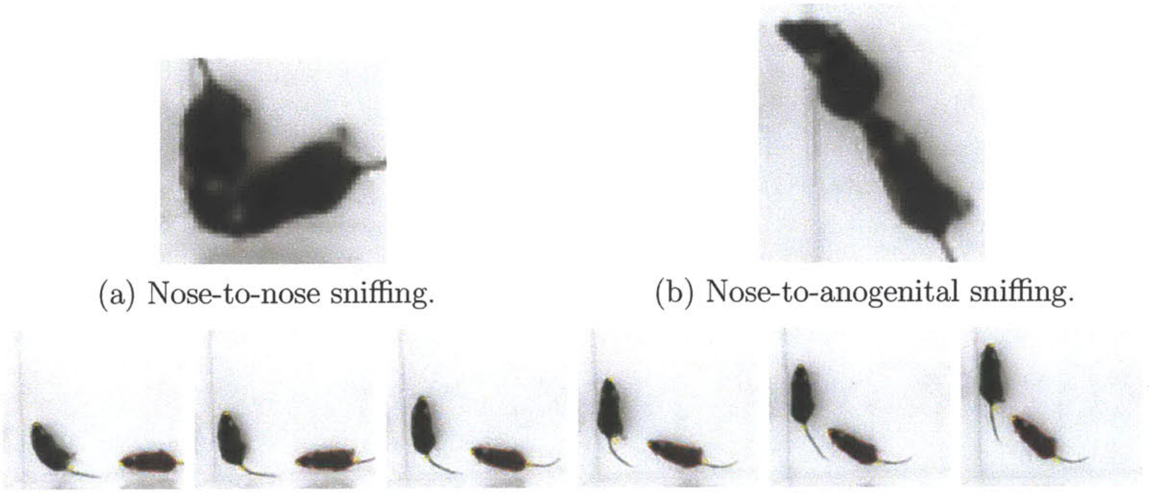
Table 5.1: The position- and orientation-based features in my behavior analysis.

on them to eliminate any abrupt change due to tracking artifacts.

5.2 Social Interactions

Based on the features introduced in the last section, I further define a set of rules for detecting behaviors. Table 5.2 lists the repertoire of social interactions that I use in my analysis, which are inspired by de Chaumont's work [6]. The system can detect social interactions commonly used in behavioral study, such as nose-to-nose sniffing or follow behaviors.

Unfortunately, due to time constraint, I'm not able to carefully validate the definition of these rules. The validation should involve comparing the detected social behaviors to annotations labeled by experts.



(a) Nose-to-nose sniffing.

(b) Nose-to-anogenital sniffing.

(c) Follow.

Figure 5-1: Sample images of 3 behaviors: nose-to-nose sniffing, nose-to-anogenital sniffing, and follow.

5.3 Results of Social Behavior Analysis

In this section, I demonstrate using the set of behavior rules to quantify the interactions between two mice on my dataset introduced in Chapter 4 the wild-wild video and the wild-shank3 video. Each video records the social reciprocal experiment of a pair of unfamiliar mice. The wild-wild video involves a pair of wild-type mice, while the wild-shank3 video involves a wild type mouse and a Shank3-knockout mouse. I analyzed on the first 5 minutes (9000 frames) of each video.

The steps of the analysis are as follows. I first ran my tracker to obtain the tracking data, which include the head/tail locations and the shape of each tracked mouse for every frame. Then I generated the corresponding positional and orientational features, and detected social interactions based on the defined rules. Finally I computed the statistics of the behaviors, such as number of frames the animals engage in a particular interaction.

The Shank3 mouse was reported to have abnormal social patterns. More specifically, during a social reciprocal test, a wild-type-Shank3 pair displays lower frequency of nose-to-nose interaction and anogenital sniffing than a wild-type-wild-type pair [15].

For a stronger outcome, multiple videos are required for each pair of strains, and

Event	Description
Contact	$f_8 \leq 3$
Small distance	$f_8 \leq 10$
Nose-to-nose contact	$f_1 \leq 5$
Nose (A)-to-anogenital (B) contact	$f_3 \leq 5$
Nose (B)-to-anogenital (A) contact	$f_4 \leq 5$
Side-by-side (same way)	$f_1 \leq 20 \wedge f_2 \leq 20$
Side-by-side (opposite way)	$f_3 \leq 20 \wedge f_4 \leq 20$
A mounts on B	$f_9 \geq 120 \wedge f_{13} = B$
B mounts on A	$f_9 \geq 120 \wedge f_{13} = A$
A is behind B	$f_7 \geq 120$
B is behind A	$f_6 \geq 120$
A approaches B	$(f_{10} \geq 1) \wedge (f_{10} > f_{11}) \wedge (f_{12} < 0) \wedge (f_6 < 45)$
B approaches A	$(f_{11} \geq 1) \wedge (f_{11} > f_{10}) \wedge (f_{12} < 0) \wedge (f_7 < 45)$
A follows B	$(f_{10} \geq 1) \wedge (f_{11} \geq 1) \wedge (f_7 \geq 120) \wedge (f_6 < 45)$
B follows A	$(f_{10} \geq 1) \wedge (f_{11} \geq 1) \wedge (f_6 \geq 120) \wedge (f_7 < 45)$

Table 5.2: Repertoire of social interaction.

Behavior	wild-shank3 video	wild-wild video
Nose-to-nose sniffing (# frames)	113	354
Nose-to-anogenital sniffing (# frames)	19	13
Follow behavior (times)	6	15

Table 5.3: Number of frames spent in different behaviors for each pair of animals.

as I have only two videos, the results presented here are preliminary.

Table 5.3 lists the quantified results of nose-to-nose sniffing, anogenital sniffing, and follow behavior by the target animal (wildtype or Shank3) towards a stimulus animal (wildtype). Notice that the duration of nose-to-nose sniffing between the wild-shank3 pair is only one-third of that between the wild-wild pair. In addition, the Shank3 mouse follows the other mouse fewer times than its wild-type counterpart. However, the distinction in the anogenital sniffing behavior is little.

The outcome above is still inconclusive, but it suggests a way forward for analyzing social behaviors.

Chapter 6

Conclusion and Future Work

In this chapter, I summarize the contributions of this thesis, and then propose future research directions to improve the system performance.

6.1 Conclusions

This thesis consists of 3 main contributions:

Reliable tracker for multiple mice with identical appearance. The proposed system combines intensity and depth information from a depth sensing camera to identity the boundary between two occluding mice, and it can accurately track the head and tail locations of mice. The proposed tracking system outperforms other existing systems in terms of identity swap rates and pose estimation accuracies, especially through occlusion frames. Detailed performance evaluation shows that the average identity swap rate is 1.6% (per occlusion event), and the average head-tail swap rate is 2.1% (per mouse per occlusion). Excluding identity and head-tail swaps, the system can locate the nose and the tail base of a mouse within 8 pixels for 80% of the occlusion frames, while the human agreement is around 3 pixels. (Here a typical mouse shape is 60 pixels in length and 30 pixels in width.)

Multiple mice tracking dataset and annotations. The dataset consists of two

30-minute videos recorded with Microsoft Kinect device, each of which contains synchronized color frames and depth frames. The two videos record the social reciprocal experiments of 2 mouse pairs: wild-type-wild-type pair, and wild-type-Shank3-knockout pair. I also collected 1000 randomly sampled frames from the two videos are labeled with mouse nose/tail locations and contours.

Preliminary social behavior analysis results. I define 13 position- and orientation-based features computable from tracking results, and then detect 15 different interactions by thresholding these features. The set of interacations include common behaviors, such as nose-to-nose sniffing, nose-to-anogenital sniffing. I demonstrated that the wild-type-Shank3 pair of mice spent only one third number of frames in nose-to-nose sniffing compared with the wild-type-wild-type pair.

6.2 Future Work

Tracking multiple mice in one cage is a challenging task, and continuous effort is required to achieve the ultimate goal of automatically phenotying mouse social behavior. I hope this thesis can drive work on automated behavior analysis in the research community. There is still much work to do towards a more reliable mice tracking system, and here I propose some directions for expanding the current work.

Training a 3D pose model. Currently the depth map is only used for segmentation, but if we take advantage of depth modality in pose estimation, we can build a 3D pose model that potentially better reflects behaviors like rearing. Combining the top-view and multiple side-views might also help us acheive more accurate pose estimation.

Detecting ear or tail for more reliable pose estimation. The current tracking system relies heavily on previous frames to initialize shape registration. Without any feature point to anchor the shape, errors can accumulate from previous frames and the next. In the top view, the color of mouse ears is distinct, and by locating the ears we may reduce the head-tail swap rate. Detecting mouse tails is also a potential approach. Since ears and tails are not always visible from the view, we need a method that integrates the noisy ear/tail observations to shape registration.

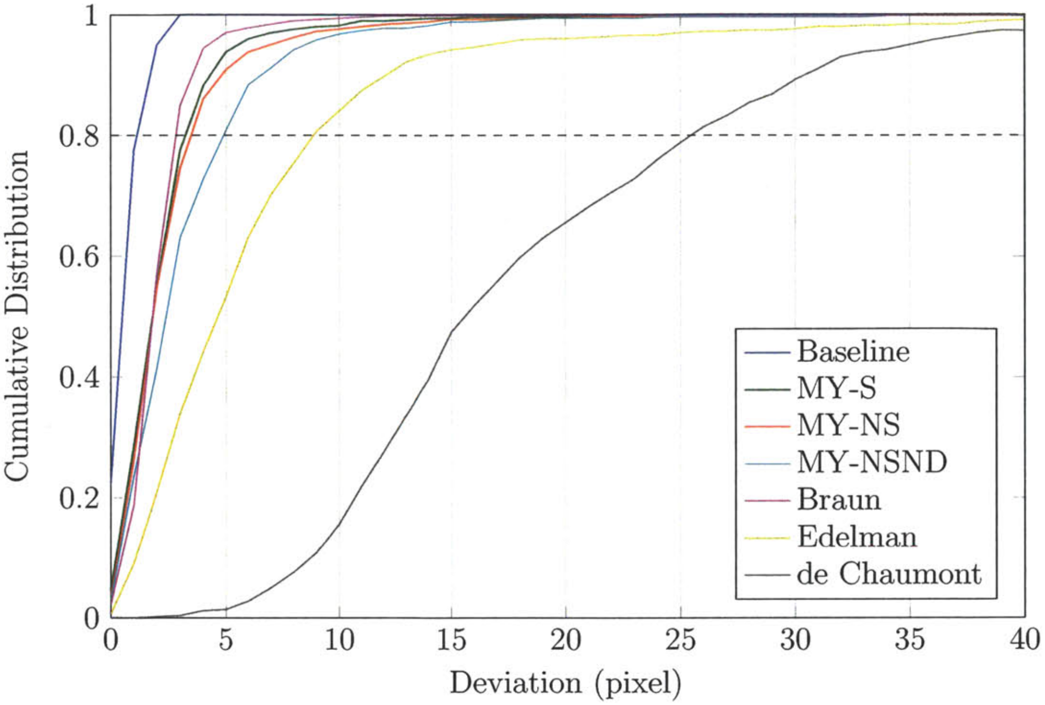
Extending the behavior analysis automation. In this thesis, I prioritized the tracking system over the behavior analysis system. With a

Identifying fine behaviors. Grooming is a fine-grained yet important behavior in mouse behavior analysis. To recognize the subtle movements of grooming, we need more than the position-based features in my current system. A potential approach is to combine Jhuang's motion-based features inspired by visual cortex model [10], or the spatial-temporal features in Burgos-Artizzu's work [3].

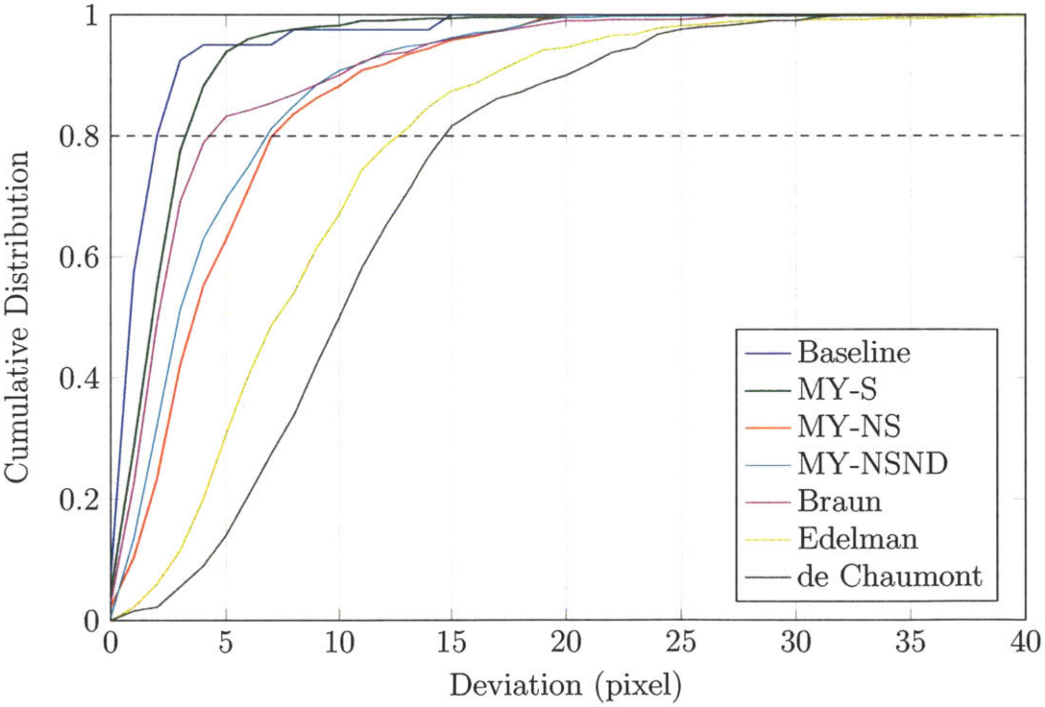
Appendix A

Detailed Results of Accuracy Evaluation

This appendix contains the detailed results of evaluation on pose estimation accuracy from Chapter 4. There are 8 plots in total; the first 4 of them are the results for non-occlusion frames, and the other 4 for occlusion frames. These plots show the cumulative distribution of each pose parameter in the following: deviation of nose location, deviation of tail location, deviation of mouse orientation, and the Jaccard index of mouse shape.

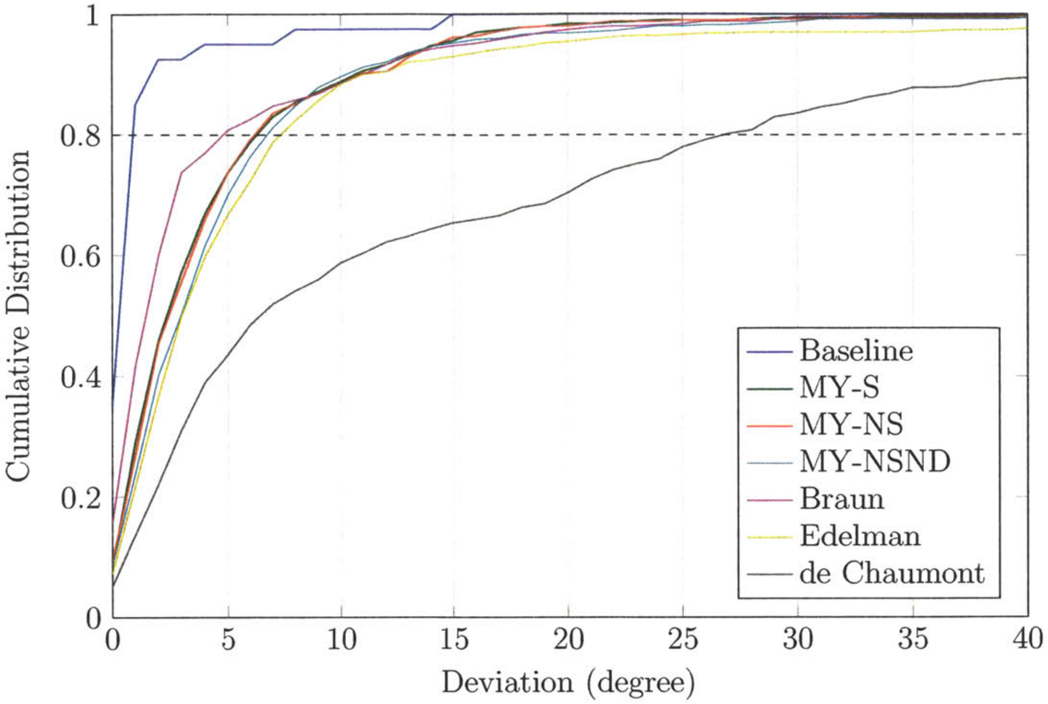


(a) Empirical CDF of nose location deviation from ground-truth.

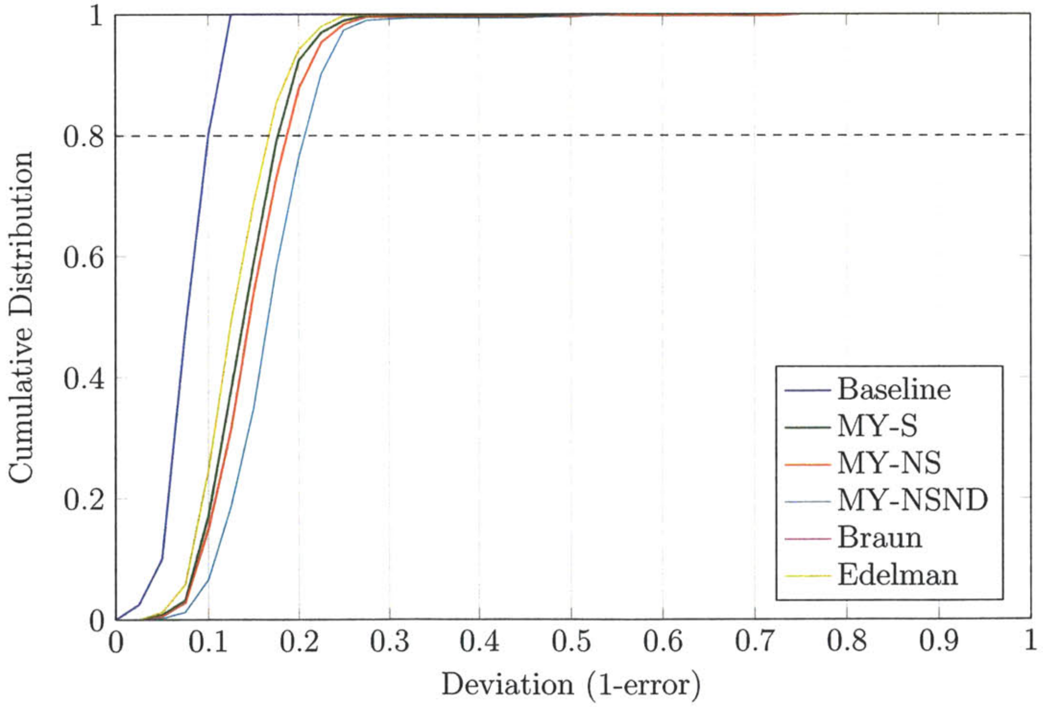


(b) Empirical CDF of tail location deviation from ground-truth.

Figure A-1: The results for **non-occlusion frames**.

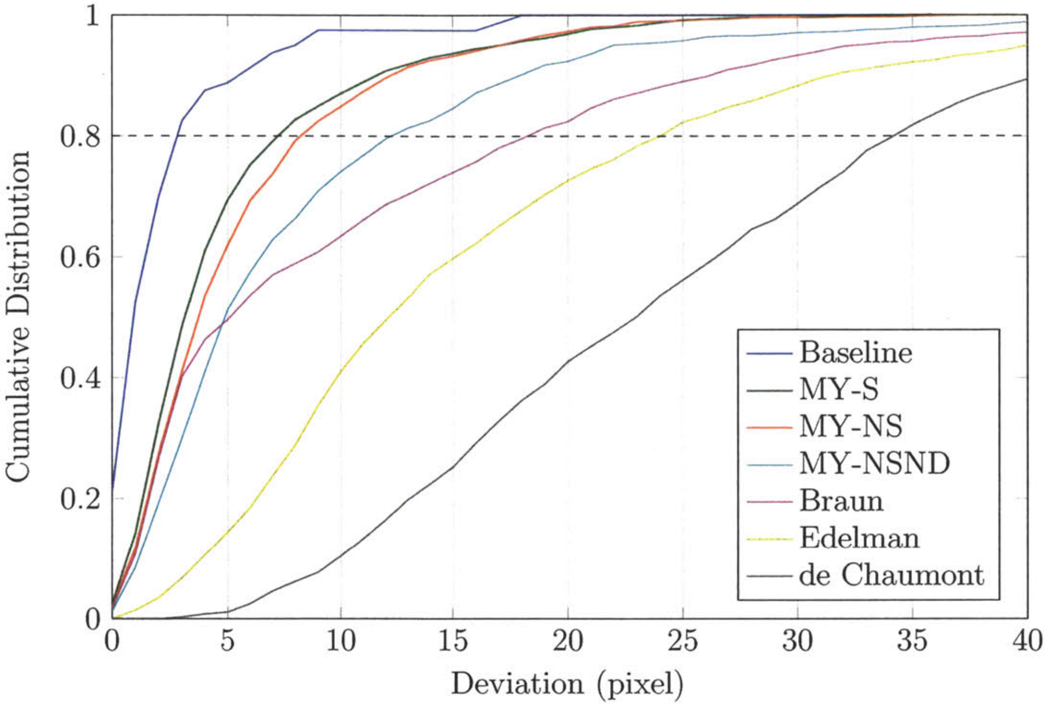


(c) Empirical CDF of mouse orientation deviation from ground-truth.

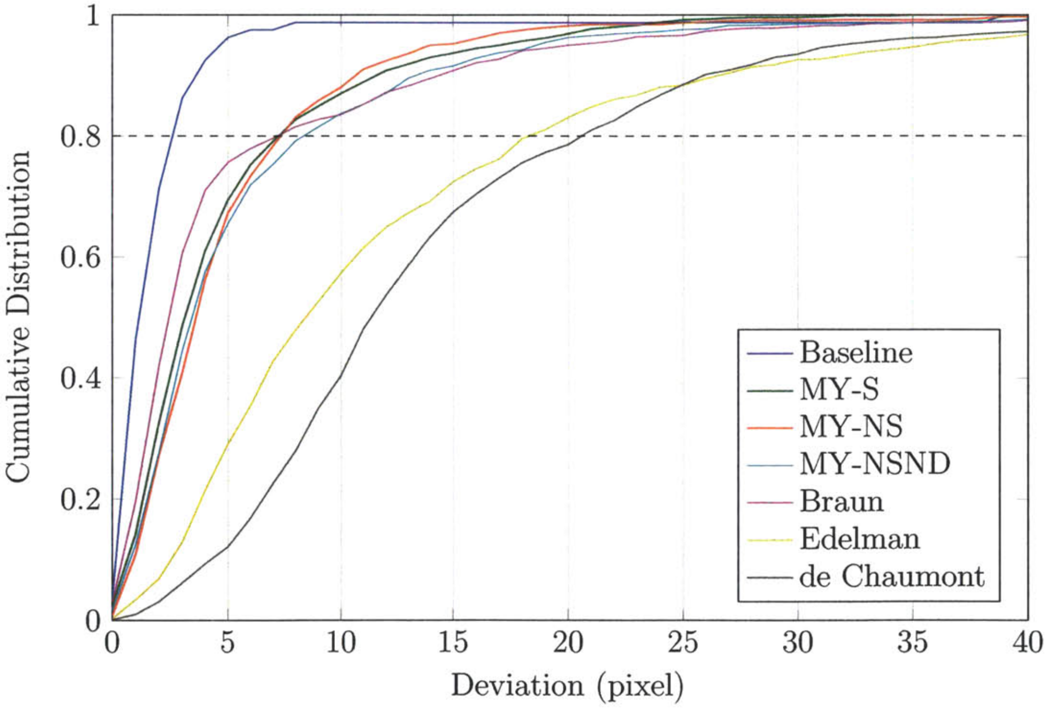


(d) Empirical CDF of Jaccard distance of the mouse shape to the ground-truth.

Figure A-1: The results for **non-occlusion frames**.

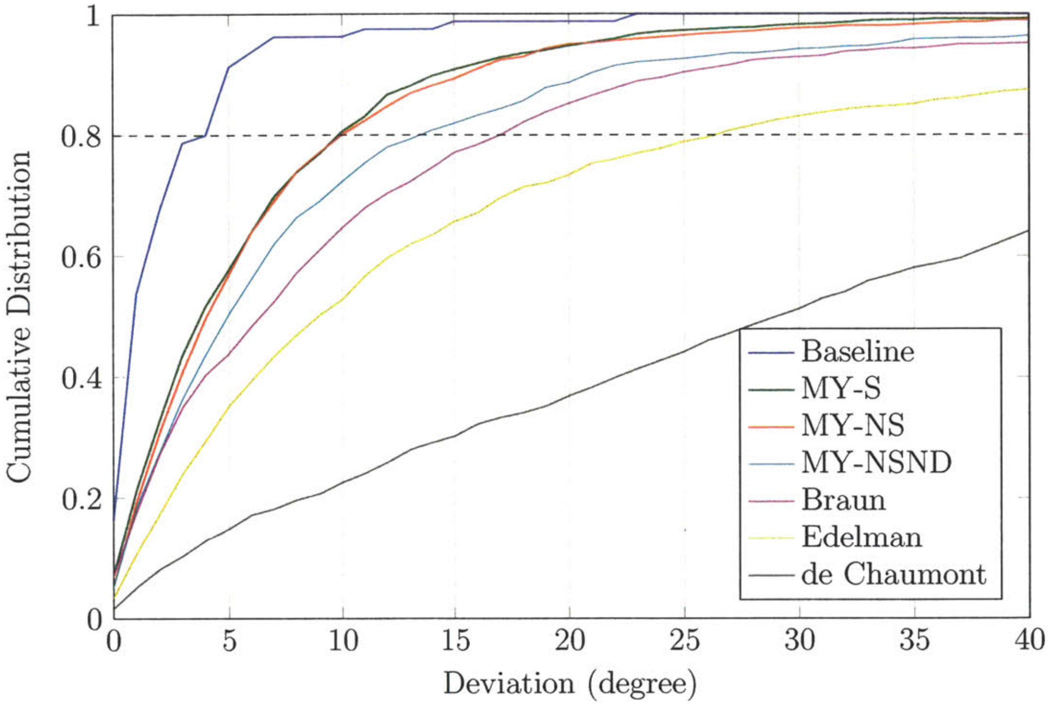


(a) Empirical CDF of nose location deviation from ground-truth.

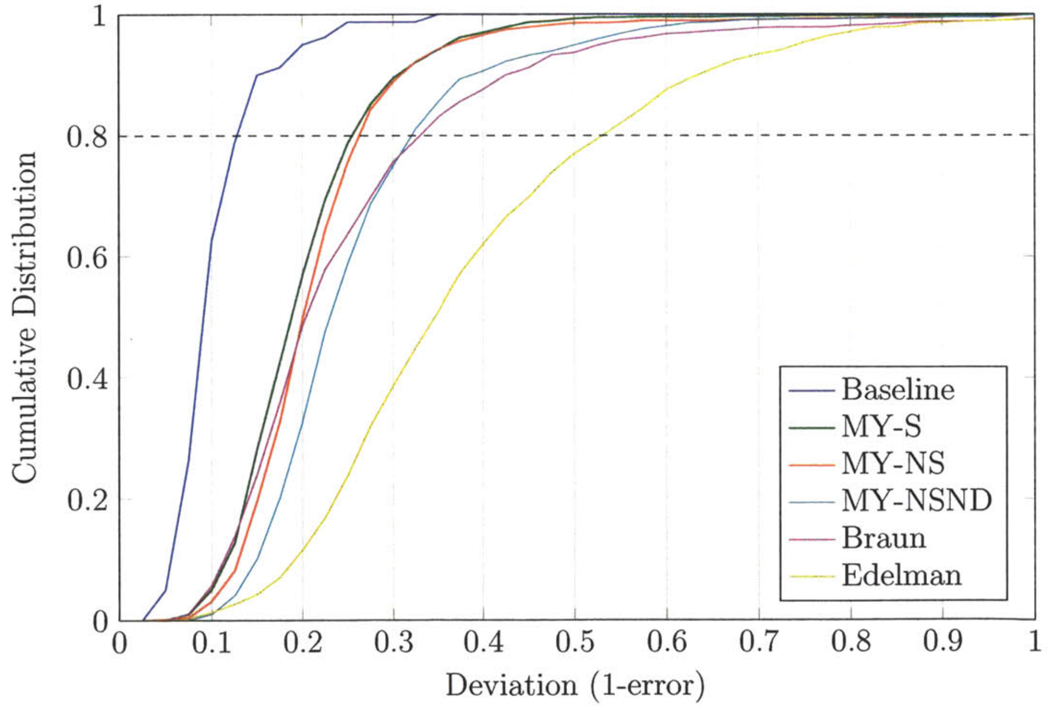


(b) Empirical CDF of tail location deviation from ground-truth.

Figure A-2: The results for **occlusion frames**.



(c) Empirical CDF of mouse orientation deviation from ground-truth.



(d) Empirical CDF of Jaccard distance of the mouse shape to the ground-truth.

Figure A-2: The results for **occlusion frames**.

Bibliography

[1] Kristin Branson and Serge Belongie. Tracking multiple mouse contours (without too many samples). In *Computer Vision and Pattern Recognition (CVPR)*, pages 1039–1046, 2005.

[2] Stav Braun. Tracking multiple mice. Master's thesis, MIT, 2012.

[3] Xavier P. Burgos-Artizzu, Piotr Dollar, Dayu Lin, David J. Anderson, and Pietro Perona. Social behavior recognition in continuous video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1322–1329, 2012.

[4] Timothy F Cootes, Christopher J Taylor, David H Cooper, Jim Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[5] Jacqueline N. Crawley. Mouse behavioral assays relevant to the symptoms of autism*. *Brain Pathology*, 17(4):448–459, 2007.

[6] Fabrice de Chaumont, Renata Dos-Santos Coura, Pierre Serreau, Arnaud Cressant, Jonathan Chabout, Sylvie Granon, and Jean-Christophe Olivo-Marin. Computerized video analysis of social interactions in mice. *Nature Methods*, 9:410–417, March 2012.

[7] Nicholas Edelman. Automated phenotyping of mouse social behavior. Master's thesis, MIT, 2011.

[8] Rana Farah, J.M. Pierre Langlois, and Guillaume-Alexandre Bilodeau. Catching a rat by its edglets. *IEEE Transactions on Image Processing*, 2011.

[9] W.N. Goncalves, J.B.O. Monteiro, J. de Andrade Silva, B.B. Machado, H. Pistori, and V. Odakura. Multiple mice tracking using a combination of particle filter and k-means. In *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on*, pages 173 –178, oct. 2007.

[10] Hueihan Jhuang, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D. Steele, and Thomas Serre. Automated home-cage behavioural phenotyping of mice. *Nature communications*, 2010.

[11] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[12] Joo P. Monteiro, Hlder P. Oliveira, Paulo Aguiar, and Jaime S. Cardoso. Depth-map images for automatic mice behavior recognition.

[13] Tai-Hsien Ou-Yang, Meng-Li Tsai, Chen-Tung Yen, and Ta-Te Lin. An infrared range camera-based approach for three-dimensional locomotion tracking and pose reconstruction in a rodent. *Journal of Neuroscience Methods*, 201(1):116 – 123, 2011.

[14] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Non-rigid registration using distance functions. *Computer Vision and Image Understanding*, 89(23):142 – 165, 2003. ¡ce:title¿Nonrigid Image Registration¡/ce:title¿.

[15] Joo Pea, Ctia Feliciano, Jonathan T. Ting, Wenting Wang, Michael F. Wells, Talaignair N. Venkatraman, Christopher D. Lascola, Zhanyan Fu, and Guoping Feng. Shank3 mutant mice display autistic-like behaviours and striatal dysfunction. *Nature*, 472(7344):437–442, 2011.

[16] Hemerson Pistori, Valguima Victoria Viana Aguiar Odakura, Joo Bosco Oliveira Monteiro, Wesley Nunes Gonalves, Antonia Railda Roel, Jonathan de Andrade Silva, and Bruno Brandoli Machado. Mice and larvae tracking using a particle filter with an auto-adjustable observation model. *Pattern Recognition Letters*, 31(4):337 – 346, 2010.

[17] Laurence H Tecott and Eric J Nestler. Neurobehavioral assessment in the information age. *Nature Neuroscience*, 7(5):462–466, 2004.

[18] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:583–598, 1991.