Catching a rat by its edglets

Rana Farah, *Member, IEEE*, J.M. Pierre Langlois, *Member, IEEE*, Guillaume-Alexandre Bilodeau, *Member, IEEE*

***Abstract*— Computer vision is a non-invasive method for monitoring laboratory animals. In this article, we propose a robust tracking method that is capable of extracting a rodent from a frame under uncontrolled normal laboratory conditions. The method consists of two steps. First, a sliding window combines three features to coarsely track the animal. Then, it uses the edglets of the rodent to adjust the tracked region to the animal's boundary. The method achieves an average tracking error smaller than a representative state-of-the-art method.**

***Index Terms*— Computer vision, edglets, edge background, overlapped histograms of intensity, animals.**

I. Introduction

An automated non-intrusive animal monitoring system is of great value in biomedical laboratories. It has the potential to dramatically increase laboratory staff efficiency and productivity by reducing or eliminating the time spent reviewing videos or directly monitoring animals. Consequently, a larger quantity of acquired data can be processed, which can lead to better research results in a shorter time.

Automated video monitoring is done using computer vision systems. However, biomedical conditions impose several challenges to those systems, especially when the animals to monitor are rodents. The challenges can be imposed by the environment or the settings, or by prerecorded videos that do not present any consideration for automatic video processing. For instance, lighting in biomedical labs is seldom customized for computer vision processing. Cages are usually stacked on shelves, which restricts the position of the camera and the field of view. Cages can also be connected to other devices, and can be made of several materials that give rise to different type of artifacts, like metal that is prone to noisy reflections or transparent glass that is scratched. The rodents can have the same color as their background and the cages usually contain bedding to ensure the comfort of the animal. The bedding is dynamic due to the rodent's motion. More importantly, rat bodies are very deformable. This characteristic makes them hard to model.

The purpose of this article is to extract rodents, from a scene, using a computer vision system under the conditions stated above. The proposed method consists of two steps. The first step combines three weak features to roughly track the target. The second step adjusts the boundaries of the tracker to extract the rodent. One contribution of this paper is the introduction of a new feature which is the overlapped histograms of intensity (*OHI*). We also propose a new segmentation method to extract the target's boundaries using an online edge-background (e-background) subtraction and edglet-based constructed pulses. Edglets are discontinuous pieces of edges. The online e-background is a continuously updated frame constructed out of the accumulation of background edglets. Our method operates in settings that are typical of a biomedical laboratory. In our test conditions, no special cages were used, lighting was unchanged, and the cages were left on their shelves, which constrained video monitoring to a side view. An early version of this work was presented previously [1]

The paper is organized as follows: Section II describes related work on animal tracking. Section III presents the problem analysis. Section IV details the method that we propose for tracking and extracting animals. Section V presents the experimental settings and results, and section VI concludes the paper.

II. Literature Review

Animal tracking algorithms have been the subject of much research in computer vision because of the available applications and the differences in morphology and behavior between animals and humans. In general, standard human tracking methods cannot be applied directly to animals. Developed methods depend on the applications, and the conditions and limitations of the environmental settings.

For laboratory animals, Pistori et al. [2] and Goncalve et al. [3] used a particle filter combined with a k-mean algorithm to track several mice in a cage. The algorithm extracts blobs to calculate the mice's center of mass and the parameters of a bounding ellipse. The algorithm is restricted to processing white mice on a dark background, because the tracking algorithm starts with a segmentation that uses simple color thresholding. The method used by Ishii et al. [4] suffers from the same restriction. The authors tracked a white mouse on a black background using simple color thresholding for segmentation. They then calculated the center of mass of the foreground to track the rat. Nie et al. [5] [6] used the same segmentation principle to track a dark mouse in a transparent

Manuscript received November 8, 2011. This work was supported by the Conseil de Recherche en Science Naturelles et Génie du Canada (CRSNG) and by the Fonds de recherche de Québec – Nature et Technologie (FQRNT).

The authors are with Departement of Computer Engineering and Software Engineering, École Polytechnique de Montréal, 2500 Chemin de Polytechnique, Montréal, Québec, Canada. (e-mail: rana.farah@polymtl.ca).

container filled with water, or a mouse in a Plexiglas cage positioned above an IR illuminator. Simple segmentation techniques require a certain level of contrast between the background and the target. This restriction is not always realistic due to environmental settings and requirements of an ongoing biomedical experiment. For instance, the animal's breed is usually imposed by the experiment or its availability and the color of the animal is determined by its breed. Moreover, in some environments, cages are stacked on shelves. In this case, transparent cages are used to enable monitoring by laboratory staff. Furthermore, cage floors are often covered with bedding to insure the comfort of the animal and avoid stressing it. As a result, the contrast required by a simple color thresholding is seldom available in biomedical environments.

Dollar et al. [7] and Belongie et al. [8] used 3D spatio-temporal gradient features to track and detect certain specific behaviors in humans and mice. The method does not use segmentation. The authors mentioned that their method does not perform well when used on mice. They explain that the number of features formed on mice isn't sufficient due to the properties of their texture. In [9], the authors used 3D spatio-temporal features preceded by a background subtraction process. On their website, the authors mention that their algorithm is restricted to dark mice over a white background [12]. Above all, a color-based background subtraction is not advisable in uncontrolled environment for three reasons. First, the cage may move due to animal motion. Second, if the background is constructed while the rodent is in the cage, the rodent may stay at one place for long durations, 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. Further, because the animal area may encompass a large portion of the image, some area of the background will be seldom visible even if the animal moves. Third, the bedding is also displaced by the animal in the cage. It is impractical to maintain a color-based background that takes into account those displacements.

In [13], Dollar et al. proposed extracting the edges of a target using a multiple feature classifier. The features include gradients, Harr Wavelets, and difference of histograms computed on filtered frames after applying a difference of Gaussian (*DoG*) or a difference of offset Gaussian (*DooG*).

Branson and Belongie [14] used a particle filter that combines a multiple blob tracker and a contour tracker to track several mice in a cage. The method relies heavily on the animals' contours to fit the tracker. Accordingly, the method requires an edge detector with special characteristics to perform. The Berkley Segmentation Engine [15] was chosen, using 12,000 annotated images to train the edge detector. In addition, it took the detector over five minutes to process one image.

Many commercial solutions exist for rodent tracking. We are aware of only two that provide a computer vision solution. The first solution, by Noldus, provides a tracking software named EthoVision XT [16]. Details of the video tracking method are not available. According to a demo video [16], a

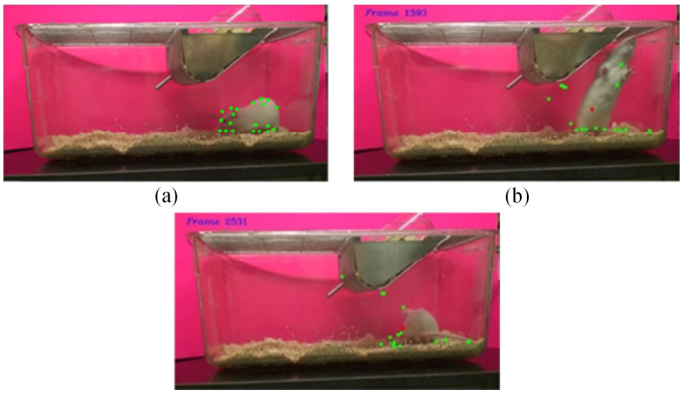


Fig. 1: Point Features for the KLT tracker at (a) frame 1: eventhough the max number of point features was set to 500, KLT initialized with 20 point features on the target, (b) frame 1593: only seven point features remain on the target, (c) frame 2531: only two feature points remain on the target.

template should be defined for each experiment. The template indicates the species of the animal to be tracked, the dimensions and the edges of the cage. Manual adjustment is also required to align the template and the cage. Specific cages are provided by the company. A combination of a visible camera and an infrared camera is required to reduce the sensitivity to fur color variations and to reduce problems caused by light reflections [17]. The second solution, by CleverSys [18] [19], uses a color-based background subtraction technique to extract the rodent as foreground. It then calculates the rodent's center of mass for tracking. For optimal results, the software needs a specialized system that provides adapted lightning and a uniform white background [20].

After considering the previous analysis, we observed that successful target extraction should be done with a method that does not rely solely on color-based background extraction, a single feature, or restrictive and tedious pre-training. Thus, we propose a track-and-refine edglet-based method that does not impose any restriction on actual environment settings as long as the cage is transparent.

III. Problem Analysis

We conducted an analysis based on several features and schemes to determine which is more suitable for target extraction.

When considering the texture, and in particular the gradients, we observe that it is difficult to model our target because its texture is extremely variable. In fact the rat has a very deformable body on which the fur changes in appearance as the rat is moving. For instance, when applying, KLT [21] to our data set, the feature points moved out of the target after a certain number of frames even though the tracker was forced to initialize on the target (see Fig. 1). This behavior is consistent with [7] and [8] as gradients were also used in those papers. Another reason why KLT failed is that the target has less texture with respect to its environment in some cases.

When considering color, the target's appearance and color distribution are extremely dynamic. The rats are often multicolored and share colors with the background. An

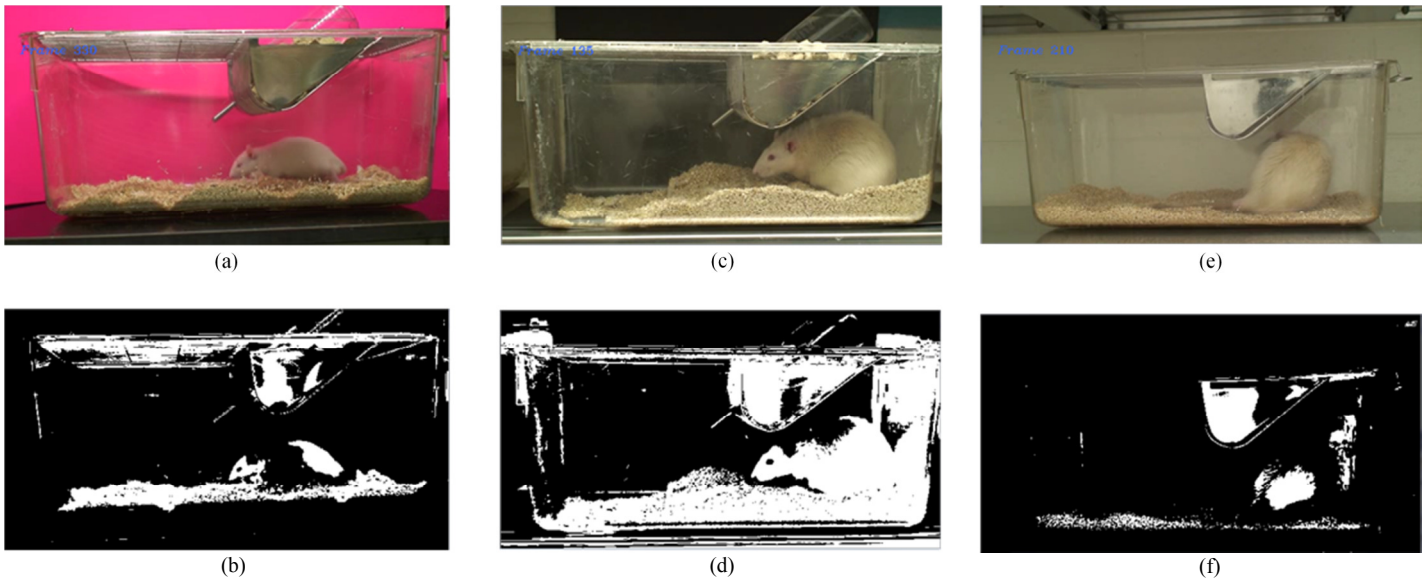


Fig. 2:Target histogram projection. (a,b) a change in the target color distribution cuts the target in two. (c,d) The target and the background share common color zones that give a larger apparent size to the target. (e,f) A change in color distribution of the target gives it a smaller apparent size.

analysis of the histogram projection of a Mean Shift algorithm [22] reveals that the target is divided into several color bands. Fig. 2 shows that the background and the target have common colors. In fact, a Camshift algorithm [23], which is a tracking algorithm based on the Mean Shift segmentation, loses tracking after a few frames. These challenges are common to most color segmentation methods such as [4], [5], [6] and [18].

Another concern arises in the case that the target stays immobile for a long time. This would pose an important challenge to a method similar to the Gaussian Mixture Model (GMM) [24]. GMM is a robust color based segmentation method. Its particularity is that it takes into consideration the background information. In fact, the background is represented by several models, in this case three, that are regularly updated. However, GMM's greatest weakness is that when the target is stationary for a long time it tends to blend with the background (See Fig. 3 (c)).

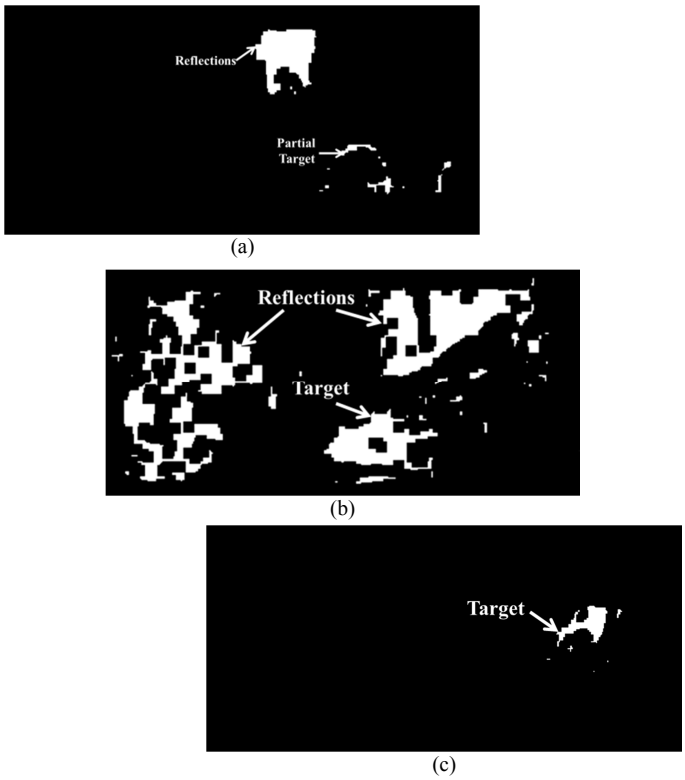


Fig. 3: Foreground segementation using GMM

We also investigated contour features. Contour features are easily distracted by noisy edges. This is also the case in an active contour model [25] that typically use contour features. Furthermore, it is challenging to control their energy function when the target is very deformable. In [26], the authors used a particle filter based on active contour method. To address the previous shortcomings, the authors used both edge information and color information to adjust their contours and they relaxed their energy function. When used on our dataset, this method still failed because both the contour and color distribution of the target were extremely dynamic.

Assuming that the target is the only mobile entity in the frame, motion would form a strong indicator on the position of the target. However, this is not always the case. First, the target may remain immobile for certain duration. Second, the dynamic bedding and the reflections of objects moving in front of the cages gives the impression of motion. Their effect is most significant when the target is stationary as the position of the target could be mistaken with the position of the reflections or the moving bedding.

According to this analysis, we conclude that color, gradients and motion based features are weak features by themselves. This justifies the conclusion reached in the previous section. Neither gradient based features nor color information appear to be sufficient by themselves to extract a target that has such dynamic shape, texture and color distribution from a dynamic background that shares some of the color distribution of the target.

IV. Target Extraction

The proposed method uses a sliding window approach to coarsely localize the target. It then adapts the tracked region boundaries to fit the contour of the target using the target's edglets and an e-background subtraction. Adaptation is necessary to account for changes in the animal's pose, scale and deformation.

A. Coarse Animal Localization

The sliding window scans the vicinity of the rodent's previous position to estimate the current position. The window's dimensions are kept fixed ($N \times M$) for all frames due to several reasons. First, the rodent's body is very deformable and may change shape quickly. Accordingly, it is very difficult to predict and adapt the window size automatically and reliably. Second, the size of the window affects the speed of computation. As a result, if we vary the size of the window to test several size and scale hypotheses, the processed frame rate would be negatively affected. In addition, the window size may become unstable and grow or shrink indefinitely due to noisy structures that can be mistakenly associated with the target.

The system is initialized by manually drawing the first window around the target in the first frame. The target posture and position are irrelevant given that in the next phase the boundaries of the window will be adjusted. A large tracking window also leads to increase the processing time. Consequently, if the target occupies a large area in the first frame, it is better to select a smaller window on the body of the target, taking into consideration that the window should be large enough to include as much information about the target as possible. The window content should be representative of the target color distribution and texture and avoid background zones.

For each displacement of the sliding window, the bounded region is considered as a candidate. The target is chosen as the candidate region that minimizes a fitness cost function (S_f). The fitness cost function S_f is based on a composite strong feature that combines three weak features: the histograms of oriented gradients (*HOG*), the overlapped histograms of intensity (*OHI*), and the absence of motion A_m.

Histograms of oriented gradients (HOG)

HOG was chosen to account for texture. *HOG* is designed to be mostly invariant to the target's geometric transformations and changes in illumination. The *HOG* feature is based on the algorithm described by Dallal et al. in [27, 28]. *HOG* is calculated as follows:

1) The candidate region is divided into $n \times m$ overlapping cells and k will refer to the index of one of the cells.
2) The gradient orientation histogram h_{HOG} is computed for each cell.

$$h_{HOG}^{k}(r_a) = \sum_{p=1}^{n_a} i_p, \quad (1)$$

where r_a is an orientation interval, n_a the number of gradients that have an orientation included in r_a, and i_p the magnitude of each gradient.

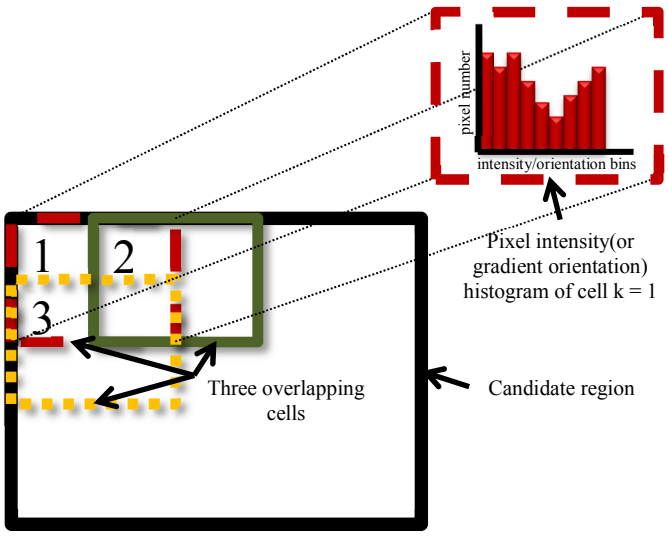


Fig. 4: *HOG* and *OHI* calculation

3) The *HOG* feature is, then, constructed as an $nm \times a$ matrix where

$$HOG(k) = \frac{h_{HOG}^{k}}{\|HOG\|}, \quad (2)$$

where h_{HOG}^{k} is the histogram of the k^{th}cell and $\|HOG\|$ is the norm of the $nm \times a$ *HOG* matrix.

Overlapped histograms of intensity (OHI)

The histogram of intensity (*HoI*) is the classical method to model a region of interest's intensity distribution. In [29], the authors used *HoI* to detect and track people in a scene. In [30], the author combined *HoI* with *HOG*, the motion history image (*MHI*) [31], the saliency likelihood map [32] and the template likelihood map [30] to detect objects in a scene. Histograms of intensity are commonly used because they are robust to change in scale and rotation. However, *HoI* fails to capture local intensity distributions. Inspired by *HOG's* structure, we propose the overlapped histograms of intensities (*OHI*s) as a compromise between *HOI* and the fine granularity provided by individual pixels, because it is calculated on small cells. *OHI* calculation is similar to *HOG's*.

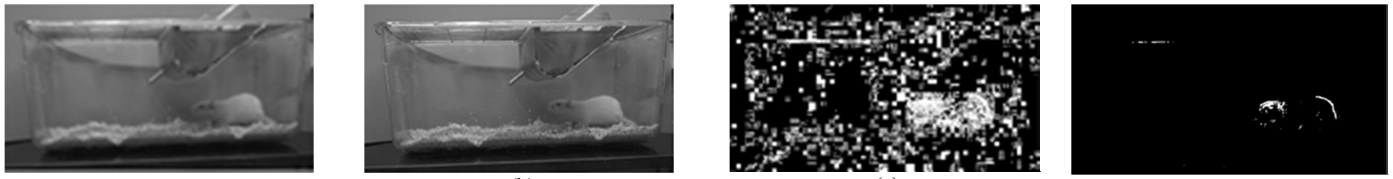
1) The candidate region is divided into $n \times m$ overlapping cells.
2) An intensity histogram of b bins is calculated for each cell as follows:

$$h^{k}(r_b) = n_b, \quad (3)$$

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.
3) We constructed the *OHI* feature matrix as a $nm \times b$ matrix where

$$OHI(k) = \frac{h^{k}}{\|OHI\|}, \quad (4)$$

h^{k} is the histogram of the k^{th}cell and $\|OHI\|$ is the norm for the $nm \times b$ *OHI* matrix. *HOG* and *OHI* are illustrated in Fig. 4.



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

Fig. 5: A_m calculation. (a) Frame t, (b) Frame $t+1$,(c) $|(\text{Frame t}) - (\text{Frame t}+1)|$, (d) The frame after thresholding (c) [1]

Absence of motion A_m

The target should be the only mobile entity in the cage. Accordingly, any detected motion is a strong indicator to the position of the target, hence the utility of using the absence of motion (A_m) in our fitness cost function.

$$D_i = \begin{cases} 1 & if\ |F_t - F_{t+1}| > \epsilon_1 \\ 0 & otherwise \end{cases}, \quad (5)$$

$$A_m = (M \times N) - \sum_{i=1}^{M \times N} D_i, \quad (6)$$

where F_t and F_{t+1} are two consecutive grayscale frames, and ϵ_1 a given threshold Fig. 5 illustrates A_m calculation.

Cost function

The fitness cost function (S_f) is calculated as follows

$$S_{f(w_i)} = \alpha_1 D_{HOG(w_i)} + \alpha_2 D_{OHI(w_i)} + \alpha_3 A_{m(w_i)} \quad (7)$$

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_{OHI} is the distance between their *OHI* feature matrices. D_{HOG} and D_{OHI} are calculated using the Euclidian distance. The Euclidean distance is used because it satisfies the requirements of the method in addition to being simple to implement. The alphas are weights that are attributed to each component of the cost function.

B. Boundary Refinement

The sliding window tracker approximates the location of the target, but its dimensions and position should be adapted to extract the target from the frame. We found that the edge map is useful to undertake the boundary refinement of the window because we can isolate many of the target's edglets. The edge map is also advantageous with respect to the color map because it is insensitive to the changes in the color distribution of the target, and because it allows for a simple way to isolate the highly textured regions. In addition, the probability of the background and target sharing similar color zones is greater than the probability of the background and the target sharing edges. This is due to the significantly smaller area that an edge covers with respect to a color zone. Consequently, in the edge map it is less likely to assign target parts to the background than in the color map.

(a) (b)

Fig. 6: Edglets (a) an edge map in a high contrast video (b) an edge map in low contrast video

Edglets

When computing the edge map, the resulting contour of the rat is not continuous. It is composed of short groups of edges that we refer to as edglets. The discontinuity of the contour is due to contrast imperfection between the rat and the background. Fig. 6 shows examples of edglets. The edglets are longer and fewer in number when the contrast is more pronounced.

The edge map contains a large number of background edglets. These edglets will cause distraction to the refinement process and result in incorrect target dimensions. To suppress these edglets, we propose a new method for background edge subtraction. Similarly to color-based background subtraction, we aim at removing from the foreground the edges (and thus edglets) that belong to the background.

Online e-background

The e-background consists of the background edglets. In fact, an e-background could be constructed in the same manner as a color-based background. Nevertheless, instead of using the repetition of the same pixel's intensity to include it in the background, we consider its edglet occurrence repetition to include it in the background.

Given eB_{t-1} the online e-background calculated at time $t-1$, eC_t the edgelet map calculated at time t, and eT_t the edglets in eC_t that are included in the target's bounding box, the online e-background at time t is calculated as follows:

$$eB_t = \alpha \times (eC_t - eT_t) + (1-\alpha) \times eB_{t-1} \quad (8)$$

For this operation, the target coordinates are taken as the ones calculated at $t-1$. eC_t is calculated using Canny's algorithm [33] applied on the gray scale frame. The OpenCV libraries were used to compute gray scale transformation and the edge map using Canny's algorithm.

The e-background and the result of the e-background subtraction are illustrated in Fig. 7. Notice in Fig. 7(b), the edglets that are constant in the e-background are represented with higher values (lighter colors) due to accumulation.

However, because the edglets in highly textured regions with weaker magnitude have a tendency to shift position because of noise from one video frame to the next, considering only the (x,y) coordinates to detect foreground edglets does not give satisfying results. Thus, we must account for the slight edglet shifts that may occur between frames in high edglet density background regions. For this purpose, we use an

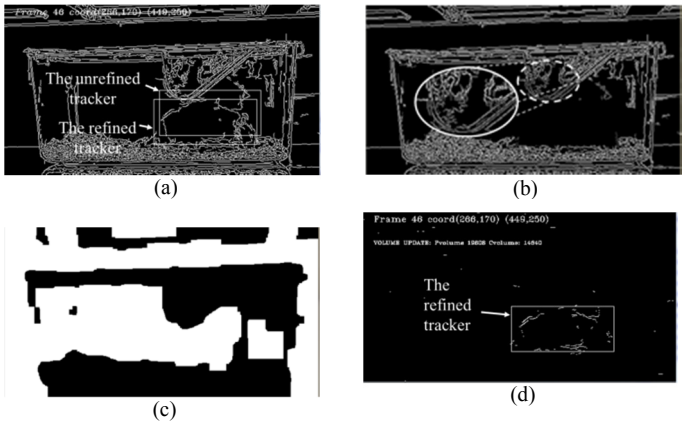


Fig. 7: Online e-background subtraction. (a) an edge map. (b) the constructed e-background. (c) the map of edge high density. (d) the foreground edglet map.

operator, D_{He}, to evaluate the edglets density in the e-background. In other words, the edglets density is calculated by dividing the edge map into a grid of $\eta \times \eta$ pixels squares. Given $Ne(l)$ as the number of edglet pixels in the l^{th} square of the grid, the edglets density, in the l^{th} square is calculated as:

$$D_{He}(l) = \begin{cases} 0 & if\ N_e(l) > \epsilon_2 \\ 1 & otherwise \end{cases}, \quad (9)$$

Fig. 7(b and c) shows, respectively, a frame edge representation, and the result of calculating D_{He}.

Finally, the target edglets eT are constructed from all the pixels that are excluded of eB and D_{He}

$$eT(i_e) = \begin{cases} 1 & if\ (\overline{eB_{t-1}(i_e)} \times eC_t(i_e)) > \epsilon_3\ and\ D_{He}(i_e) = 1 \\ 0 & otherwise \end{cases}, \quad (10)$$

where i_e is a pixel in the edglets map, $\overline{eB_{t-1}(i_e)}$ is the complement of the background and $D_{He}(i_e)$ is edglet density of the square to which that pixel belongs.

Final Localization of the Animal

Using the foreground edglets, we now reconstruct the regions corresponding to the animal. To do that, the vicinity of the tracking window is scanned to produce pulse graphs in the edge map (Fig. 8(b)). The horizontal pulse graph P_h is constructed by scanning the region horizontally.

$$P_h(x) = \begin{cases} 1 & if\ \sum_{y=y1}^{y2} F(x,y) > 0 \\ 0 & otherwise \end{cases}, \quad (11)$$

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

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

$$P_v(y) = \begin{cases} 1 & if\ \sum_{x=x1}^{x2} F(x,y) > 0 \\ 0 & otherwise \end{cases}, \quad (12)$$

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

We assume that the rat's edglets are close to each other; consequently their corresponding pulses are close to each other as well. Those pulses are merged to assemble the complete target entity coverage. This procedure is added because the contour of the target is seldom completely continuous. Its objective is to merge its constituent edglets.

The merge process of positive pulses, for the horizontal pulse series (P_{h_M}) is described in (13). Short zero pulses are detected and converted to positive pulses:

$$P_{h_M}(x) = \begin{cases} 1 & if\ P_h(x) = 1 \\ 1 & if\ d < \epsilon_4\ and\ P_h(x, x+d) = 0, \\ 0 & otherwise \end{cases} \quad (13)$$

where d is the length of a zero pulse and $P_h(x, x+d)$ is a pulse that has a magnitude of zero and extends between the coordinates x and $x+d$. In other words, a point on the pulse graph is assigned a magnitude of 1 if it belongs to a pulse of magnitude 1 or if it belongs to a pulse of magnitude 0 which has a width $d < \epsilon_4$. The same procedure is applied to the vertical pulse series.

Afterwards, the resulting pulses are projected on the frame, and the region R_t^{max} that maximizes the intersection of two of those projections is chosen to set the boundaries of the tracker according to the following condition

$$T_t = \begin{cases} R_{max} & if\ R_{max} > \beta T_{t-1} \\ T_{t-1} & otherwise \end{cases}, \quad (14)$$

where

$$R_{max} = max(Proj(P_{h_M}(i)) \cap Proj(P_{v_M}(j))), \quad (15)$$

$Proj\left(P_{h_M}(i)\right)$ and $Proj\left(P_{v_M}(j)\right)$ are the projections of two piecewise constant functions that belong to P_{h_M} and P_{v_M}, respectively. T_t and T_{t-1} are the bounding boxes at time t and $t-1$ respectively .

We also assume that the target cannot have a drastic change of area between two consecutive frames. This is why an area change constraint is applied in (14).

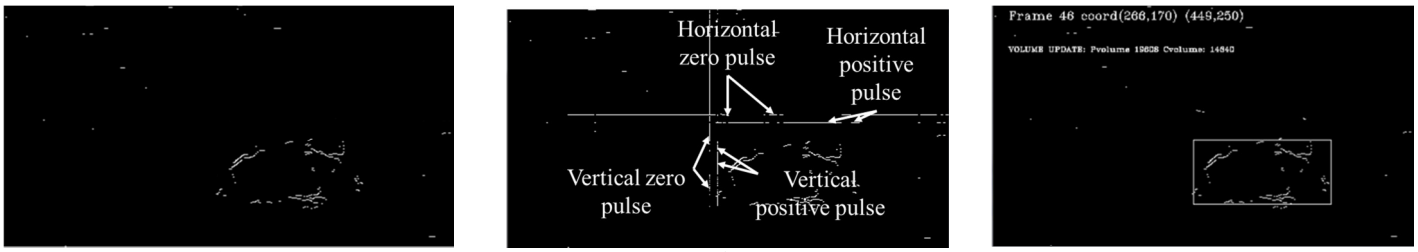
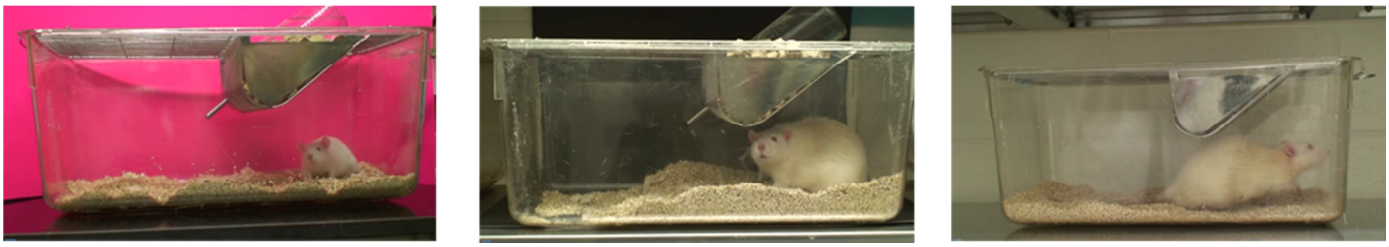


Fig. 8: Boundary Refinement. (a) an edge map after e-background subtraction. (b) The calculated pulses. (c) The tracking window after refinement.



(a) (b) (c)

Fig. 9: Snapshots from (a) video 1, (b) video 2, and (c) video 3

V. Experimental Results

To test the proposed methodology, videos of rats in cages were recorded at the research center in Sainte-Justine Children Hospital. The camera set up and data acquisition did not disturb the ongoing experiments, or the actual environment conditions. The cages were set on shelves and illumination was provided by florescent lamps from the ceiling. The cages had the same dimensions, were transparent, and in some of the cases, a pink or a black cardboard was placed behind the cage. The introduction of the colored cardboard is for the purpose of testing a variety of backgrounds. The information about the experimental settings is summarized in Table I. For video 3, no cardboard was added while for video 1 and video 2, a pink cardboard and a black cardboard were added, respectively (Fig. 9). The rats were white and of different size classes. The real dimensions of the rat were not recorded. We measured the maximum pixel length recoded for each rat and normalized it with the pixel width of the frame according to (16)

$$Relative\ rat\ size = \frac{\max(rat\ pixel\ length)}{(cagewidth_{min}+cagewidth_{max})}, \quad (16)$$

Although this is not an accurate method to measure the dimensions of the target, still, it illustrates the size classes of the targets.

TABLE I. Video information

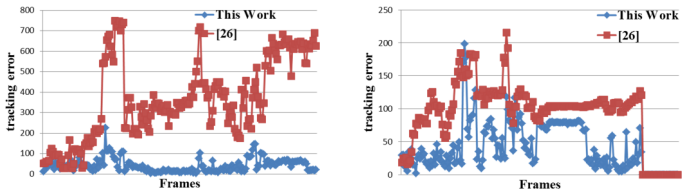
	Video 1	Video 2	Video 3
Width	560		
Height	304		
Frame Number	8235	15279	8835
Frame rate	25 frames/second		
Animal color	white	white	white
Background color	pink	black	environment wall (white)
Relative rat size	0.58	0.95	0.82

Table II summarizes experimental parameters. Even though the parameters are determined empirically, they appear not to be dependent on the size of the rat, nor the colors involved, given the extent of the experiments. In fact, the same parameters gave the optimal results when used for the three videos. The parameters were chosen after extensive testing on Video 1. However, different variations did not improve the results on the other two videos. Essentially, the parameters were set to achieve a reasonable balance and compromise. For instance, the values obtained for A_m (the absence of motion) were two orders of magnitude larger than the values obtained for D_{HOG} or D_{OHI} (the distances calculated for the histograms of motion and the overlapped histograms of intensity). Accordingly, the values of α_1, α_2 and α_3 were initially set to balance the contributions of the three features. The size of the edge map grid cells ($\eta \times \eta$ pixels squares) was chosen as a granularity compromise. The cells should be small enough to insure a fine granularity and large enough to contain sufficient information. The corresponding threshold ϵ_2 was obtained by training. Samples were collected on high edge density regions and their average was computed. ϵ_1 is set as a compromise between removing as many noise pixels as possible while preserving the motion pixels. Similarly, ϵ_3 is set as a compromise between removing as many noise pixels as possible while preserving foreground pixels. ϵ_4 is set after observation of the width that separates edglets on the target contour. ϵ_4 should not be too high so that noise edglets are combined with the target. We do not expect the target area to change significantly between two consecutive frames. Therefore, we restricted the change of the area by the factor β.

For *HOG* and *OHI* calculations, we used the values suggested in [27] for the cell's dimensions and the number of histogram bins.

TABLE II. Experimental parameters

parameters	Equation involved	values
$(\boldsymbol{\alpha_1, \alpha_2, \alpha_3})$	Parameters used in the cost function calculation (7)	(1,1, 0.01)
$\boldsymbol{\alpha}$	e-background update factor (8)	0.4
$\boldsymbol{\eta \times \eta}$	edge map grid dimensions for edge high density calculation	15×15
$\boldsymbol{\epsilon_1}$	Threshold used in motion extraction (5)	50
$\boldsymbol{\epsilon_2}$	Threshold used in edge high density calculation (9)	5
$\boldsymbol{\epsilon_3}$	Threshold used in e-background subtraction (10)	0.5
$\boldsymbol{\epsilon_4}$	Maximum zero pulse width (13)	20
$\boldsymbol{\beta}$	Parameter used for area adaptation restriction (14)	1.7



(a) (b)

Fig. 10: Tracking results from [26] and this work. (a) video 1. (b) video 2.

Feature Validation

To evaluate and to validate the three features (*HOG*, *HI*, A_m) in (7), we considered two strategies. The first one is to isolate each feature and to measure its contribution to the tracking result. The second one is to measure the effects of its absence on the tracking result. For both strategies, only the tracking part of the algorithm was considered (section IV-A).

The three videos were tested. For each execution, 200 frames were selected randomly for evaluation. The ground truth was selected as the center of the animal. We performed a manual segmentation to extract the target in each frame. Then, the center of the animal was calculated as the center of its bounding box. The same frames were used in each video for all combination tested. The calculated error, err_{center} (17) is the error between the ground truth center position and the tracker's center position normalized by the tracker's window size.

$$err_{center} = \sqrt{\left(\frac{x_{gt}-x_t}{tracker\ width}\right)^2 + \left(\frac{y_{gt}-y_t}{tracker\ height}\right)^2} \times 100, \quad (17)$$

where, (x_t, y_t) are the coordinates of the tracker center, and (x_{gt}, y_{gt}) are the coordinates ground truth center.

The mean and standard deviation (std) of the error, in pixels, are calculated and displayed in Table III. For video 3, the combination of "*OHI* and A_m" gave the best result. In fact, it has a minor advantage on the combination of the three features. However, for video 1, the combination of "*OHI* and A_m" did not match the performance of the combination of the three features and was even outperformed by the "*HOG* and A_m"combination. For video 2, the combination "*HOG* and A_m" gave the best results but not far from the result that corresponds to the three features. Finally, no single feature or combination of two features was dominant in general. The combination of the three features is the most stable and is necessary to insure good tracking in most cases. It is important to note that this is not the final error. It is the tracking error which is further reduced in the boundary refinement phase.

Comparison with the state-of-the-art

To compare our complete method with the state-of-the-art, we have selected the method described in [26], which is similar to a method previously applied to rodents [14], and for which source code was provided. The algorithm described in [26] is meant to extract the exact contour of the target while the proposed algorithm draws a bounding box around the target. To compare both algorithms, we considered the bounding box around the contour calculated using [26].

To objectively evaluate the quality of the complete algorithm, the ground truth set was built by drawing a manual bounding box around the rodent at each of 200 randomly selected frames.

TABLE III. FEATURE EVALUATION(%)

Features	Video 1		Video 2		Video 3	
	Mean error	std	Mean error	std	Mean error	std
HOG	107.05	50.62	66.30	26.46	62.27	37.89
OHI	49.32	49.32	54.52	22.57	63.59	44.16
A_m	61.68	20.62	81.87	19.18	58.93	23.14
HOG and *OHI*	41.43	28.32	66.78	21.88	58.48	39.07
HOG and A_m	43.97	43.97	41.78	17.90	49.80	44.98
OHI and A_m	43.89	43.89	42.67	15.88	18.98	10.21
All features	36.90	29.12	42.09	14.55	22.90	11.044

The results of the percentage coverage area error, for the algorithms, are shown in Fig. 10 while the means of the results are summarized in Table IV. The same frames were used to evaluate both algorithms. The mean coverage area is calculated as follows:

$$err_{area} = \frac{(Area_T \cup Area_{GT}) - 2\times(Area_T \cap Area_{GT})}{Area_{GT}} \times 100, \quad (18)$$

Where $Area_T$ is the area covered by the tracker and $Area_{GT}$ is the area covered by the ground truth.

Our algorithm's clear advantage over the algorithm described in [26] is mainly due to the first step (the tracker). The tracker does not only rely on colors and gradients alone to track the animal, it also exploits motion information. Thus, the tracker steers the boundary refinement step. This keeps the boundary refinement module from being trapped by noise edges. In comparison, the tracker in [26] cannot recover if it makes tracking errors when distracted by nearby noise. Another reason for the difference in performance is the e-background subtraction which reduces the distraction to the algorithm caused by background edges.

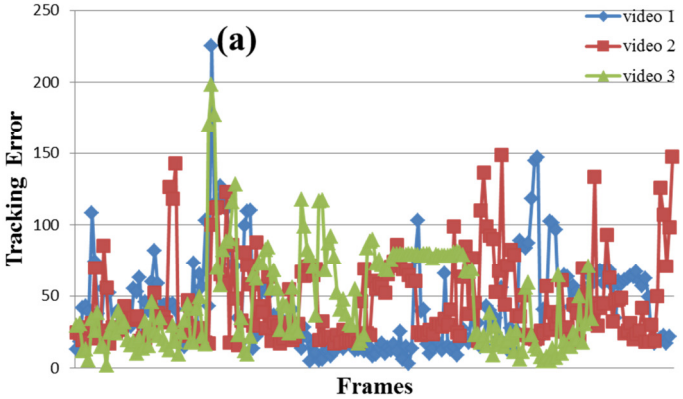
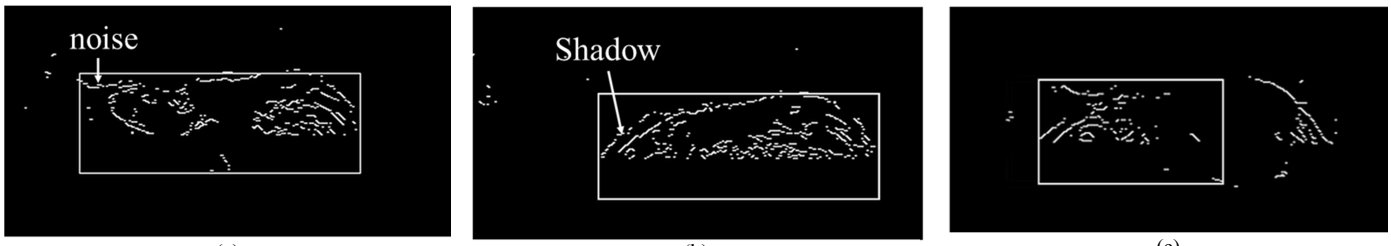


Fig. 11: The calculated error for the randomly selected frames.



(a) (b) (c)

Fig. 12: source of error in boundary refinement. (a) error cause by nearby noise in the e-background. (b) error caused by shadows. (c) error caused by far target edglets.

When considering the performance of our algorithm for the three videos, the advantage in video 1 is due to the pink background. The pink background ensures better contrast between the target and the background and produced fewer reflections. In video 2, the black background produced a mirror effect that highly increased the creation of reflections. The reflections are the main source of noise. This noise is difficult to suppress even with e-background subtraction. In video 3, the contrast between the background and the target is low. Therefore, the edglets created are shorter and more discontinuous. This increases the probability of giving a smaller target apparent size when two consequent edglets are more than 20 pixels apart.

Fig. 11 shows the calculated error for every ground-truth frame for the proposed algorithm. The frames are sorted chronologically. Fig. 11 shows some cases where the error goes above the average. This error is mainly due to the discontinuity in the rat's edglets, which splits the rat in several parts during boundary refinement. Fig. 12(c) illustrates this case. The edglet on the upper boundary of the rat is split in two. These two parts are separated by more than 20 pixels and are not joined together during the refinement stage. Consequently, the rat will be represented by two separate pulses in the horizontal pulse graph, and only one of them will be chosen to represent the width of the rat.

TABLE IV. PERCENTAGE COVERAGE AREA ERROR

	Video 1	Video 2	Video 3
[26]	366.14	Failed	106.97
This work	42.98	46.61	47.80

Shadows are another cause of error. Shadows do not belong to the background so they cannot be suppressed while doing the e-background subtraction. The effect of shadows is illustrated in Fig. 12(b). In that situation, the shadow produces a larger apparent target size. Shadow detection algorithms could be used to minimize their effects on the tracking process.

Noise edglets that are too close to the rat are another source of error. For example, in Fig. 12(a), some noise edglets, that were not removed by the e-background subtraction are close to the upper boundary of the rat. These edglets were considered as part of the rat and resulted in a larger apparent height.

Despite these shortcomings, the method's performance is consistent under all the conditions tested, as shown in Table IV. The method also proved to be efficient in uncontrolled hard settings. In video 3, even though both the rat and the background were white, the method still performed with a mean error less than 48%. Another advantage of the proposed method is that even when the tracker is subject to error, the error does not have a permanent effect and the algorithm can recover at any time. In Fig. 11, we see that even though the tracker had a big error at point (a), it was able to recover later and reduce the error to zero. Fig. 13 shows extraction results for the three videos.

Table V shows results for initializing the tracker with different postures and positions of the target. The tracker was initialized at different frames for the same video, and the same frames were used to calculate the error. Fig. 14 shows the different postures and positions. The results show that the tracker is insensitive to the initialization window.

VI. CONCLUSION

In this paper, we proposed a robust method that tracks a rodent in a cage under uncontrolled conditions. The method is formed of two steps. In the first step, three weak features are used to coarsely track the target using a sliding window approach. Step two considers the frame edglets maps to adjust the limits of the tracker to the boundaries of the target. The method uses e-background subtraction to extract the target edglets. Pulses are constructed using the remaining edglets and are used to adjust the tracking window. The method was tested under representative biomedical environment conditions. The method's performance is consistent when applied to three videos exhibiting different backgrounds and rat sizes.

TABLE V. PERCENTAGE POSITION ERROR FOR DIFFERENT INITIALISATION (VIDEO 1)

Video 1	Frame 1	Frame 180	Frame 530
	42.98	66.01	43.75
Video 2	Frame 1	Frame 1250	
	46.61	46.81	
Video 3	Frame 1	Frame 180	Frame 230
	47.80	43.19	60.29

ACKNOWLEDGMENT

The authors thank Dr. Sébastien Desgent for his assistance and support.

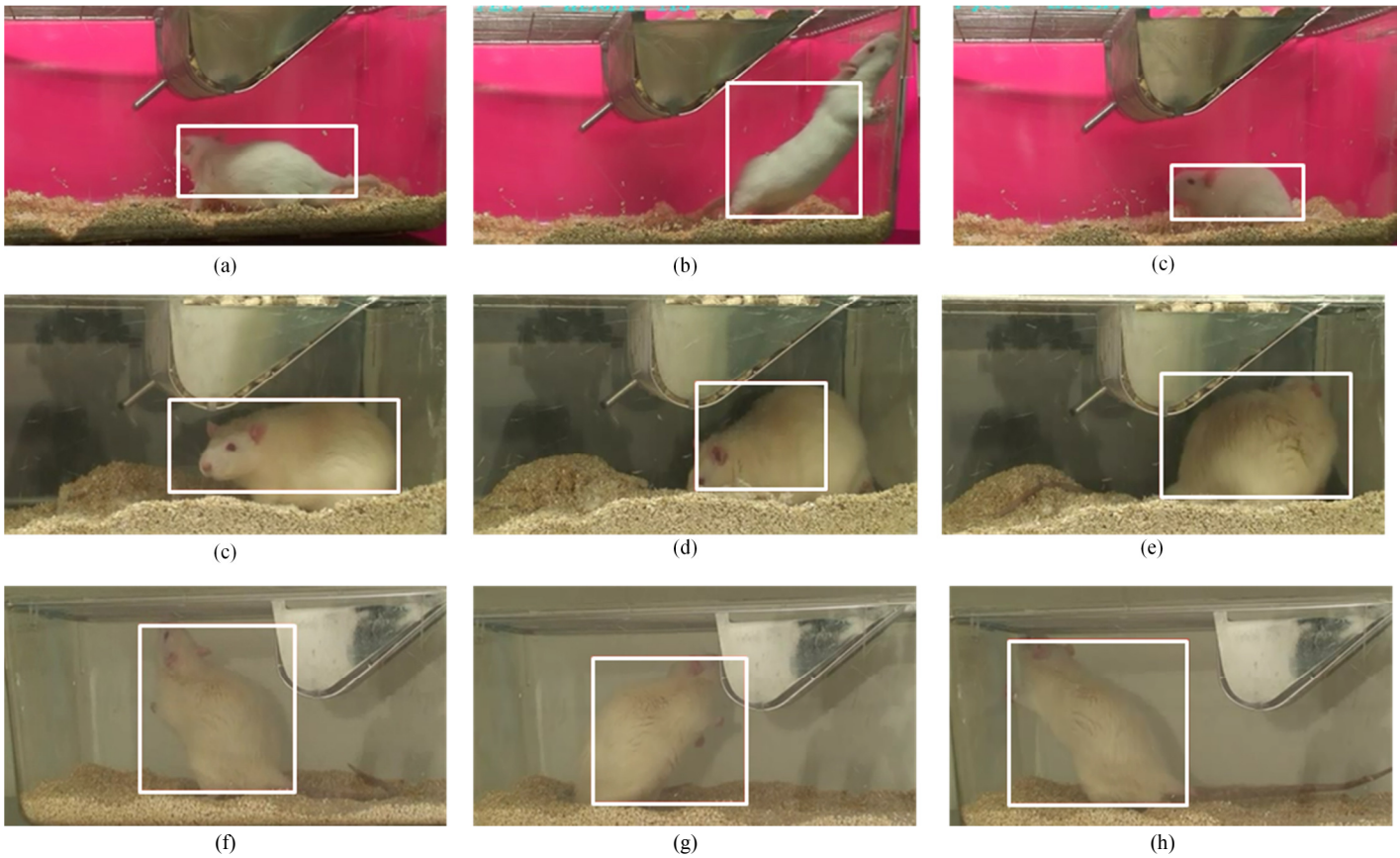


Fig. 13: Tracking results. (a,b,c) video 1. (a) the extraction error is caused by nearby noise to the rat. (b) the error is caused by groups of edglets that are too far away. (c) successful target extraction. (d,e,f) video 2. (d) the shadow causes a larger target apparent size. (e) error caused by two widely separated pulses. (f) successful target extraction. (g, h, i) video 3. (g, h) error caused by nearby noise. (h) successful target extraction.

REFERENCES

[1] R. Farah, J.M.P. Langlois, G.A. Bilodeau, "RAT: Robust Animal Tracking," International Symposium on Robotic and Sensors Environment, pp. 65-71, 2011.

[2] 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.

[3] 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.

[4] Y. Nie, T. Takaki, H. Ishii, H. Matsuda, "Behavior Recognition in Laboratory Mice Using HFR Video Analysis," IEEE International Conference on Robotics and Automation, pp. 1595-1600, 2011.

[5] 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.

[6] 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.

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

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

[9] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A.D. Steele, T. Serre, "Automated home-cage behavioral phenotyping of mice," Nature Communication, 2010.

[10] H. Jhuang, E. Garrote, N. Edelman, T. Poggio, A. Steele, T. Serre, "Trainable, vision-based automated home cage behavioral phenotyping," International Conference on Methods and Techniques in Behavioral Research, 2010.

[11] H. Jhuang, E. Garrote, T. Poggio, A. Steele, T. Serre, "Vision-based automated recognition of mice home-cage behaviors," ICPR, 2010.

[12] "Vision-Based System for Automated Mouse Behavior Recognition," http://cbcl.mit.edu/software-datasets/mouse, July 25, 2011 [August 8 2011].

[13] P. Dollar, Z. Tu, and S. Belongie, "Supervised Learning of Edges and Object Boundaries," CVPR, pp 1964-1971, 2006.

[14] K. Branson, S. Belongie, "Tracking Multiple Mouse Contours (without Too Many Samples)," CVPR, vol. 1, pp.451-457, 2005.

[15] D. Martin, C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using loacl brightness, color, and texture cues," PAMI, vol. 26(5) pp. 530-549, 2004

[16] "EthoVision XT." http://www.noldus.com/animal-behavior-research/products/ethovision-xt, [August 8 2011].

[17] "Computer, video, and data acquisition systems." http://www.noldus.com/animal-behavior-research/accessories/computer-video-and-daq-systems, [August 8 2011].

[18] S.L. Wilks, W. Wolf, Y. Liang, V. Kobla, X. Bais, Y. Zhang, L. S. Crnic, "Unified system and method for animal behavior characterization from top view using video analysis," U.S. Patent 7 643 655, Jan 5, 2004

[19] Y. Liang, V. Kobla, X. Bais, Y. Zhang, "Unified system and method for animal characterization in home cages using video analysis," U.S. Patent 7 209 588, April 24, 2007.

[20] "Home Cage Environement," http://www.cleversysinc.com/products/hardware/home-cage-environment, [October 4 2011]

[21] J. Shi and C. Tomasi, "Good Features to Track," CVPR, pp. 593-600, 1994.

[22] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.

[23] G.R. Bradski, "Computer video face tracking for use in a perceptual user interface," Intel Technology Journal, Q2, 1998.

[24] C. Stauffer, W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking", pp. 246-252, CVPR 1999.

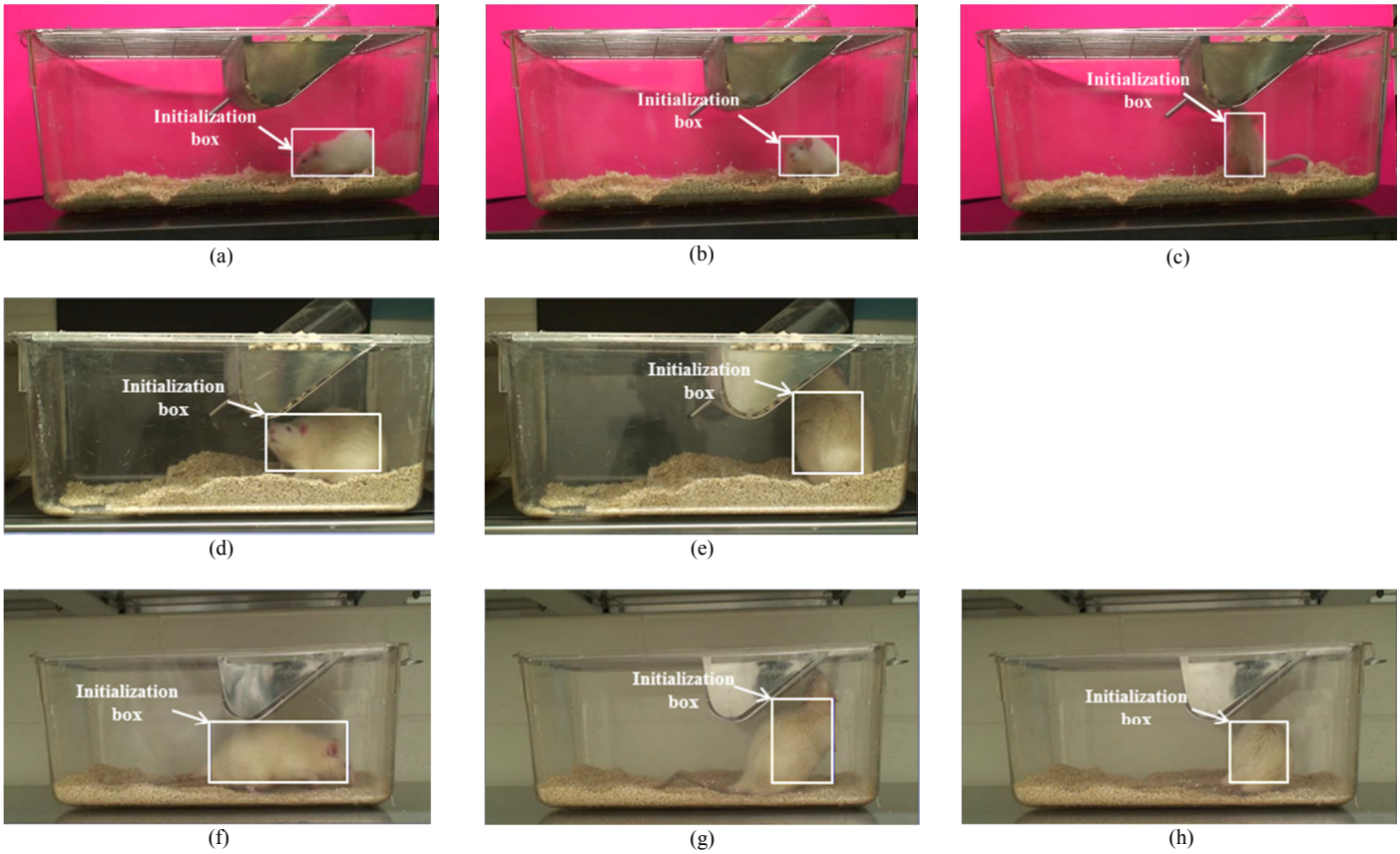


Fig. 14: Diffirent initialization instances: (a) Video1 - Frame 1, (b) Video 1 - Frame 180, (c) Video 1 - Frame 530, (d) Video 2 – Frame 1, (e) Video 2 – Frame 1250, (f) Video 3 – Frame 1, (g) Video 3 – Frame 180, (h), Video 3 – Frame 530.

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

[26] N. Vaswani, Y. Rathi, A. Yezzi, A. Tannenbaum, "PF-MT with an Interpolation Effective Basis for Tracking Local Contour Deformations," IEEE Trans. Image Processing, vol. 19 (4), 841–857, 2010.

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

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

[29] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25(5), pp. 564-577, May 2003.

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

[31] Z. Yin and R. Collins, "Moving Object Localization in Thermal Imagery by Forward-backward MHI," Conference on Computer Vision and Pattern Recognition Workshop, pp. 133, 2006.

[32] X. Hou and L. Zhang, "Saliency Detection: a Spectral Residual Approach," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR07), pp. 1-8, June 2007.

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

Rana Farah (S'00) received a B.E. in electrical engineering and an M.S. in computer science from the Lebanese American University in Lebanon, in 2003 and 2006 respectively. Rana is currently enrolled in a Ph.D. in computer vision program at the École Polytechnique de Montréal.

She worked for a year as a system administrator at the Lebanese American University, in 2008. Her research interests involve System on Ships related algorithms and computer vision applications.

J.M. Pierre Langlois (S'00-M'03) was born in Sherbrooke, Québec, in 1967. He received the B.Eng. degree in Electrical Engineering and the M.Eng. and Ph.D. degrees in Computer Engineering from the Royal Military College of Canada (RMC) in 1990, 1999 and 2002, respectively.

He served as engineering officer in the Canadian Navy from 1990 until 1997. He was an Assistant Professor in the Department of Electrical and Computer Engineering of RMC from 2000 until 2005. Since 2005, he has been an Associate Professor in the Department of Computer and Software Engineering of École Polytechnique de Montréal.

Dr. Langlois is a member of the Regroupement Stratégique en Microsystèmes du Québec and of the Ordre des Ingénieurs du Québec. His research interests center on the embedded implementation of DSP and video processing algorithms. Some of his current projects focus on configurable processors with applications in computer vision, security and indoor navigation. He is author or co-author of 50 journal and refereed conference papers.

Guillaume-Alexandre Bilodeau (M'10) received the B.Sc.A. degree in computer engineering and the Ph.D. degree in electrical engineering from Université Laval in Canada, in 1997 and 2004, respectively.

In 2004, he was appointed Assistant Professor at École Polytechnique de Montréal, Canada. He is currently an Associate Professor at the same institution. His research interests encompass image and video processing, video surveillance, object recognition, content-based image retrieval, and medical applications of computer vision.

Dr. Bilodeau is a member of the Province of Quebec's Association of Professional Engineers (OIQ).