RAT: Robust Animal Tracking

Rana Farah, J.M. Pierre Langlois, Guillaume-Alexandre Bilodeau
Computer and Software Engineering Department
École Polytechnique de Montréal
Montréal, Quebec, Canada
{rana.farah, pierre.langlois, guillaume-alexandre.bilodeau}@polymtl.ca

***Abstract*—Determining the motion pattern of laboratory animals is very important in order to monitor their reaction to various stimuli. In this paper, we propose a robust method to track animals, and consequently determine their motion pattern. The method is designed to work under uncontrolled normal laboratory conditions. It consists of two steps. The first step tracks the animal coarsely, using the combination of four features, while the second step refines the boundaries of the tracked area, in order to fit more precisely the boundaries of the animal. The method achieves an average tracking error smaller than 5% for our test videos.**

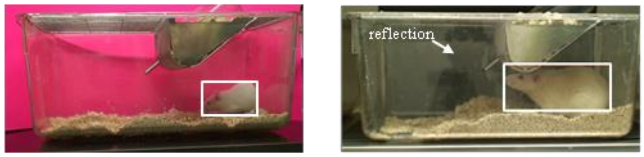
Keywords - Computer vision; animal tracking; optimizing score; dimension refinement.

I. INTRODUCTION

In biomedical experiments, the behaviors and reactions of the experimental subjects are very strong indicators of their emotional state. For instance, the exploration patterns of rodents (and rats in particular) are indicators of their stress condition. In fact, when a rat is stressed, it has a tendency to stay stationary in one place, most often in a corner of its cage. Whereas, when the rat is more at ease and feels safe, it will move around its cage in a pattern of exploration. The stress pattern is often an indicator for the rodent's reaction to drugs or other stimuli, which help researchers draw or confirm conclusions.

Several computer vision algorithms have been developed to track a target and determine its position in space and time, and then deduce its exploration pattern. However, actual biomedical settings impose severe operational constraints on these algorithms. For instance, in a normal laboratory setting, lighting is seldom controlled and cages are stored on shelves, or are connected to other devices. This severely restricts camera positions, angles and fields of view. Animal cages usually contain bedding, which arrangement is very dynamic due to the movements of the rodent. The material, with which the cages are made, may cause reflections (see Fig. 1-b) or may have scratches on them. The animal to observe may have a color similar to its background and may blend in it. Rat's bodies in nature are very deformable making them very hard to model and to track.

In this work, we aim to track an animal in an uncontrolled environment as described above. The method that we propose is composed of two steps. The first step combines four features to constitute a robust tracker, and the second step adjusts the tracked region to the boundaries of the rat. Our method is based on previous works that use several features for tracking (for instance, Yin et al. [1]). One of the contributions of this paper consists in introducing a new feature when tracking a rat in a cage, the edge density D_{He}. This new feature allows us to deal with the dynamic nature of the bedding and reflections on metallic parts if they exist. We also propose a new way to delimit the boundaries of a target using edge-based constructed pulses. The target area is defined by the largest pulses intersecting in X and Y. Our method operates in settings that are typical of a biomedical laboratory except for the introduction of a colored cardboard in some cases to reduce reflections (see Fig 1). No special cages were used, lighting was unchanged, and the cages were left on their shelves, which made only side views possible.



(a) (b)

Figure 1: Initialization at first frame.

The paper is structured as follows: Section II describes related work on animal tracking. Section III details the methodology that we used for tracking animals. Section IV presents the experimental settings and results. Section V concludes this paper.

II. LITERATURE REVIEW

In this section, we will present previous works that involved tracking various types of animals. Cangar et al. [2] and Tillett et al. [3] used active contours to track cows and pigs in their stalls. The video sequences were recorded from a top view. In such case, active contours are a good choice, because the shapes of cows and pigs, especially from a top view, are not very deformable, which makes it easier to tune the energy function of the active contour.

Other works involved tracking several mice at a time. Among these works, we can cite Pistori et al. [4] and Goncalve et al. [5] that use a particle filter and a k-mean algorithm in order to track the targets. However, these works use white mice on a black background and tracking is preceded by a segmentation using simple thresholding. A certain contrast is needed between the background and the object of interest for the segmentation to be successful. A similar work that concerns the tracking of a single white

This work is funded by CRSNG - Conseil de recherche en science naturelles et en génie du Canada.

mouse on a black background was done by Ishii et al. [6]. Furthermore, Nie et al. [7] [8] also used a background subtraction by having a dark mouse in a transparent container filled with water or by using a Plexiglas cage positioned on top of an IR illuminator.

In all the works cited above, the experiments are controlled in such a way to ensure a certain contrast between the rodent and its background so that a simple extraction can be applied. However, these conditions are not always available in a biomedical environment. In fact the floor of cages are often covered with bedding to insure the comfort of the animal inside the cage (and not to cause stress), and the cages are transparent, so that they allow visual observation to the technicians. Besides, the color of the animal is dictated by its breed, which is, in turn, dictated by the requirements of the ongoing experiments or by the availability of the animals.

Dollar et al. [9] and Belongie et al. [10] did not apply background subtraction or environmental controls. The authors used 3D space-temporal features, in order to determine specific mice behaviors. The 3D features are based on gradients. The authors commented that their method does not perform well because the number of features that are found on the mouse is relatively small, which affected the accuracy of the detector. In several works ([11], [12], [13]), the authors also used 3D space-temporal features preceded by a background subtraction process. A background subtraction is not advisable in our case for three reasons. First, if the background is constructed from an empty cage, then when placing the animal in the cage afterward, the cage can move resulting from the experimenter motion or the motion of the animal. Second, if the background is constructed while the rodent is in the cage, the rodent may spend long durations in one place, and the resulting background will not be reliable as it will contain a phantom shape of the rat at the place where the rat was stationary. Third, the bedding is also displaced by the animal in the cage. It is hard to maintain a background that takes into account those displacements.

Based on previous works, our hypothesis to successfully track the animal is to use a method that does not rely on background subtraction. Thus, we propose a detect and match method based on the previous position of the animal for the search area.

III. METHODOLOGY

To track the animal, a tracker locates the position of the target using a sliding window approach based on gradient and intensity features. Instead of scanning the entire image to detect the animal, our method uses temporal information and scans the image only in the vicinity of the previous animal position. After coarse localization of the animal, the tracked region boundaries are refined to account for change in pose, scale, and deformation of the animal using edge information.

A. Coarse Animal Localization

The tracker is based on a sliding window approach. Because the animal body is highly deformable and may change shape very quickly, it is quite difficult to predict and to adapt the window size automatically and reliably. Thus, we elected to track the animal in two steps. The first step just aims to localize roughly the position of the animal. As such, our sliding window tracker relies on a $N \times M$ window having fixed dimensions for all the video frames. The dimensions of the window are kept fixed for practical reasons. The dimension of the window affects the speed of computation, and allowing the window to change size to test many hypotheses would results in tracking at multiple scales, which would slow the computation. Also, the system will become unstable, and the window may grow indefinitely due to noisy edge structures that may erroneously be associated to the animal. Currently, the initial $N \times M$ window is drawn manually around the target at the first frame. In Fig. 1, the initialization at the first frame is shown for two animals of different size.

Determining the position of the target at time $t+1$ is done by scanning an area that is located around the target at time t, by sliding the selected $N \times M$ window. At each displacement of the probing window, the region bounded by it will represent a candidate for the target and a corresponding fitness cost function (S_f) is evaluated. The target location at time $t+1$ is the position that minimize S_f.

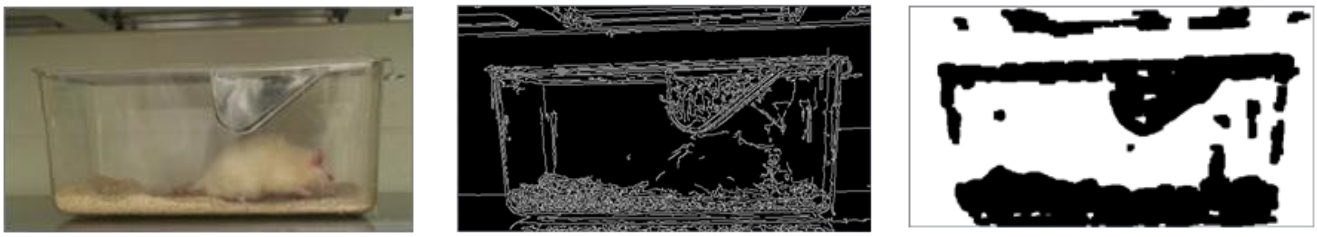
The fitness cost function S_f is based on four features which are: the histograms of oriented gradients (*HOG*), the histograms of intensity (*HI*), the quantity of motion, and edge density.

All these features constitute weak tracking features on their own. However, the combination of the four features forms a strong composite feature that leads to a more robust tracker.

HOG was chosen because, to a certain extent, it remains invariant to geometric transformations on the target and to changes of illumination, since it operates on small regions (cells). However, the noisy nature of the bedding and the rest of the background may distract a tracker based solely on *HOG*.

Similarly, *HI* was chosen in order to exploit the color characteristics of the target keeping in mind that target deformation and illumination variance may affect its color distribution. Still, the background may share some of the colors that belong to the target, which may decrease the effectiveness of the *HI* feature.

The quantity of motion is taken into consideration, because, except for the bedding, the target is the only entity that is dynamic in the scene. When motion is detected, it gives a strong indication of the position of the target. Nevertheless, motion information may be corrupted by reflection on the cage. These reflections would give the impression of a moving target, while the target is stationary. In addition, the motion of the rat causes displacement in the bedding. These displacements will also be attributed to the rat and will effectively increase the apparent size of the rat.



(a) (b) (c)

Figure 2: Mulch extraction - (a) A snapshot of an RGB frame. (b) The frame after computing the canny edge detection on a gray frame. (c) The extracted $\mathbf{D_{He}}$.

Moreover, bedding has a texture that is characterized by a high density of edges. The reflection on the metallic part of the cage, also results in a high density of edges. For this reason, the edges density is considered a feature of interest. It is used to suppress the bedding and the metallic part in the scene, resulting in less distraction for the three other features.

The fitness cost function (S_f) is calculated as follows

$$S_{f(w_i)} = \alpha_1 D_{HOG(w_i)} + \alpha_2 D_{HI(w_i)} - \alpha_3 Q_{m(w_i)} + \alpha_4 D_{He(w_i)} \quad (1)$$

where w_i is the i^{th} candidate window, D_{HOG} is the distance between the *HOG* feature matrices of the target window at time t and the candidate window at time $t+1$, D_{HI} is the distance between their *HI* feature matrices. The Euclidean distance is used to calculate these two distances. Q_m represents the quantity of motion in the given region. D_{He} is the edge density. As mentioned above, S_f should be minimized. This is justified by the significance of the terms that constitutes S_f and their signs. D_{HI} and D_{HOG} are both distances, which means, that they represent a penalty: the bigger the distance, the less is the resemblance between the window and the target. D_{He} represents the edge density. Given that we should avoid the bedding, the highest D_{He} is in a window, the more the window should be penalized. In contrast, Q_m represents the quantity of motion. Motion is attributed to the target, so the more a window includes motion pixels the more it should be rewarded. Having positive signs attributed to the penalties and negative signs attributed to the rewards, the best S_f is the smallest.

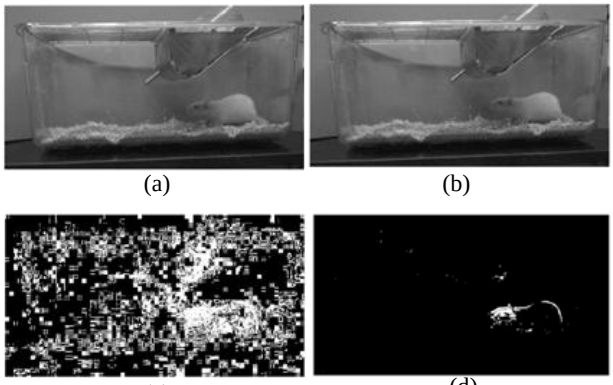
The *HOG* feature is derived from the algorithm described Dallal et al. in [14].

The *HI* feature matrix is calculated in a similar manner:

1. The candidate region is divided into $n \times m$ overlapping cells.
2. An intensity histogram, of b bins, is calculated for each cell as follows:

$$h(r_b) = n_b, \quad (2)$$

where r_b is one of the calculated intensity intervals, n_b is the number of pixels in the frame which intensities are included in r_b.



(a) (b) (c) (d)

Figure 3: Motion Extraction - (a) Frame 17. (b) Frame 18. (c)|Frame 17 - Frame 18|. (d) The frame after thresholding

3. The *HI* feature matrix is constructed as a $nm \times b$ matrix where

$$HI(k) = \frac{h_k}{norm(HI)}, \quad (3)$$

and h_k is the histogram of the k^{th}cell.

D_{He} is calculated by extracting the edges in the frame using Canny's edge detector [16], and then dividing the frame with a grid of small squares, of $\eta \times \eta$ pixels. Then the number of pixels that belongs to an edge in each of those squares,$N_e(l)$, is counted. l represents the l^{th} square. Subsequently, given ϵ_1 as threshold,

$$D_{He}(l) = \begin{cases} 1 & if\ N_e(l) > \epsilon_1 \\ 0 & otherwise \end{cases}, \quad (4)$$

Fig. 2(a, b, and c) shows, respectively, an original frame, its edge representation , and the result of calculating D_{He}.

Fourth, *Qm* which represents the quantity of motion is calculated as follows:

$$Q_m = \begin{cases} 1 & if\ |F_t - F_{t+1}| > \epsilon_2 \\ 0 & otherwise \end{cases}, \quad (5)$$

where $F_t\ and\ F_{t+1}$ are two consecutive grayscale frames and ϵ_2 a given threshold. Fig. 3 illustrates Q_m calculation.

B. *Boundary Refinement*

The tracker finds the coarse location of the target, but its location and size need to be refined in order to accommodate the possible change in dimensions of the target and to correct tracking errors.

We found that the edge information is convenient to accomplish the boundary refinement because we can take advantage of the edges that delimits the boundaries of the target. However, the edge map contains a large number of edges that belong to the background and to the bedding. These edges will cause distraction to the process of refinement and result in incorrect dimensions. To minimize those edges the regions of high density that were calculated in the first step are subtracted from the frame, leaving only the edges that belong to the target and the remaining noise.

The window position is assumed to be on the target but not necessarily centered on it. In addition, the window dimensions may be smaller than the target, which puts the window inside of the target, or larger than the target, which puts the window's boundaries around the target. Based on these scenarios, the following strategy is applied.

The vicinity of the window (the solid line rectangle in Fig. 4) is scanned to produce pulse graphs. The horizontal pulse graph P_h is constructed by scanning the region horizontally as follow:

$$P_h(x) = \begin{cases} 1 & \text{if } \sum_{y=y1}^{y2} i(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where y1 and y2 are the upper and lower limits of the scanned region and $i(x,y)$ is the intensity of the pixel at (x,y).

Similarly, to produce the vertical pulse graph P_v, the region is scanned vertically and the pulses are produced as follows:

$$P_v(y) = \begin{cases} 1 & \text{if } \sum_{x=x1}^{x2} i(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where x1 and x2 are the left and right limits of the scanned region and $i(x,y)$ is the intensity of the pixel at (x,y).

Afterwards, the resulting pulses are projected on the frame (the light rectangles in Fig. 4), and the region R_{max} that maximizes the intersection of two of those projections is chosen to set the upper, left and right boundaries of the tracker (the dashed lines in Fig. 4).

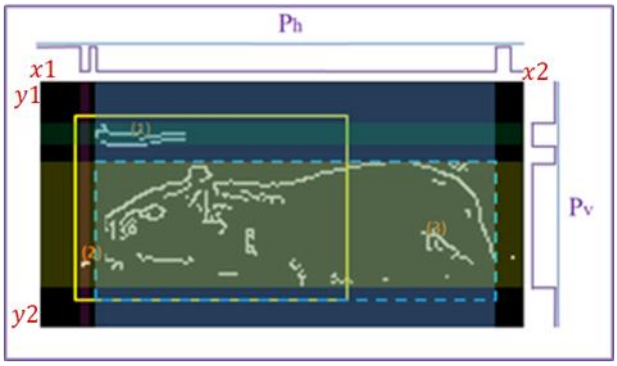


Figure 2: Fitting the window to the target

$$R_{max} = \max(\text{Proj}(P_h(i)) \cap \text{Proj}(P_v(j))), \quad (8)$$

where $\text{Proj}(P_h(i))$ and $\text{Proj}(P_v(j))$ are the projections of two piecewise constant function that belongs to P_h and P_v, respectively.

It must be noted, that edge refinement alone would be insufficient to track because of the noise. If the target moves too fast, the boundaries of the tracker will lock on nearby noise. This weakness is shared with the traditional active contour method [17], as addressed by Hao et al. in [18].

IV. Experimental Results

To test the proposed methodology, we recorded videos of rats in the biomedical laboratories of a research hospital. We set up the camera in a way to avoid disturbing the ongoing experiments, and we did not change any of the actual conditions of the environment. The cages were set on shelves and illumination was provided by florescent lamps from the ceiling. The rats were white and of two different size classes (big and small). The cages were transparent, and in some of the cases, we introduced a pink cardboard behind the cage. The cardboard was used to limit the reflection visible on the transparent acrylic cage. The cardboard does not disturb the environment, is easy to setup, and is not intrusive.

Two videos were processed; the first video (video1) features a small rat and a pink cardboard (Fig. 1-a), and the second video (video 2), a large rat without a cardboard (Fig. 1-b). These two videos were chosen to represent different settings. The number of frames, frame per second (fps), width and height of the frames and the duration of each of those videos are shown in Table I.

TABLE I. Video information

	Video 1	Video 2
Number of frames	8235	15279
fps	25	
Width	560	
Height	304	
Duration (seconds)	330	611

The same parameters were used for both videos. Table II, summarizes the parameters that were used in the experiments. These parameters are determined empirically, however, they do not seem dependent on the size of the rat, nor the colors involved, given the extent of this experiment. They are more related to the size of the frame and the frame rate. For instance, n and m could be smaller if the size of the frame was smaller to allow a finer granularity.

To provide an objective evaluation of the quality of the proposed tracker, we considered the following quantitative protocol. Sixty frames were selected randomly from each video. A bounding box was drawn manually around the rat in each of those frames and the error on the center distance from the origin is calculated as follows

$$err_{center} = \frac{\sqrt{x_{man}^2+y_{man}^2}-\sqrt{x_{track}^2+y_{track}^2}}{\sqrt{x_{man}^2+y_{man}^2}} \times 100, \qquad (9)$$

where, (x_{track}, y_{track}) are the coordinates of the center of the refined window, and (x_{man}, y_{man}) are the coordinates of the manually drawn bounding box.

TABLE II. THE EXPERIMENTAL PARAMETERS

parameters	Equation involved	values
$n \times m$		16×16
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	(1)	(1,1, 0.01,0.0005)
b	(2)	9
$\eta \times \eta$		5×5
$\in_1$	(4)	50
$\in_2$	(5)	5

TABLE III. TRACKING ERROR (%)

	median	mean	std
Video 1	1.60	1.94	1.64
Video 2	2.32	4.48	5.01

This metric was chosen to calculate the accuracy of the position of the refined window. The results of the mean, median and standard deviation (std) error calculation are shown in Table III. Fig. 5 shows the calculated error for every ground-truth frame. Both Table III and Fig. 5 show that the error in video 2 is greater than that of video 1. This is mostly cause by a bigger quantity of reflections present in video 2. In addition Fig. 5 show some cases were the error goes beyond the average. This error is mainly due to the discontinuity of the rat's edges, which splits the rat in several parts, during boundary refinement. Fig. 6 illustrates this case. The edge on the upper boundary of the rat is split in two. Consequently, it will be represented with two separate pulses in the horizontal pulse graph, and one of them will be chosen to represent the width of the rat.

Noise edges that are too close to the rat constitute another

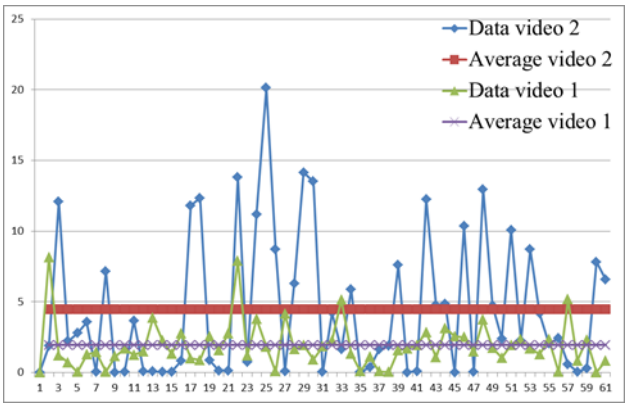


Figure 5: percentage error in video 1 and video 2

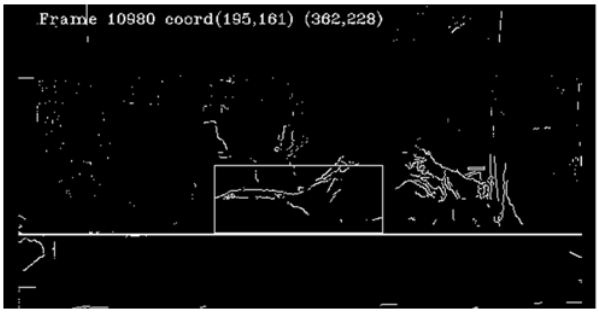


Figure 6: erroneous boundary refinement

source of error. For example, in Fig. 7, some noise edges are close to the upper boundary of the rat. These edges were considered as part of the rat and resulted in a larger apparent height.

Fig. 8 shows some tracking results for videos 1 and 2. The first three snapshots are extracted from video 1 while the last three snapshots are extracted from video 2. Fig. 8 (a, b) show two cases where the tracking was not perfect in video 1,while Fig. 8(c) shows a case where the rat was extracted adequately in video 1. Similarly, Fig. 8 (d, f) represent cases where the tracker was not very accurate in video 2. However, Fig. 8 (f) illustrates a case where the tracking was adequate in video 2. The inaccuracies are due to one or more of the sources of errors described above. For instance, in Fig. 8(b), the small pieces of bedding that are on the front of the cage constitute noise edges that are close to the body of the rat and that are considered as a part of it. However, despite of the mentioned error causes, we calculated an average error that is less than 5% for both videos, which represent a good tracking result.

V. CONCLUSION

In this paper, we presented a method that tracks animals in cages under actual laboratory conditions. The method consists of two steps; the first step tracks the animal coarsely using a constant size window and four features, while the second step refines the boundaries of the tracked area in order to get a better fit on the animal. The tracker uses the combination of four features which are weak when taken individually. Combining the characteristics of all those features, results in a more robust tracker. The tracker's purpose is to achieve good results in an uncontrolled environment. This environment

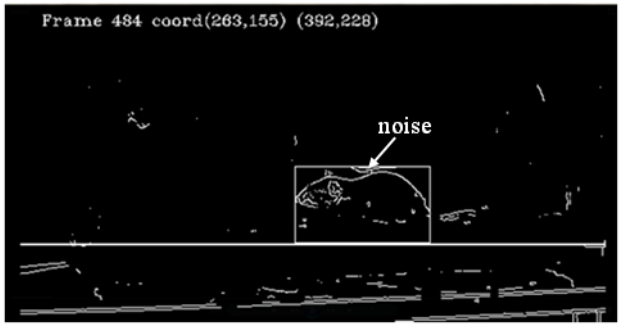


Figure 7: erroneous boundary refinement due to some close noise edges

should be compliant to what is found in a regular biomedical lab. We conducted our experiments in an actual biomedical lab, where we did not alter any of the settings. According to the results that were shown earlier in the article, we succeeded in achieving adequate tracking with an average error less that 5%.

REFERENCES

[1] Z. Yin, F. Porikli, R. Collins, "Likelihood Map Fusion for Visual Object Tracking," IEEE Workshop on Applications of Computer Vision (WACV), pp. 1–7, 2008.

[2] Ö. Cangar, T. Leroy, M. Guarino, E. Vranken, R. Fallon, J. Lenehan, J. Mee, D. Berckmans, "Automatic real-time monitoring of locomotion and posture behaviour of pregnant cows prior to calving using online image analysi," Computers and Electronics in Agriculture, vol. 64(1), pp. 53–60, 2008.

[3] R. D. Tillett , C. M. Onyango and J. A. Marchant, "Using model-based image processing to track animal movements," Computers and Electronics in Agriculture, vol. 17, pp. 249–261, 1997.

[4] H. Pistori, V. Odakura, J.B.O. Monteiro, W.N. Gonçalves, A. Roel, J. de Andrade Silva , B.B. Machado, "Mice and larvae tracking using a particle filter with an auto-adjustable observation model,"Pattern Recognition Letters, vol. 31, pp. 337–346, 2010.

[5] W.N. Goncalves, J.B.O. Monteiro, J. de Andrade Silva, B.B. Machado, H. Pistori, V. Odakura, "Multiple Mice Tracking using a Combination of Particle Filter and K-Means," Computer Graphics and Image, pp.173 – 178, 2007.

[6] H. Ishii, M. Ogura, S. Kurisu, A. Komura, A. Takanishi, N. Iida, H. Kimura, "Development of autonomous experimental setup for behavior analysis of rats," Intelligent Robots and Systems, pp. 4152 – 4157, 2007.

[7] Y. Nie, I. Ishii, K. Yamamoto, T. Takaki, K. Orito, H. Matsuda, "High-speed video analysis of laboratory rats behaviors in forced swim test," Automation Science and Engineering, pp. 206–211, 2008.

[8] Y. Nie, I. Ishii, K. Yamamoto, K. Orito and H. Matsuda,"Real-time scratching behavior quantification system for laboratory mice using high-speed vision," Journal of real-time image processing, vol. 4, pp.181–190, 2009.

[9] P. Dollar, V. Raboud, G. Cottrell, S. Belongie, "Behavior recognition via sparse spatio-temporal features," VS-PETS, pp. 65–72, 2005.

[10] S. Belongie, K. Branson, P. Dollar, V. Rabaud, "Monitoring animal behavior in the smart vivarium," Measuring Behavior, pp. 70–72, 2005.

[11] H. Jhuang, E. Garrote, N. Edelman, T. Poggio, A. Steele, T. Serre, "Trainable, vision-based automated home cage behavioral phenotyping," Methods and Techniques in Behavioral Research, 2007.

[12] H. Jhuang, E. Garrote, T. Poggio, A. D Steele, T. Serre, Vision-based automated recognition of mice home-cage behaviors. Visual observation and analysis of animal and insect behavior, ICPR, 2010.

[13] X. Yu, A.D. Steele, V. Khilnani, E. Garrote, H. Jhuang, T. Serre, T. Poggio, "Automated home-cage behavioral phenotyping of mice," MIT CSAIL Technical Reports, 2003.

[14] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection," CVPR, pp.886–893, 2005.

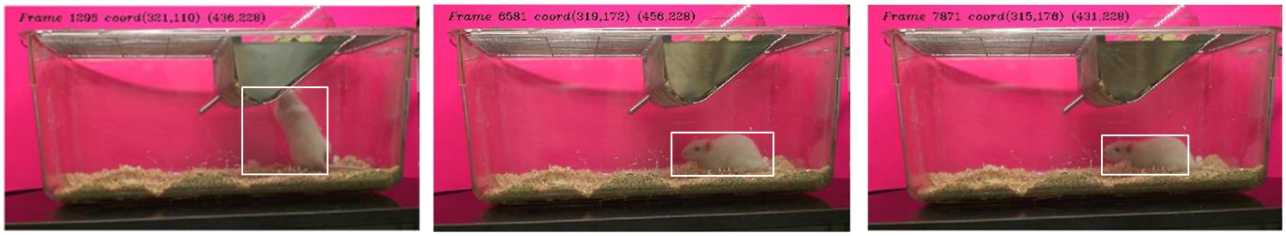
[15] N. Dallal, "Finding people in images and videos," Institut National Polytechnique de Grenoble, 2006.

[16] J. Canny, "A computational approach to edge detection," Transactions on Pattern Analysis and Machine Intelligence, vol 8, pp. 679–698, 1986.

[17] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, "Active shape models - their training and application," Computer Vision and Image Understanding, vol. 61, pp.38–59, 1995.

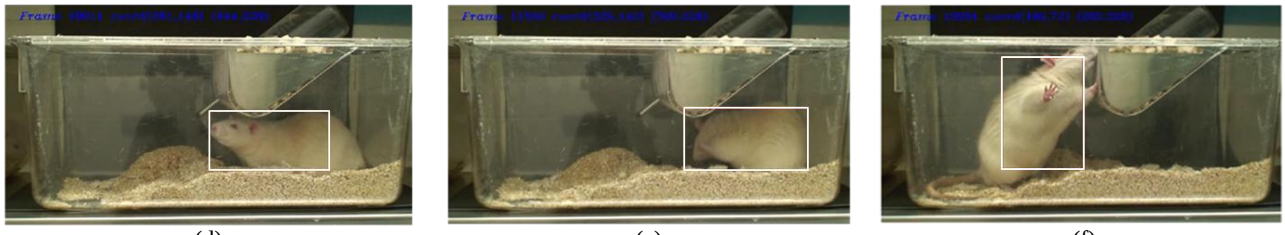
[18] J. Hao and M. S. Drew, "A predictive contour inertia snake model for general video tracking," IEEE ICIP, pp. 413–416, 2002.

Video 1



(a) (b) (c)

Video 2



(d) (e) (f)

Figure 8: Tracking results for video 1 and video 2