Mining Motion Patterns using Color Motion Map Clustering

Chih Lai, Taras Rafa, Dwight E. Nelson*
Graduate Program in Software Engineering, Department of Biology*
University of St. Thomas
St. Paul, MN 55105
{clai, trafa, denelson*}@stthomas.edu

ABSTRACT

Automatically extracting previously unknown behavior patterns from videos that track animals with various physical conditions can accelerate our understanding of animal behaviors and their influential factors, resulting in major medical and economic benefits. Unfortunately, extracting behavior patterns from videos recordings remains as a very challenging task due to their extensive duration and the unstructured natures. This task is further complicated in a completely darken animal cage with inconsistent infrared lighting, moving reflections, or other cage debris such as the cage bedding. In this research, we propose a new motion model that enables us to measure the similarities among different animal movements in high precision so a clustering method can correctly separate recurring movements from infrequent random movements. More specifically, our model first transforms the spatial and temporal features of animal movements into a sequence of color images, referred to as *color motion maps* (*CMMs*). The task of mining recurring behavior patterns is then reduced to clustering similar color images in a database. We will use a real infrared video to demonstrate the capability of our model in capturing distinguished but brief animal movements that are embedded within a sequence of other animal movements.

Keywords

Binary motion map (BMM), color motion map (CMM), color autocorrelogram.

1. INTRODUCTION

Observations of lab animal behaviors have been historically used to understand animals and their interactions with environment. While labor and time intensive, behavioral observation has been employed in studying adverse effects of new drugs on brain functions, learning and memory, depression, and other neurological disorders [1], [4], [5], [7].

Electronic monitoring of animals offers an alternative in screening different animal activities. For example, a sensor or switch may be attached to sensatory devices such as running wheel, water bottle or cage floor to monitor running, drinking or walking activities. However, since sensors can only detect the presence of animals in a specific cage area, many previously unrecognized behaviors can remain undetected under this approach.

With advances in digital image technology, surveillance cameras can be easily mounted on an animal cage to indiscriminately and continuously track all movements that an animal might make. Unfortunately, current approaches rely heavily on location-specific knowledge, not animal movements, in extracting predefined activities from videos [4], [7]. For example, biologists must manually specify which artificial region of an animal cage corresponds to what activity so an analysis program can label each video image with a predefined animal activity based on animal's location in the image. As a result, errant and fluctuating activities may be generated by the program whenever animals cross the boundary of artificial areas which may vary in each cage. Hence, automated detection of animal activity remains as a very difficult task.

Our goal in this research is to develop an efficient approach that can recognize subtle animal activities that could not be obtained before. In order to achieve this goal, we need a new model that not only can effectively capture the spatial and temporal features of animal movements, but also can be efficiently transformed into quantitative motion descriptors for further mining processes. Additionally, this new model must be robust and flexible enough to handle the noisy videos that are contaminated by inconsistent infrared lighting in an otherwise dark cage, moving reflections or shadow cast by animals, or other debris clutters such as the cage bedding.

Under our approach, the bright intensity of the HSI (*H*ue-*S*aturation-*I*ntensity) color space is first extracted from each video image and converted into a black-and-white image because there is no color information in an infrared video. The spatial movement cast by animals over two consecutive video frames can then be easily obtained by using an XOR-operation. We refer the output produced by each XOR-operation as a *binary motion map* (*BMM*) because it reflects the spatial movement in a binary format. If we stack *w* number of *BMM*s together, each element of the stacked *BMM*s will contain *w* binary numbers that can be treated a color. As a result, the stacked *BMM*s become a color image, or a *color motion map* (*CMM*), that contains both the spatial and temporal information about the body movements in a duration *w*. Next, the spatial correlation of colors in each *CMM* can then be measured as a numeric vector which becomes the signature of each *CMM*.

In other words, the spatial and temporal features of animal movements within a temporal period are first captured as a color image. The features embedded in each color image are then transformed into a numeric vector by measuring the spatial correlation of colors in the color image. As a result, the task of mining recurring behavior patterns can be reduced to the task of clustering similar color images in a database. We will use a real infrared video to demonstrate the capability of our model in

capturing distinguished but brief animal movements that are embedded within a sequence of other animal movements.

This paper is organized in the following sections. Section 2 reviews related works. Section 3 discusses in detail the proposed motion model. The time and space complexity of our model are also analyzed in this section. Section 4 explains the procedures in clustering motion patterns and creating meaningful description for each cluster. We present experiment results in Section 5 and demonstrate the capability of our model in capturing distinguished but brief animal movements that are embedded within a sequence of other animal movements. Section 6 concludes this paper.

2. Related Work

Using computer vision technology in tracking animal movements can serve as a powerful tool for monitoring sentinel cages in potential bioterrorism targets and chemical agent research facilities. Authors in [4] used top-mounted cameras to monitor fishes in a fish tank that is contaminated by a toxic agent *MS222*. The goal of this work is to study the behavioral alterations relevant for ecotoxicological assays. This study compared the velocity, total distance traveled, space utilization of fishes in the contaminated and uncontaminated fish tanks. However, the approach used in [4] relied on the location-specific knowledge of the fish tank because it divided each fish tank into several artificial areas for tracking fish movement.

Authors from the Stanford University and Monterey Bay Aquarium used underwater cameras to study jelly fish [8]. The goal of this project is to understand the body movement of jelly fish so lead information can be generated to guide the underwater cameras in following the target jelly fish. Few existing types of jelly fish motion are first defined as motion states and the relationships between these motion states are then connected as a *finite state machine* (*FSM*). Hence, this work is built based on predefined motion knowledge of jelly fishes. An algorithm was constructed to recognize jelly fish movement by matching real-time video frames against the predefined motion states.

There are two unique features in the study discussed in [2]. First feature is that a camera is used to track multiple mice in a single cage. The second feature is that the camera is placed on the side of a cage, not mounted on the top of a cage as in most of studies, to capture the side view of animals. Authors showed that their algorithm, based on affine transformation, can create an approximate blob that continuously follows individual mouse. An algorithm is developed to classify the shape features of each blob into one of the four well-defined mouse activities such as moving and drinking.

Authors in [6] designed a system which can separate normal motions from abnormal ones in a video. Under their approach, the amount of motions in a video frame is first computed as a 2-*D* motion matrix by taking the color differences between this video frame and a pre-selected background frame. The total motion of a video frame is then computed by summing the motion in the 2-*D* motion matrix. The locality of motion is also computed by summing the values in the individual rows and columns of the 2-*D* motion matrix. A clustering method is then used to cluster these features into different activities. Unfortunately, changes in cage bedding made by animals can be falsely identified as a movement if motion information is obtained by comparing video frames to a pre-selected background frame. Moreover, although certain motion locality can be obtained by summing the motions in rows and columns of a video frame, detailed body movement is lost in this approximate computation.

3. Motion Model and Motion Descriptors

As described in Section 1 an effective motion descriptor that can closely characterize the spatial and temporal features of animal movements is needed so a clustering method can detect frequently recurred animal movements from the descriptors extracted from a video. We first explain the underlying concepts of our motion model in Section 3.1. The method in extracting the motion descriptors from the proposed model is discussed in Section 3.2.

3.1 Proposed Motion Model—Binary and Color Motion Maps

Under our approach, the bright intensity (of the HSI color space) is first extracted from each video frame image because there is no color information in an infrared video. A threshold is then applied to convert the brightness values in each video frame into a black-and-white image. Note that although the selection of different thresholds may result in different black-and-white images with different distributions of noises, the impact of noises can be greatly reduced by our motion model as we will explain later.

The spatial feature of animal movement can then be easily captured by XOR-operations which compare the pixel-to-pixel differences between two consecutive frames. We refer the output produced this operation as a *binary motion map* (*BMM*) because it represents the animal's spatial movement over two consecutive time units in a binary format.

Formally, each pixel at the r^{th} row and the c^{th} column of a *BMM* constructed from the two consecutive black-and-white images I_t and I_{t+1} is denoted as

$$BMM_t(r, c) = I_t(r, c) \oplus I_{t+1}(r, c)$$

We use the following figures to illustrate this operation. Let Figure 1 be a sequence of six-second binary (black-and-white) video frames that are converted from an original infrared video. To simplify our discussion, we focus our attention only on the pixels that are in the upper-right and lower-left corners of each video frame.

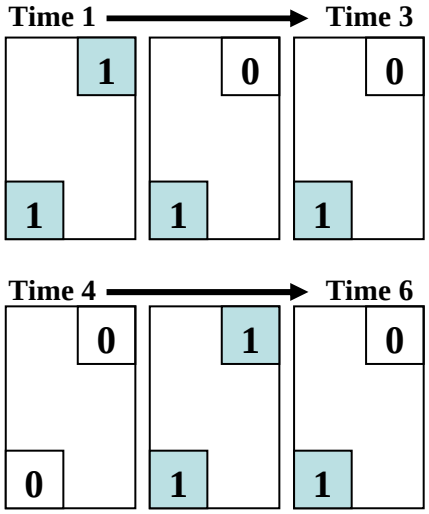

Figure 1. Consecutive video frames over six time units.

If we apply the XOR-operation to every consecutive pair of images in Figure 1, we obtain a sequence of *BMMs* shown in Figure 2. Again, we show only on the pixels that are in the upper-right and lower-left corners of each *BMM* in Figure 2.

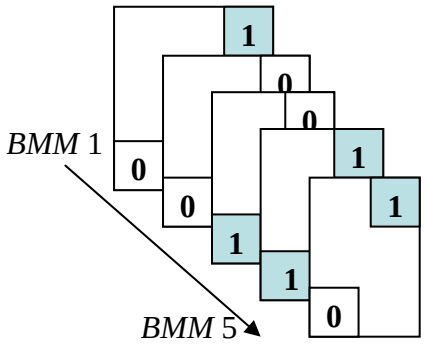

Figure 2. Binary Motion Maps (*BMMs*) computed from Figure 1.

If we stack w ($w \geq 1$) number of *BMM*s together, the stacked *BMM*s can now be treated as a color image, or a *color motion map* (*CMM*), that contains both the spatial and temporal information about the body movements in a duration w. Moreover, we will later prove our hypothesis through a series of experiments that different movements will result in different color images (*CMM*s) and similar movements will result in similar color images (*CMM*s). That is, after constructing *CMM*s from a video, we can reduce the task of mining recurring behavior patterns to the task of clustering similar color images in a database as we will show in Section 4.

Formally, if we stack w number of *BMM*s (i.e. BMM_{t-w}, BMM_{t-w+1}, …, BMM_{t-1}) together at time t ($t > w$), each pixel at the r^{th} row and the c^{th} column of a CMM_t is denoted as

$$CMM_t(r, c) = \sum_{i=1}^{w} BMM_{t-i}(r, c) \times 2^{(i-1)}$$

For the first few frames in a video where $t \leq w$, $CMM_t(r, c) = 0$ (i.e. white color, no motion). Note that the $2^{(i-1)}$ computation in the above stacking formula can be efficiently implemented by a sequence of bit-shift operations. Similarly, the $\sum$ computation in the above formula can also be easily implemented by a sequence of bit-OR operations. As a result, each pixel in a *CMM* must be in the range of 0 to 2^w–1, where w is the number of *BMMs* being stacked together.

For example, after stacking together the upper-right pixels of *BMMs* in Figure 2, the stacked binary number is 10011, which can be interpreted as a decimal unsigned integer 19. This unsigned integer can serve as an index that references to a specific color (i.e. green) defined in a customized color map shown at the right edge of Figure 3. Similarly, after stacking together the bottom-left corner pixels of *BMMs* in Figure 2, the binary number is 00110, or an unsigned integer 6, which references to a pink color in the color map. Note that a color map can be created such that a lower index value (i.e. lighter color) indicates a more recent movement in a *CMM*.

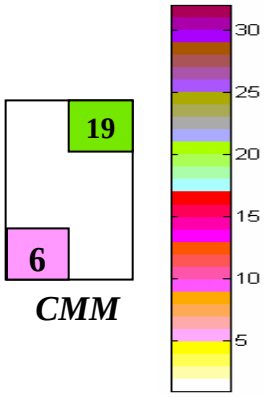

Figure 3. Stacking *Binary Motion Maps* (*BMMs*) of Figure 2 **into a *Color Motion Map* (*CMM*).**

We use the following two figures, Figure 4 and Figure 5, to show that different animal movements will be converted and represented by very different color images under our proposed model.

The sub-figures (a)—(f) of Figure 4 show a sequence of video frames that were taken 500 milliseconds apart over three seconds, during which the mouse was running on the wheel in its cage. The frame numbers are also shown on the top of each video frame. The sub-figures (g)—(k) of Figure 4 show *BMMs* obtained by taking the XOR of the black-and-white images converted from sub-figures (a)—(f). For example, sub-figure (g) of Figure 4 is the XOR-result of the black-and-white images of sub-figures (a) and (b), and sub-figure (h) of Figure 4 is the XOR-result of the black-and-white images of sub-figures (b) and (c), etc. Sub-figure (l) is the *CMM* obtained by stacking *BMMs* from (g) to (k) of Figure 4.

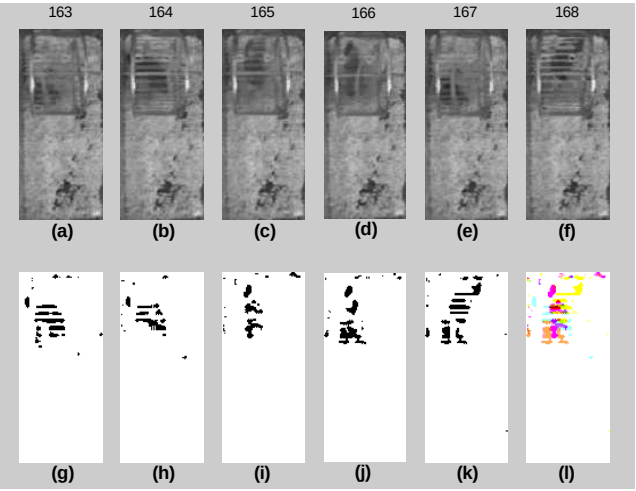

Figure 4. (a)—(f) show a sequence of video frames in which the mouse was running on the wheel. (g)—(f) are *BMMs* constructed from (a) to (f). (l) is the *CMM* constructed from (g) to (k).

Similarly, Figure 5 shows another 3-second video sequence during which the mouse got off the wheel and moved toward the other end of its cage.

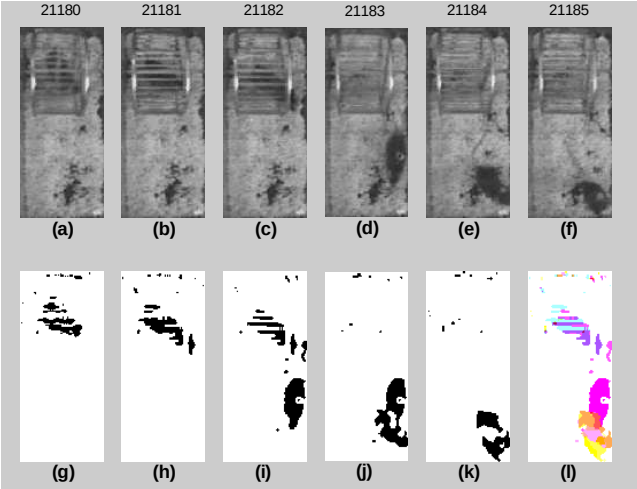

Figure 5. (a)—(f) show a sequence of video frames in which the mouse left the wheel and moved toward the other end of the cage. (g)—(f) are *BMMs* constructed from (a) to (f). (l) is the *CMM* constructed from (g) to (k).

Finally, under our model, we will compute a *BMM* for each consecutive pair of video frames, and stack every w consecutive *BMM*s together to generate a sequence of *CMM*s. For example, frames 21180 to 21185 shown in Figure 5 are used to generate a set of *BMM*s and a *CMM* shown in subfigures (g) to (l) of Figure 5. The next *CMM* will be generated from *BMM*s that are generated from frames 21181 to 21186. This is generally referred to as a *sliding window* approach with the window size equal to w, the number of *BMM*s being stacked together. More specifically, for a video with m video frames, we need to compute *BMM* and *CMM* for (m–1) and (m–w–1) times, respectively.

Note that although the sliding window approach is used in computing *CMM*s, the system only needs to keep one latest *CMM* at any time during the entire video processing. This is because of the following two reasons. First, a new *CMM* can be easily obtained by shifting one bit of all the pixels in the current *CMM* followed by an OR operation to integrate the latest *BMM*. Second, after constructing a *CMM*, the *CMM* will be immediately transformed and saved as a motion vector (see next subsection). As a result, there is no need to keep multiple *CMM*s in the computer memory.

Since $w << m$ is usually the case, the space and time complexity in generating *CMM*s from a video with m frames are $O(n)$ and $O(n \times m)$, respectively, where n is the number of pixels in a video frame. The space and time complexity for computing *BMM*s are the same.

3.2 Extracting Motion Descriptors

While Figure 4 and Figure 5 show the capability of the proposed *CMM* technique in distinguishing different animal activities, a method in quantifying *CMM*s into numeric vectors is needed so that data mining techniques can measure motion-similarity of *CMM*s and objectively cluster movements into various animal activities. Moreover, this quantifying method must be able to tolerate noises generated by the imperfect cage environment and the threshold used in converting each video frame into black-and-white image as discussed at the beginning of Section 3.1.

In this research, we adopt the *Color Autocorrelogram* (*CAG*) approach [3] [9] in converting a *CMM* into a vector. The basic idea of *CAG* can be informally summarized as follow: First, for each pixel p of a *CMM*, we calculate the probability of finding the same color from neighboring pixels of p that are at the *chessboard* distance of d. Next, the probabilities calculated for all pixels are then aggregated into a *CAG* vector based on individual colors. As a result, a color autocorrelogram expresses how the spatial correlation of colors changes with distance. A color autocorrelogram is different from a color histogram which captures only the color distribution in an image and does not include any spatial correlation information.

Next, we give the formal definition of color autocorrelogram. Let M_c denote a set of all possible colors in a color motion map M. In our case the size of M_c is 2^w, where w is the number of *BMM*s being stacked together. Furthermore, the i^{th} color in M_c is denoted as M_{ci}. A pixel $p_2 = (x_2, y_2)$ is said to be a neighboring pixel of $p_1 = (x_1, y_1)$ at the chessboard distance $d = |p_1 - p_2|$ if $\max(|x_1 - x_2|, |y_1 - y_2|) = d$. The color autocorrelogram of M with all the possible colors M_c is denoted as:

$$\gamma_{C_i}^{(d)}(M) \equiv \Pr[\,|\, p_1 - p_2 \,|= d, p_1 \in M_{C_i} \;\&\; p_2 \in M_{C_i}\,]$$

We use the following example to illustrate the color autocorrelogram. Figure 6 shows two different 3×4 *CMM*s with each pixel being either black or white (i.e. $w = 1$). For each pixel $p_{r,c}$, the probability of finding the same color in its immediate neighbors (i.e. $d = 1$) is also displayed. Note that the neighboring pixels that fall outside the *CMM*s are not counted in the probability calculation since their colors are presumably unknown. For example, there are three pixels ($p_{1,2}$, $p_{2,1}$, $p_{2,2}$) at distance 1 from the pixel $p_{1,1}$, the pixel at the top-left corner of the left *CMM*. Since the color of $p_{1,1}$ is white and there are two white pixels among its three neighbors, the probability of $p_{1,1}$ is calculated as 2/3. If $d = 2$, the neighbors of $p_{1,1}$ are $p_{1,3}$, $p_{2,3}$, $p_{3,1}$, $p_{3,2}$, $p_{3,3}$ and the probability of $p_{1,1}$ is 4/5.

2/3	3/5	4/5	2/3
0/5	6/8	0/8	4/5
2/3	3/5	4/5	2/3

2/3	3/5	3/5	2/3
4/5	1/8	1/8	4/5
2/3	3/5	3/5	2/3

Figure 6. Two simple *CMMs* and the probabilities of finding the same color in the immediate neighbors of each pixel.

After summing up the probability from each pixel based on individual colors, the final *CAG* vector for the left *CMM* is: V_L = <0/8, 842/120>, indicating the spatial correlation of the black and white colors in the left *CMM*. The *CAG* vector for the right image is V_R = <2/8, 800/120>. Obviously, $V_L \neq V_R$ because these two *CMM*s are different.

Next, we use real video frames shown in Figure 4 and Figure 5 to demonstrate that different behaviors in a video can lead to very different *CAG* vectors. Figure 7.(a) and Figure 7.(b) give the *CAG* vectors that are derived from the *CMM*s shown in Figure 4.(*l*) and Figure 5.(*l*), respectively. As we can see that the *CAG* vectors of

the two different mouse activities appear very different. Note that since color 0 is the dominated white background color of *CMM*s, the probability of finding white neighboring pixels is relatively high as indicated in Figure 7. Hence, we zoom in to show the detailed probabilities of colors 2-31 in Figure 7.

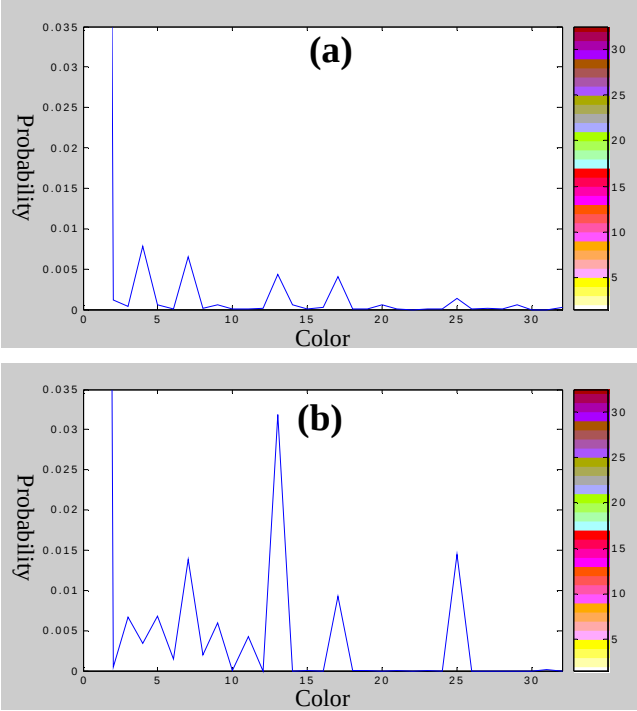

Figure 7. (a) *CAG* vector of Figure 4 (mouse running on the wheel). (b) *CAG* vector of Figure 5 (mouse moving in the cage).

There are few important advantages in transforming a *CMM* by using the color autocorrelogram approach. The first advantage is that the *CAG* vectors can be efficiently computed: From the above discussion, we know that we need to search the same color for each pixel in its maximum 8×*d* neighbors. Hence, the complexity for computing a *CAG* vector from a *CMM* of *n* pixels is O(*n*×*d*). As a result, the complexity of generating all the *CAG* vectors from a video with *m* frames is O(*m*×*n*×*d*).

The second advantage of *CAG* is that it is invariant to rotation while retaining the spatial correlation of colors in an image [2]. For instance, we will obtain the same *CAG* vectors even if we rotate the images in Figure 6. This feature is very important in identifying similar/dissimilar motions for the following reason: although the same movements that occur toward different directions at different locations may appear as different color images, these color images should have the same (or very similar) color correlations and be converted into the same (or similar) *CAG* vectors. The distance (dissimilarity) between two *CAG* vectors will increase only if the corresponding *CMM*s have very different color correlations that represent different activities.

The third advantage is that the impact of noise in videos can be decreased during the *CAG* computation. This is because the noise caused by inconsistent infrared lighting or other factors usually appears at random locations in different video frames. This spatial and temporal randomness make the *CAG* computation harder to find the same color in the neighbors of noise pixels, reducing the influences of noise pixels in the final *CAG* vectors.

As we explained before, it is unlikely the color autocorrelogram will assign similar *CAG* vectors to represent very different animal movements. However, it is possible that different *CAG* vectors may be computed for slightly different animal movements that are considered as similar by human beings. For example, while biologists may consider there is only one type of grooming activity, this activity may generate different *CAG* vectors due to the speed and the shape of body movement. This issue can be addressed by the users when they attached meaningful annotations to the video frames which we will discuss in the next section.

4. Clustering Motion Descriptors into Meaningful Behavior Patterns

After generating *CAG* vectors from a video, the task of mining recurring behavior patterns can now be reduced to the task of clustering similar color images that are represented by *CAG* vectors in a database. In this section we explain how to cluster *CAG* vectors and annotate each cluster (and its constituent video frames) with useful descriptions. Figure 8 summarizes this process. The circles in Figure 8 indicate the sub-tasks of the mining process, while the boxes represent the data entities produced by those sub-tasks.

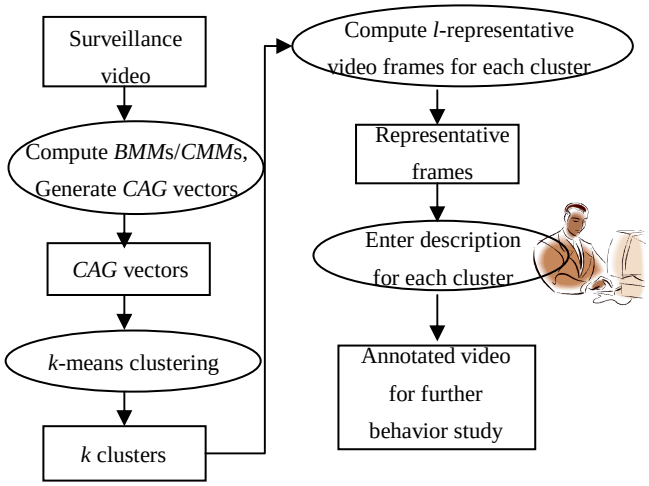

Figure 8. Summarized clustering process.

In Figure 8, each pair of consecutive gray-scale video frames are first converted to a *BMM*, and each sliding window of *w BMM*s are attached together to form a *CMM* and a corresponding *CAG* vector is extracted and saved in a database as we discussed in Section 3.

A *k*-means method is then used to cluster the *CAG* vectors into *k* clusters. Note that scientists may not have prior knowledge of how many distinct behaviors an animal with different physical conditions may display. Hence, the purpose of using *k*-means method in our clustering process is not to extract exactly *k*-types of different behaviors. Instead, the purpose is to summarize a prohibitedly long video into a manageable number of clusters such that each cluster contains similar video frames with highly similar animal movements.

These *k* clusters enable scientists to focus their attentions on studying the few representative frames extracted from each cluster and creating meaningful annotations (in natural language) for each cluster. The annotation created for each cluster is also automatically attached to each video frame belonging to the cluster. Figure 10 to Figure 13 in the next section show some sample video frames that are automatically annotated with the descriptions entered by a user. The annotations associated with video frames can also be saved into a database for further behavioral analysis.

The representative frames of each cluster can be automatically selected by the following steps. First, the center of each cluster is calculated by averaging all the *CAG* vectors in the cluster. Secondly, the *l*-nearest vectors to the center of each cluster are then identified as representative vectors of each cluster. Finally, the video frames correspond to those representative vectors are displayed to assist scientists enter meaningful description for each cluster. If a cluster has less than *l* vectors, movements in the video frames of this cluster can also be identified as rare behaviors or outliers.

5. Experiments

In this section we apply our approach to extract behavior patterns from a 30-minute surveillance video to demonstrate the applicability of our approach. We also compute the precision of our approach at the end of this section.

In this experiment, we set our system to take a video shot every second and set the sliding window size to $w = 3$. More specifically, our system stacks every three consecutive *BMM*s together to form a *CMM* (with maximum 2^3 different colors) and to generate a *CAG* vector. We then use the *k*-means method to create $k = 100$ clusters and compute their centers. For each cluster, our system displays $l = 3$ representative frames and asks users to enter a short description to describe the animal movements in the cluster.

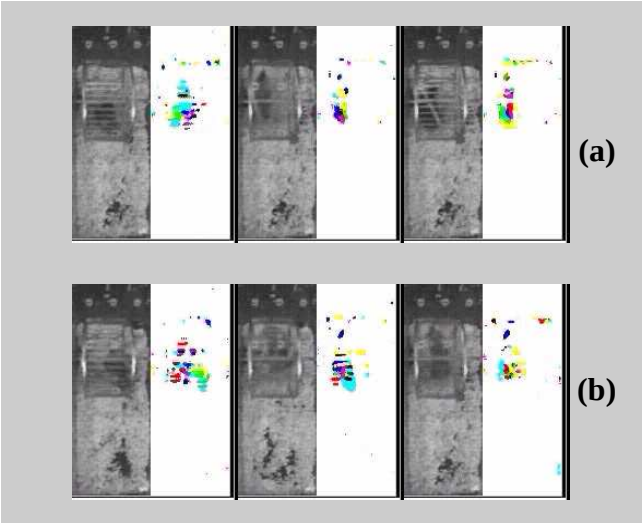

Figure 9. (a) Three representative frames of cluster 2. (b) Three representative frames of cluster 36.

Figure 9 (a) and (b) show the three representative frames for cluster 2 and cluster 36, respectively. Although those frames are clustered into two clusters based on their *CAG* vectors (i.e. *CMM*s), they simply show different stages of wheel-running activities. Hence, we enter the description "wheel" for both clusters. If more detailed separation on the general wheel activity is needed, different descriptions, such as "slow-wheel-activity" and "fast-wheel-activity", can also be entered for individual clusters.

After entering descriptions for all clusters, scientists can then study the videos that are automatically annotated with the descriptions. We use next few figures to show few short periods within the 30-minute video that contain different mouse activities.

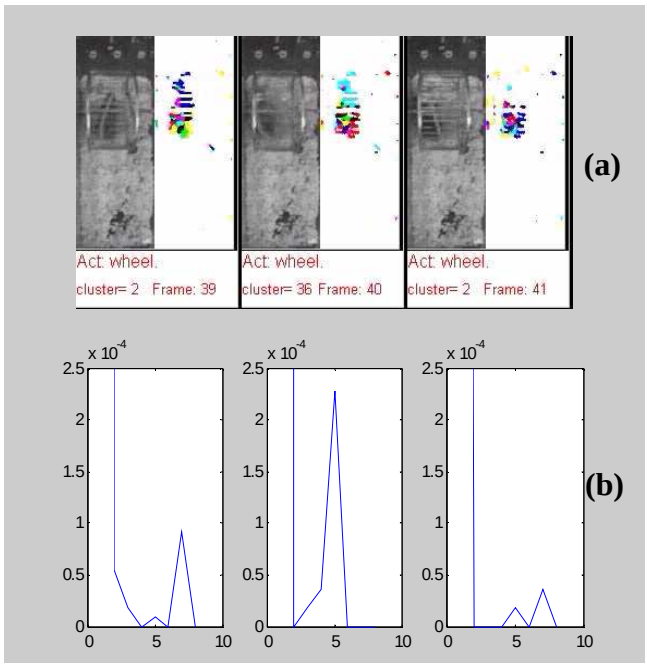

Figure 10. (a) "wheel" activity that comes from two different clusters (2 and 36) over a three-second period. (b) The plot of the *CAG* vectors from these three frames.

Figure 10.(a) shows the mouse running on the wheel over three consecutive video frames (3 seconds). The *CAG* vectors of these frames are also given in Figure 10.(b). Note that although the descriptions for these frames indicate the "wheel" activity, these frames actually belong to two different clusters because the *CAG* vector derived from the frame 40 is different from the vectors in the two other frames.

Similarly, Figure 11 and Figure 12 show two different kinds of mouse activities: grooming and quite (inactive).

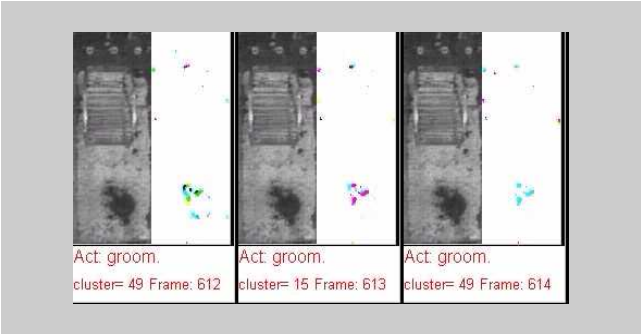

Figure 11. Grooming activity over a three-second period.

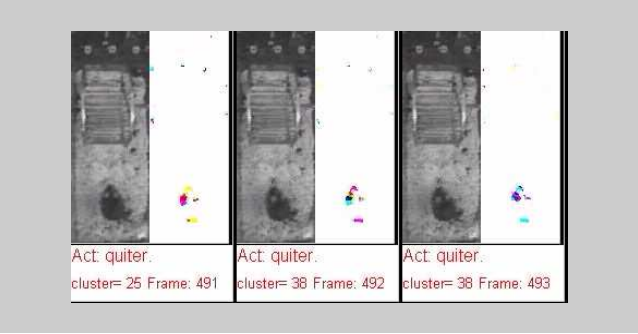

Figure 12. Inactive mouse over a three-second period from frame 491 to 493.

Figure 13 at the end of this paper (next page) shows a sequence of video frames over a short 10-second period. During this period the mouse displayed multiple brief activities that are embedded in a sequence of other movements: It started with a long period of running on the wheel (frame 905—906 and earlier frames), got off the wheel (frame 907—908), quickly turned around (frame 909), and moved back toward the wheel (frame 910—912), and eventually ended up running on the wheel again (frame 913—914 and more later frames). This series of video frames demonstrates the capability of our motion model in correctly capturing very brief but distinguished motions that are embedded in a long sequence of video frames.

Finally, after our system automatically generates the annotated video, we went through each video frame to manually verify whether the automatically attached annotation matches the content of the frame. We found that, out of 1793 video frames (29.88-minute video), 1576 frames are correctly clustered and annotated. This gives us 88% of precision rate. We believe that this precision rate can be further improved if video frames are taken in a higher rate (i.e. 500 ms) or a larger *w* is used in stacking *BMM*s together into a *CMM*. We are planning to conduct more experiments to verify our hypothesis.

6. Conclusion

In this paper we present a new motion model that can efficiently capture the spatial and temporal features of animal movements from a video. Our model converts those spatial and temporal motion features into a sequence of color images that we refer to as color motion maps (*CMM*s). Each *CMM* is then transformed into a color autocorrelogram (*CAG*) vector that measures the spatial correlation of colors in a *CMM*. A clustering method can then be used to cluster similar *CAG* vectors into clusters. The objective of each cluster is to collect video frames that contain similar animal motions. A small number of representative frames from each cluster can then be automatically selected and presented to scientists for soliciting annotations. Solicited annotations are then automatically attached to the video frames of the cluster for further behavior study.

Our experiments show that our model is able to identify different animal activities in high precision (88%) from surveillance videos, even when activities are brief and embedded in between many other activities. We are planning to study the precision of our approach by taking video in a higher rate or setting a larger *w* in stacking *BMM*s together into a *CMM*.

Under our current approach, scientists must first capture videos before they can perform the off-line analysis of animal's behaviors. This off-line analysis can only provide feedbacks to the scientists after an experiment ends. Moreover, this approach also requires large disk space to store long videos. Hence, we are also planning to apply some classification methods, such as the decision tree method, to extract rules from the *CAG* vectors generated in this research. We hope that the extracted rules can enable us to perform online analysis in which animal movement in a video frame can be classified into certain type of activities in real-time. This online analysis can provide much faster feedbacks to scientists who can then adjust the experiment settings based on the early results.

7. ACKNOWLEDGMENTS

The authors wish to express their gratitude toward the anonymous reviewers for their valuable comments.

8. REFERENCES

[1] S. Belongie, K. Branson, P. Dollar, and V. Rabaud, "*Monitoring Animal Behavior in the Smart Vivarium*", 5th International Conference on Methods and Techniques in Behavioral Research, 30 August - 2 September 2005, Wageningen, The Netherlands.

[2] R. C. Gonzalez, and R. E. Woods, "*Digital Image Processing*", Prentice Hall 2002.

[3] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms", IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Washington D.C., 1997.

[4] A. S. Kane, J. D. Salierno, G. T. Gipson, t. C. A. Molteno, and C. Hunter, "*A Video-Based Movement Analysis System to Quantify Behavioral Stree Responses of Fish*", Journal of Water Research, 38 (2004), pp. 3993-4001.

[5] Y. Liang, V. Kobla, X. Bai, and Y. Zhang, "*One Step Forward Toward Understanding Lab Animal Behaviors Using Digital Video Technologies*", National Institutes of Health Biomedical Information Science and Technology Initiative (BISTI) meeting, Bethesda, Maryland, 2003.

[6] Junghwan Oh, and B. Bandi, "*Multimedia Data Mining Framework for Raw Video Sequences*", 3rd International Workshop on Multimedia Data Mining, MDM/KDD'02, Edmonton, Alberta, Canada, 2002, pp1-10.

[7] L. Noldus, "*Observing Behavior by Computer*", Scientific Computing, http://www.scientific-computing.com/feature5b.html.

[8] A. Plotnik, and S.M. Rock, "*Quantification of cyclic motion of marine animals from computer vision*", MTS/IEEE Oceans, Biloxi, Mississippi, 2002, pp. 1,575-1581.

[9] Qi Zhao and Hai Tao, "*Object tracking using color correlogram*",The Second IEEE International Workshop on Visual Surveillance and Performance, October 2005.

About the authors:

Chih Lai received the PhD degrees in computer Science from Oregon State University in 1999. He is an associate professor at University of St. Thomas. His research interests include data mining, multimedia databases, neuroinformatics, and real-time systems. Formally, he worked as a principle software engineer on an FAA project and received various U.S. and European patents on aircraft collision avoidance systems. He is a member of IEEE Computer Society and ACM SIGKDD.

Taras Rafa is a research assistant at University of St. Thomas. He received the PhD degree in mathematical modeling and computational methods from Ternopil State Ivan Pul'uj Technical University (Ukraine) in 2003. He then worked as an assistant professor in the Computer Science and Biotechnical Systems department in this university. His research interests are in data mining, computed tomography, and image processing. He is a member of IEEE.

Dwight E. Nelson is on the faculty in Biology and conducts neuroscience research at the University of St. Thomas. His research focuses on mammalian (mouse) circadian rhtyhmicity and behavior . He earned a Ph.D. from the Department of Neurobiology and Physiology at Northwestern University in Evanston, IL, and conducted post-doctoral work at the University of Virginia Center for Biological Timing (Charlottesville, VA) and the Pharmaceutical Products Division at Abbott Laboratories (North Chicago, IL). His research is supported by the National Institute of Mental Health.

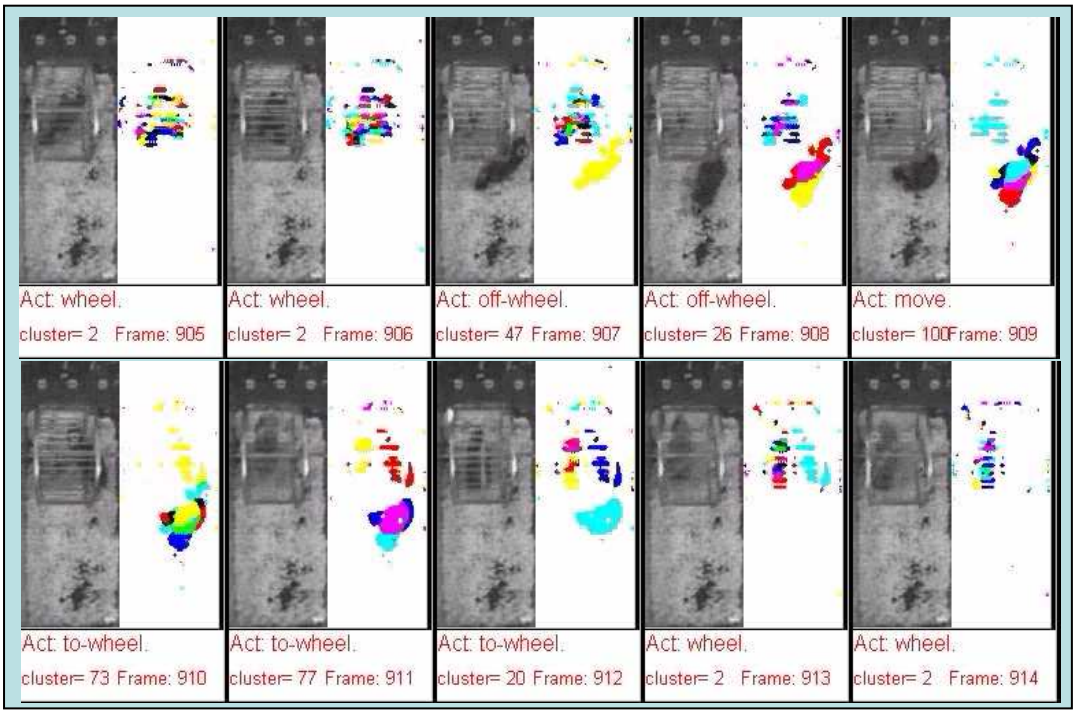

Figure 13. Our motion model correctly captures multiple brief and rapid activities occurred in a short 10-second period.