Scalable Vision System for Mouse Homecage Ethology

Ghadi Salem[1(✉)], Jonathan Krynitsky[1], Brett Kirkland[1], Eugene Lin[1], Aaron Chan[1], Simeon Anfinrud[1], Sarah Anderson[1], Marcial Garmendia-Cedillos[1], Rhamy Belayachi[1], Juan Alonso-Cruz[1], Joshua Yu[1], Anthony Iano-Fletcher[1], George Dold[1], Tom Talbot[1], Alexxai V. Kravitz[1], James B. Mitchell[1], Guanhang Wu[2], John U. Dennis[2], Monson Hayes[3], Kristin Branson[4], and Thomas Pohida[1]

[1] National Institutes of Health, Bethesda, MD, USA
ghadi.salem@nih.gov
[2] Food and Drug Administration, Silver Spring, MD, USA
[3] George Mason University, Fairfax, VA, USA
[4] Howard Hughes Medical Institute (JFRC), Ashburn, VA, USA

Abstract. In recent years, researchers and laboratory support companies have recognized the utility of automated profiling of laboratory mouse activity and behavior in the home-cage. Video-based systems have emerged as a viable solution for non-invasive mouse monitoring. Wider use of vision systems for ethology studies requires the development of scalable hardware seamlessly integrated with vivarium ventilated racks. Compact hardware combined with automated video analysis would greatly impact animal science and animal-based research. Automated vision systems, free of bias and intensive labor, can accurately assess rodent activity (e.g., well-being) and behavior 24-7 during research studies within primary home-cages. Scalable compact hardware designs impose constraints, such as use of fisheye lenses, placing greater burden (e.g., distorted image) on downstream video analysis algorithms. We present novel methods for analysis of video acquired through such specialized hardware. Our algorithms estimate the 3D pose of mouse from monocular images. We present a thorough examination of the algorithm training parameters' influence on system accuracy. Overall, the methods presented offer novel approaches for accurate activity and behavior estimation practical for large-scale use of vision systems in animal facilities.

1 Introduction

The application of vision systems' technologies could have a huge impact on animal-based medical research, including corresponding animal care. During recent decades, the use of laboratory mice in biomedical research increased considerably [1]. Laboratory animals including mice are used to gain new knowledge for improving the health and well-being of both humans and other animals [2].

J. Blanc-Talon et al. (Eds.): ACIVS 2016, LNCS 10016, pp. 626–637, 2016.
DOI: 10.1007/978-3-319-48680-2_55

However, mice are the most frequently characterized and used mammals in biomedical research because of their small size and ease of use, including relative ease for sophisticated genetic manipulation [1]. To achieve high-density housing while maintaining consistent, controlled microenvironments, animal facility managers frequently utilize individually ventilated cages that mate with specialized racks. These ventilated cage environments have become the standard in laboratory facilities as they provide protection for personnel (e.g., infectious agent and allergen containment) and maintain low levels of ammonia and CO_2 allowing an increased number of cages in animal holding rooms. When there are hundreds or thousands of cages in one institution, monitoring of animal health and activity is infrequent, of limited measures, and rather subjective. The use of vision systems could significantly reduce the workload and more appropriately focus the efforts of trained animal care staff by providing continual automated monitoring. This would increase efficiency and reduce bias (e.g., due to fatigue or drift [3]). For example, abnormalities in behavior patterns can be automatically identified leading to early detection of illness, which can be quickly treated or managed. The activity measures are of use to researchers conducting phenotyping, drug-efficacy, and animal model characterization studies. While many commercial and academic systems have been developed to automate home-cage ethology, the wide use of vision systems is contingent on availability of minimal footprint hardware with seamless integration in ventilated racks. Salem et al. [4] reported on the first video-based hardware design specifically targeted for use in cage-racks. This system integrates into the ventilated rack without modification to the cages or racks, nor alteration to animal husbandry procedures. The resulting video poses processing challenges as mouse appearance exhibits large variations induced by the nonlinearity of fisheye lenses, which is exacerbated by lens placement in very close proximity to the cage. The position estimation presented by the authors is limited to predicting the mouse 2D physical centroid projected to the cage-floor, and only in cases when the mouse has all its limbs on the cage-floor.

In this work, we begin addressing the task of mouse pose estimation using the challenging video output from the system described in [4]. We present a novel approach to producing accurate 3D pose estimates from monocular images. The approach utilizes a rich dataset with mouse posterior/anterior annotations from two orthogonal views. We describe slight modifications to the hardware system that enable gathering of a unique training set. We investigate estimation accuracy as a function of training parameters. The prototype example output images and pose estimation results are shown in Fig. 1. The datasets are made publicly available (scorhe.nih.gov) to encourage further development in field.

2 Related Work

Automated analysis of mice activity and behavior has attracted commercial and academic interest over the past two decades [3,5,6]. Desired analysis output measures range from pose estimation to detection of predefined behaviors [6–8].

We limit our review to video-based systems for mice home-cage monitoring. We explore hardware systems as well as pose estimation methods.

2.1 Hardware Systems

Hardware systems employed in academic works are typically simple prototypes and ad hoc setups [6] that use cameras fitted with standard lenses, positioned a sufficient distance from the cage ensuring the field-of-view encompasses the cage volume. Some setups rely on overhead cameras [8,9]. However, as noted by many [7,10], such placement is not well-suited for scalability due to cage and rack obstructions in high-density housing. Commercial hardware systems are reviewed in [4].

2.2 Pose Estimation Methods

One sought output of automated video analysis is a per-frame pose estimate, which can be subsequently used for motion analysis. Two defining components of pose estimation are the pose model (i.e., pose parameters) and pose detection method. For pose model, ellipses are used by [11–13]. Oriented ellipses (i.e.,

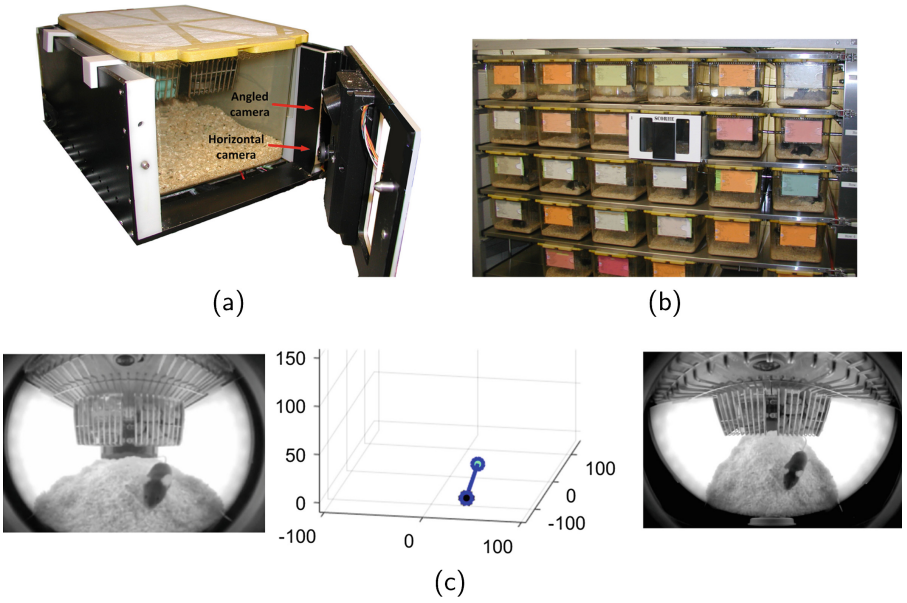

Fig. 1. (a) The video acquisition system used for this work. (b) The system is designed to seamlessly integrate into vivarium cage racks. (c) Example images from the horizontal (leftmost image) and angled (rightmost image) cameras acquired at the same time instance, along with the corresponding 3D position (in mm) of posterior (lighter circle) and anterior (darker circle) estimated by our algorithm.

with an ellipse axis end-point identified as the anterior of the mouse) are used in [8] and the commercial Ethovision package by Noldus [14]. Twelve deformable contour templates are used as the pose model by Branson and Belongie [7]. A lower dimensional ellipse pose model is used to localize the detection area for the more elaborate deformable template model. de Chaumont et al. [15] model the mouse as a head, belly, and neck, with corresponding constrained displacements between each part. Each part is represented by a 2D articulated rigid body model. The parts are linked together through a physics model that defines the motion constraints between the parts. For pose detection, [11–13] simply fit an ellipse to the observed foreground. During occlusion, Pistori et al. [11] employ a particle filter to predict pose, while Branson et al. [12] generate the poses acausally once occlusion ends. A more elaborate cascaded pose regression method [16] is used in [17]. Branson and Belongie [7] use a mutliblob tracker for the ellipse detection and particle-filter contour tracker for contour detection. de Chaumont et al. [15] use the foreground binary mask and edges for initial alignment and mean-shift processes drive the physics engine for refinements.

The hardware setups for all the described methods make it straightforward to define pose parameters in 2D image domain. Given the geometry and optics, physical correspondence is easily established by scaling. Scaled orthography, however, does not apply for the hardware system used for the work presented in this paper for two reasons: (1) the use of a fisheye lens, and (2) the very close proximity of the lens to the monitoring arena. Instead of defining pose parameters in image domain, which might well be uninformative, we instead define pose parameters as the 3D physical coordinates of the mouse posterior and anterior.

3 Method

In what follows, we describe the pose estimation methods employed for solitary home-cage housed mice. Since the ingenuity of the approach is motivated by the uniqueness of the hardware configuration, we start with a quick overview of the hardware system. We then describe the segmentation and pose estimation processing modules.

3.1 Hardware System

The original hardware design is thoroughly described in [4]. A quick overview is herein presented to make this paper self-contained. The video acquisition system is designed to operate in the near-infrared (NIR) spectrum, hence producing monochromatic video. The design employs two cameras fitted with fisheye lenses that are positioned very close to the cage (i.e., $< 5\,\text{mm}$). The lenses are mounted near the top of the cage front and rear walls with a downwards tilt of 25°. We have enhanced the NIR illumination uniformity of the prototype by replacing the NIR LED strips within each side assembly with custom designed LED array. The LED array spans the majority of surface of each side assembly. Translucent acrylic is used to diffuse the LED sources resulting in uniform illumination.

We also augmented the prototype with two additional cameras, one at each end of the cage. The cameras were positioned at mid-height and pointed horizontally (i.e., no tilt) into the cage. Lastly, we've designed an overhead camera system to synchronously capture top-down view of the cage. The overhead system is strictly used for video acquisition related to building training sets, and is not utilized at runtime. The additional cameras proved instrumental in both generating the unique datasets and enabling the novel development and validation presented in this paper. Figure 1a shows an image of the prototype. The front cameras are labeled in the figure, whereas the rear cameras are not seen. Due to cage obstructions (e.g., water and food baskets), each camera view is mainly limited to its side of the cage (e.g., rear or front). The algorithms are coded such that if the mouse is closer to the front of the cage, the estimation is done through a front camera image, and vice-versa. It is also noted that despite our augmented hardware system having two cameras at each end (i.e., horizontal and angled), the algorithms are trained on a pre-chosen camera, and subsequently run strictly on images from the camera on which they are trained. In other words, the addition of the horizontal camera to the original system was not with the intent of making the system a binocular vision system, but rather to facilitate construction of training sets. However, we do take advantage of the availability of the horizontal camera to compare the estimation accuracy between it and the angled camera. While using both tilted and horizontal views at once would likely lead to more accurate results, using a single camera for each end of the cage would be desirable if one is concerned about video storage requirements, processing expense (e.g., future real-time processing), and hardware cost and simplicity.

3.2 Segmentation

Segmentation identifies mouse pixels in the image. Although the mouse is generally darker than background, segmentation based simple intensity thresholding produces poor results due to four main factors: (1) the large disparity in pixel intensity values between the backside of the mouse and its underside, (2) the presence of dark regions in the cage with pixel intensity ranges overlapping those of the mouse (e.g., between food and water baskets), (3) the variability in background intensity patterns in and around the cage-floor region resulting from frequent bedding changes, and (4) the significant shadows cast by the mouse on the bedding. Figure 4 shows example frames highlighting the challenges of segmentation. Our segmentation method capitalizes on the constrained environment and the constant camera position. We build a segmentation model for each camera (i.e., horizontal and angled at both front and rear). Mouse pixels were manually annotated in a set of 250 images from each camera. The images were selected to account for the varied mouse appearance in different positions and poses within the cage. Approximately 350,000 foreground and 350,000 background pixels are chosen randomly from the images and used to train each tree of an 8-tree decision forest classifier to predict a binary label (foreground vs background) for each image pixel. To derive discriminative features, a set of information channels registered to the image are obtained through linear and nonlinear transformations

of the image as per [18]. Namely, we use the intensity gradient magnitude and the 4-bin histogram of orientation of the gradients (HOG). The feature vector for each labeled pixel includes its intensity value along with the values of the feature channels at the pixel position. Additionally, to exploit the stationary camera placement, the pixel's (x, y) location in the image are included in the feature set. Using the pixel image location as a feature results in region-specific classification rules and more robust thresholds (e.g., a more robust intensity threshold against the bright panels, etc.). To segment an incoming image during run-time, the feature channels are first computed for the whole image. The feature vectors for each pixel are formed by concatenating its intensity value, x and y image locations, and feature channel values (i.e., gradient magnitude and HOG as used in training). The decision forest is evaluated for each pixel's feature vector. The returned result is a value representing the probability that the pixel is foreground. A segmentation probability image map, which is pixel-to-pixel registered to the intensity image, is formed by setting the value of location (x, y) to the returned foreground probability value. The foreground probability map is converted to a binary image by thresholding. The threshold level can be selected empirically based on visualizing segmentation results. Alternatively, a more systematic method of tuning a threshold to achieve a desired precision or recall can be employed. Connected component analysis is run on the binary segmentation mask. Size-based filtering is employed to discard small connected components deemed as noise. The largest connected component is regarded as the mouse. Statistics of the binary silhouette such as ellipse fit parameters, ellipse axes end points, bounding box, and area are computed. The area is used to decide which camera will be used for pose estimation. Namely, if the mouse area in the front camera's image is bigger than the area in the rear camera's image, then the front camera image is used and vice-versa.

3.3 Pose Estimation

Pose estimation recovers the three dimensional physical coordinates of the mouse anterior and posterior from a monocular image. The motivation behind 3D posterior/anterior position estimation is to obtain a meaningful measure of mouse activity and behavior. Motion analysis relying on 2D image positions could be non-informative due to the significant distortions resulting from the fisheye lens and its close proximity to the cage. The pose estimation problem is formulated as a non-parametric regression supervised learning task. Hence the objective is to find a mapping $f(\cdot)$ from feature space $\mathcal{X}$ to continuous pose parameters space $\mathcal{Y} \in \mathbb{R}^D$ given a training set $\mathcal{S} \subset \mathcal{X} \times \mathcal{Y}$. Each pose parameter entry $y \in \mathcal{Y}$ in the training set constitutes six parameters denoting 3D coordinates of the mouse posterior (p) and anterior (a). Namely, $y = [p, a], p = [p_i, p_j, p_k], a = [a_i, a_j, a_k]$, where i, j, k are the three axes of the Cartesian coordinates system. Corresponding to each y is a vector $x \in \mathcal{X}$ of features drawn from the image of the mouse having pose y. The challenge in learning $f(\cdot)$, however, is to construct the ground truth set for the pose 3D coordinates, i.e. $y's$. The mouse is highly deformable and the fisheye lens placed close to the cage rules out the assumption of scaled

orthography. Both factors impede recovery of 3D coordinates from a single image. Hence, as described in Sect. 3.1, we augmented the system with an overhead camera with acquisition synchronized to the side-view cameras. All cameras are calibrated such that each image point maps to a line in 3D space. The same mouse key point (e.g., anterior or posterior) is manually marked in two views, namely the horizontal view and the overhead view as shown in Fig. 2. For each view, the 3D line corresponding to the marked image point is computed. The 3D point at which distance between the two resulting lines in minimum is regarded as the ground truth 3D position for the key point. The full training set comprises approximately 200,000 annotations (e.g., two points on side view image and two points on top-down image). To aid the human annotators and speed up the task, the frames were segmented to isolate the mouse and posterior/anterior were pre-annotated as the fitting ellipse major axis end points. Since the end points were arbitrarily designated as posterior/anterior, in most of the cases the annotator's task was to reverse the designation. In some cases, where the major axis end points did not align well with posterior/anterior of the mouse, the annotator would displace the pre-annotated points to more suitable locations in the image.

The vector x is populated with two sets of features. The first set is statistics drawn from the binary silhouette returned by the segmentation module. The features include silhouette area, ellipse fit parameters (e.g., orientation, centroid, length of major and minor axes), and bounding box parameters. The second set is pixel intensity value lookups for randomly chosen locations within the detection window. To compute N such features, a set of positions $\{\phi_n\}, n \in \{1, ..., N\}$ is randomly chosen at training time. Each position ϕ_i is specified as relative offsets from the binary silhouette bounding box, i.e. $\phi_i = (o_x, o_y), o \in [0, 1]$. To compute the feature value, the offsets are scaled to the size of the bounding box, i.e., $\phi_i^s = (o_x \cdot b_w, o_y \cdot b_h)$, where b_w and b_h are the bounding box width and height respectively and the superscript s denotes the scaled ϕ. The feature is then simply computed as $I_b(\phi_i^s)$, where I_b denotes the subset of the whole image I enclosed by the binary silhouette bounding box. The feature extraction concept is illustrated in the cartoon shown in Fig. 3 for $N = 6$. Our implementation uses $N = 125$.

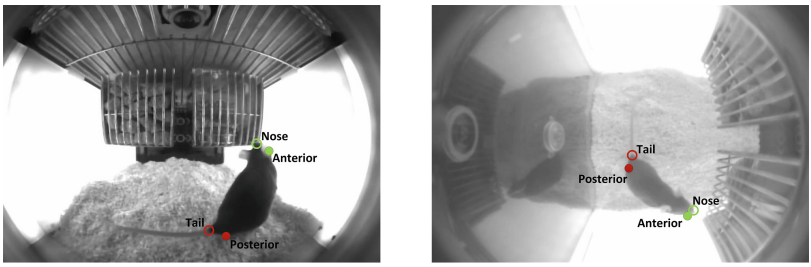

Fig. 2. Mouse in different camera views at the same time instance, shown with example posterior/anterior as well as tail/nose manual annotations. (a) horizontal camera view (b) overhead mounted camera view.

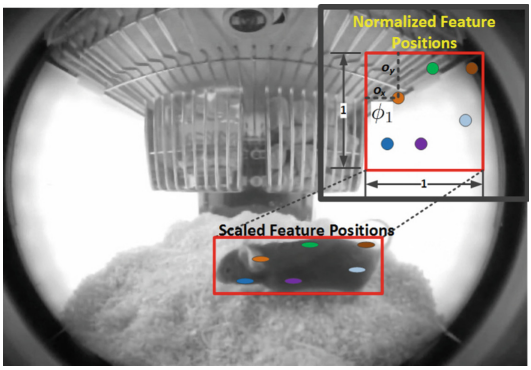

Fig. 3. Cartoon demonstrating the scaling of normalized feature positions to compute the ferns-like features.

The set of discriminative feature vectors each paired with the corresponding ground truth pose are used to train regression forests to act as the mapping function $f(\cdot)$. We treat the parameters (e.g., coordinates) as uncorrelated, and for each of the six parameters of y, a regression forest $f_i(\cdot), i \in \{1, \ldots, 6\}$ is trained to estimate parameter y_i separately. Regression forests constitute an effective non-parameteric regression technique and are well described in literature, e.g. [19].

4 Results

We have built four segmentation models, one for each camera (cage-front angled camera, cage-front horizontal camera, cage-rear angled camera, and cage-rear horizontal camera). Since the decision forests for segmentation yield a probability map for foreground, we use 0.7 as a threshold to convert the map to a binary image. The threshold was selected empirically such that the foreground pixels are well matched to the mouse pixels in the image. We used 60 images with ground truth annotations that were set aside for testing purposes (i.e., not included in segmentation classifier training) to compute the precision/recall for the chosen threshold. The computed values achieved 94 % precision with 85 % recall. Figure 4 shows example segmentation results.

To establish a basis for assessing key point estimation performance, a set R of approximately 6,000 frames was redundantly annotated to provide a range of acceptable deviation between annotations. The chosen frames account for a wide variety of mouse posture in different positions within the cage. Noting that posterior and anterior do not correspond to a single well-defined point on the mouse, but are rather proximity designations (mainly corresponding to the ellipse endpoints as explained in Sect. 3.3), the redundant annotations aimed to establish a Euclidean distance range for 3D posterior position deviation relative to the mouse tail and 3D anterior position deviation relative to mouse nose (refer to Fig. 2). Hence, for the frames in R having posterior/anterior annotations, an

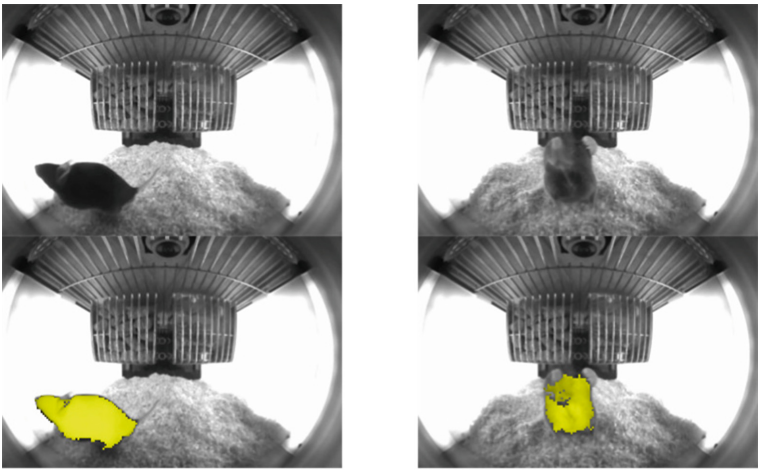

Fig. 4. Example segmentation output for challenging frames highlighting the method's (a) robustness to shadows and dark background regions (i.e., center of image) (b) detection of lighter underside of the mouse.

annotator carefully labeled the tail and nose. For each frame in the redundantly annotated set, the 3D points for posterior/anterior $y = [p_i, p_j, p_k, a_i, a_j, a_k]$ and the 3D points for tail/nose $\bar{y} = [t_i, t_j, t_k, n_i, n_j, n_k]$ were reconstructed via the calibration mappings. The variance, σ_i^2 of each coordinate in $y(i)$ around the corresponding coordinate in $\bar{y}(i)$ was computed as the variance of the distance $\|y(i) - \bar{y}(i)\|$, $i \in \{1, \cdots, 6\}$, where $\|\cdot\|$ is defined as Euclidean distance. We define a distance measure similar to that proposed in [16] utilizing the observed variances σ_i for all six coordinates to equally weigh the estimation errors for each coordinate. The distance, $d(\hat{y}, \bar{y})$, is computed between the regression models output $\hat{y} = [\hat{p}, \hat{a}]$ and the earlier defined $\bar{y} = [t, n]$ which is regarded as ground truth. Namely,

$$d(\hat{y}, \bar{y}) = \sqrt{\frac{1}{6} \sum_{i=1}^{6} \frac{1}{\sigma_i^2} (\hat{y}(i) - \bar{y}(i))^2} \tag{1}$$

In addition to the distance measure, we define a metric to deem an estimation output $\hat{y}$ as either a success or a failure, as is done in [16]. The metric is based on the normalized distance measure of Eq. (1) and an unweighted overall distance measure defined as $\tilde{d}(\hat{y}, \bar{y}) = \|p - t\| + \|a - n\|$. We let d_{thr} be the normalized distance (1) such that 99 % of R, the redundantly annotated frames, are within d_{thr} of each other. We let $\tilde{d}_{thr}$ be the unweighted overall distance such that 99 % of R are within $\tilde{d}_{thr}$ of each other. Our metric for success is if $d(\hat{y}, \bar{y}) < d_{thr}$ and $\tilde{d}(\hat{y}, \bar{y}) < \tilde{d}_{thr}$. The computed thresholds for R where $d_{thr} = 2.73$ and $\tilde{d}_{thr} = 69\,\text{mm}$.

To evaluate the accuracy of the regression models, the models were applied to the frames in set R, which was held out of training. For each frame, the feature vector is formed as described in Sect. 3.3. The vector is then fed to all regression forests $f_i(\cdot)$ to separately estimate each parameter in $\hat{y}$. The output $\hat{y}$ is compared to $\bar{y}$, the reconstructed 3D points for the tail/nose annotations, which are regarded as ground truth. Namely, we compute both distance measures, $d(\hat{y}, \bar{y})$ and $\tilde{d}(\hat{y}, \bar{y})$. The computed distance measures are then compared to the thresholds to set the failure rates. For successful estimates, a mean d and $\tilde{d}$ are computed as well. To analyze the influence of training parameters on accuracy, different regression models were built by varying training parameters including number of trees, image resolution, and training set size. Another variation was to compare taking the median versus the mean of the leaf-node predictions from the trees. Additionally, the hardware system equipped with a horizontal and angled camera offers a unique opportunity to assess accuracy as a function of camera view-point. Recall that the horizontal cameras were added, as stated in Sect. 3.1, to aid in generating training sets. While having a system with two cameras (i.e., horizontal and angled) might lead to greater accuracy, it is desirable to limit the number of cameras per system. Having multiple cameras for each end of the cage would increase the storage requirements for the output video and increase the processing load. To compare the accuracy of estimates as a function of camera view-point, one set of regression models was built to estimate pose from horizontal camera images, and another set of models was built to estimate pose from angled camera images. The result of the comparison between horizontal and angled cameras helps with design choices for such compact systems (i.e., if results are more accurate using horizontal versus angled camera). The base model was chosen to be the horizontal camera, using 50 trees, taking the median of the leaf-node predictions, with features drawn from $\frac{1}{2}$ scale image. Table 1 shows the failure rates. The table also shows the mean distance

Table 1. Results of algorithm training parameters sweeps for horizontal and angled cameras

Parameters	Horizontal			Angled		
	% fail	d mean	$\tilde{d}$ mean	% fail	d mean	$\tilde{d}$ mean
Trees = 25	0.91	0.84	23.2	0.96	0.87	23.6
Trees = 50	**0.83**	**0.84**	**23.2**	0.86	0.87	23.6
Trees = 75	0.81	0.84	23.2	0.98	0.86	23.6
Trees = 100	0.83	0.84	23.2	0.93	0.86	23.6
Image Scale = 1	0.88	0.85	23.3	0.11	0.87	23.6
Image Scale = 0.25	0.91	0.84	23.2	0.10	0.87	23.6
Mean of leaf-nodes	0.78	0.88	23.9	0.88	0.90	24.1
70 % of Training Set	6.73	1.05	28.1	6.88	1.06	28.3
52 % of Training Set	7.49	1.09	28.9	7.37	1.10	29.3
45 % of Training Set	8.30	1.11	29.6	8.61	1.10	29.4

measures for successful estimates. Each entry in the table shows the results of varying a single training parameter relative to the base model. It is clear that the estimation is not sensitive to any of the parameter changes except for the training set size.

5 Discussion

We have demonstrated a viable algorithmic path for accurately estimating 3D posterior/anterior positions of a mouse from monocular fisheye distorted images. These or similar types of images will likely arise in specialized compact systems designed for large scale use in animal vivaria. Our methods capitalize on the constrained environment and known tracking subject to overcome challenges caused by the unusual camera configuration and the highly deformable tracking target. We experimented with algorithm training parameters and demonstrated, as per Table 1 that the accuracy is robust to changes in training parameters. We also experimented with two camera orientations: the horizontal view and angled view. Table 1 suggests that both camera views produce similar results. While an algorithm relying on both cameras horizontal and angled cameras (at both the cage front and rear) to estimate pose would likely be more accurate, some users may wish to decide, for practicaly reasons such as goals aimed at real-time processing, to limit the amount of video data stored and/or processed. The training and testing sets utilized for this study are for a limited mouse size range. Encompassing a larger mouse weight range would simply involve generating additional annotations for the desired mouse sizes. The uniqueness of the hardware system and the specificity of the algorithms to the custom hardware percludes direct comparison with existing state of the art. The per-frame 3D pose estimates produced by our algorithm, however, provide meaningful position information. The 3D position information can be subsequently used for accurate motion analysis. Burgos-Artizzu et al. [8] have shown that trajectory features derived from pose estimates are discriminant for behavior detection. Overall, the algorithm should provide researchers and animal care professionals accurate measures to assess well-being and phenotypical changes.

The training set and the videos are available online (scorhe.nih.gov).

References

1. Jacoby, R., Fox, J., Davisson, M.: Biology and diseases of mice. Lab. Anim. Med. **2**, 35–120 (2002)
2. Conn, P.M.: Animal Models for the Study of Human Disease. Academic Press, London (2013)
3. Noldus, L.P., Spink, A.J., Tegelenbosch, R.A.: Ethovision: a versatile video tracking system for automation of behavioral experiments. Behav. Res. Meth. Instrum. Comput. **33**(3), 398–414 (2001)
4. Salem, G.H., Dennis, J.U., Krynitsky, J., Garmendia-Cedillos, M., Swaroop, K., Malley, J.D., Pajevic, S., Abuhatzira, L., Bustin, M., Gillet, J.-P., et al.: Scorhe: a novel and practical approach to video monitoring of laboratory mice housed in vivarium cage racks. Behav. Res. Meth. **47**(1), 235–250 (2015)

5. Steele, A.D., Jackson, W.S., King, O.D., Lindquist, S.: The power of automated high-resolution behavior analysis revealed by its application to mouse models of huntington's and prion diseases. Proc. Natl. Acad. Sci. **104**(6), 1983–1988 (2007)
6. Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A.D., Serre, T.: Automated home-cage behavioural phenotyping of mice. Nat. Commun. **1**, 68 (2010)
7. Branson, K., Belongie, S.: Tracking multiple mouse contours (without too many samples). In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 1039–1046, June 2005
8. Burgos-Artizzu, X.P., Dollár, P., Lin, D., Anderson, D.J., Perona, P.: Social behavior recognition in continuous video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1322–1329. IEEE (2012)
9. Ohayon, S., Avni, O., Taylor, A.L., Perona, P., Egnor, S.R.: Automated multi-day tracking of marked mice for the analysis of social behaviour. J. Neurosci. Meth. **219**(1), 10–19 (2013)
10. Farah, R., Langlois, J., Bilodeau, G.-A.: Catching a rat by its edglets. IEEE Trans. Image Process. **22**(2), 668–678 (2013)
11. Pistori, H., Odakura, V.V.V.A., Monteiro, J.B.O., Gonçalves, W.N., Roel, A.R., de Andrade Silva, J., Machado, B.B.: Mice and larvae tracking using a particle filter with an auto-adjustable observation model. Pattern Recognit. Lett. **31**(4), 337–346 (2010)
12. Branson, K., Rabaud, V., Belongie, S.J.: Three brown mice: See how they run. In: VS-PETS Workshop at ICCV (2003)
13. Zarringhalam, K., Ka, M., Kook, Y.-H., Terranova, J.I., Suh, Y., King, O.D., Um, M.: An open system for automatic home-cage behavioral analysis and its application to male and female mouse models of huntington's disease. Behav. Brain Res. **229**(1), 216–225 (2012)
14. Noldus EthoVision-XT (2016). http://www.noldus.com/animal-behavior-research/products/ethovision-xt
15. de Chaumont, F., Coura, R.D.-S., Serreau, P., Cressant, A., Chabout, J., Granon, S., Olivo-Marin, J.-C.: Computerized video analysis of social interactions in mice. Nat. Meth. **9**(4), 410–417 (2012)
16. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1078–1085, June 2010
17. Burgos-Artizzu, X.P., Hall, D.C., Perona, P., Dollár, P.: Merging pose estimates across space and time. In: BMVC (2013)
18. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: BMVC (2009)
19. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found. Trends Comput. Graph. Vis. **7**(2–3), 81–227 (2012)