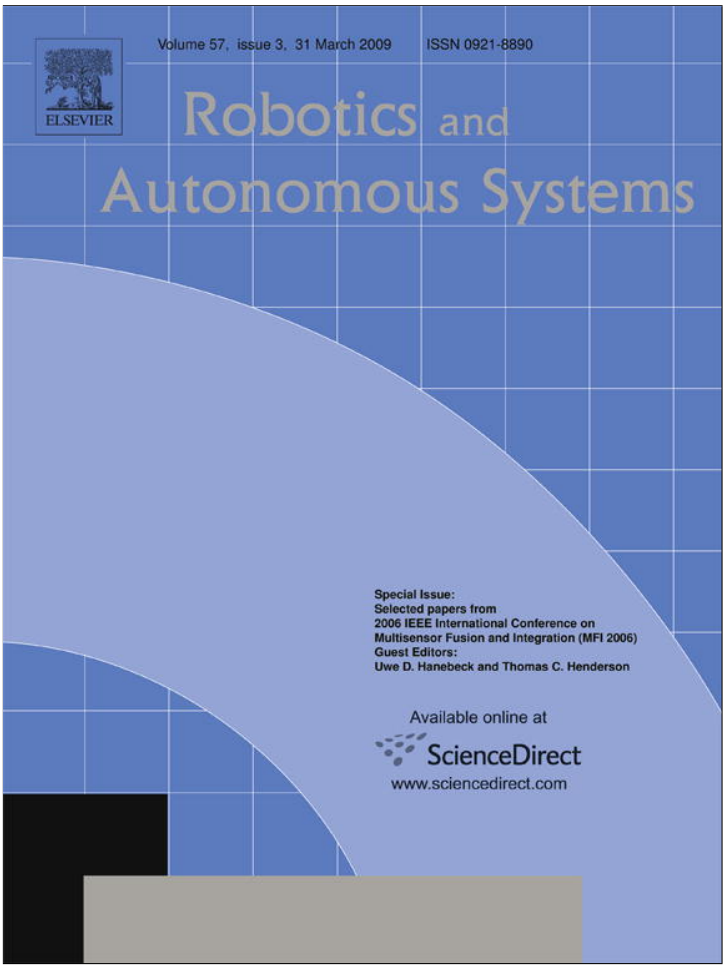


Robotics and Autonomous Systems 57 (2009) 239–248

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Feature fusion for basic behavior unit segmentation from video sequences

Xinwei Xue, Thomas C. Henderson *

*School of Computing, University of Utah, Salt Lake City, Utah 84112, USA*

ARTICLE INFO

*Article history:*
Available online 18 November 2008

*Keywords:*
Vector fusion
Affinity graph
Basic behavior unit
Feature extraction
Multiple cameras

ABSTRACT

It has become increasingly popular to study animal behaviors with the assistance of video recordings. An automated video processing and behavior analysis system is desired to replace the traditional manual annotation. We propose a framework for automatic video based behavior analysis systems, which consists of four major modules: behavior modeling, feature extraction from video sequences, basic behavior unit (BBU) discovery and complex behavior recognition. BBU discovery is performed based on features extracted from video sequences, hence the fusion of multiple dimensional features is very important. In this paper, we explore the application of feature fusion techniques to BBU discovery with one and multiple cameras. We applied the vector fusion (SBP) method, a multi-variate vector visualization technique, in fusing the features obtained from a single camera. This technique reduces the multiple dimensional data into two dimensional (SBP) space, and the spatial and temporal analysis in SBP space can help discover the underlying data groups. Then we present a simple feature fusion technique for BBU discovery from multiple cameras with the affinity graph method. Finally, we present encouraging results on a physical system and a synthetic mouse-in-a-cage scenario from one, two, and three cameras. The feature fusion methods in this paper are simple yet effective.



## 1. Introduction

It has become an increasingly important research area to automatically analyze object behaviors from visually (e.g., motion) captured data or video recordings. The major tasks are to automatically detect and track objects from video sequences and analyze its high level activities or behaviors. Humans and vehicles have been mostly the focus of the visual surveillance and behavior understanding research [1–4] for security purposes, e.g., access control in certain area, anomaly detection in crowded mass transportation area, etc.

In areas of biology, pharmacology, toxicology, entomology and animal welfare, video recordings are popularly used to analyze the behaviors of animals (e.g. lab mice, rodents, poultry, wild animals, etc.) The traditional human annotation approach is time consuming and results may vary from one observer to another. Hence the automatic animal behavior analysis from visual data is drawing more and more attention in both the research and industrial community [5,6].

In the area of visual robot control, it is also desired for robots to automatically learn and recognize behaviors from motion capture or visual data [7–9], which would enable the intelligent robots to respond according to the visual information captured by cameras.

Among all the efforts made in an automated behavior analysis system, the basic behavior unit (BBU) classification (or segmentation) is one important task [10]. Usually the sequences of visual data from images need first to be grouped into BBUs [11], or primitive (atomic) behaviors [7], and then complex behaviors are analyzed based upon the relationship between the BBUs and context. Prior to the BBU segmentation step, spatiotemporal features are usually extracted. In the literature, interest points, shape properties of the detected object blobs, contours, or features derived thereby are used to perform BBU classification. Feature extraction itself is an important task.

In the literature, researchers has been trying to solve the BBU classification and feature extraction tasks separately. In this paper, we take a integrated approach and propose a framework for such an automatic behavior analysis system. We first present the framework, and then focus on investigating feature fusion techniques in BBU discovery: we will present the exploration of the vector fusion method [12] in feature dimension reduction, and the fusion of features from multiple cameras using the affinity graph method.

Our research is motivated by the need of a professor in medicine, who is interested in the automatic video analysis of behavior changes before and after injecting certain medicine to the lab mouse, as shown in Fig. 1. The behaviors interested includes resting, eating, exploring, and mostly importantly, grooming. In this paper, we use behaviors of the mouse-in-cage scenario for our experiments and analysis.

* Corresponding author.
*E-mail addresses:* xwxue@cs.utah.edu (X. Xue), tch@cs.utah.edu (T.C. Henderson).

0921-8890/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2008.10.018

**Fig. 1.** Mouse in cage scenario.

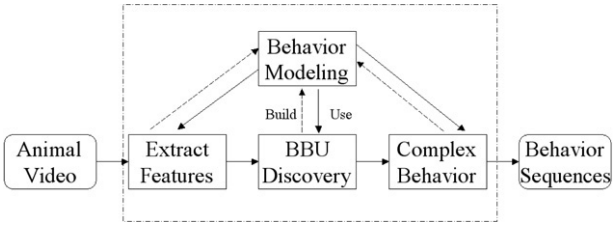



**Fig. 2.** Work-flow for Video Based Behavior Analysis.

## 2. Automatic animal behavior analysis framework

Here we present a four-module framework for video animal behavior analysis: behavior modeling, feature extraction, basic behavior unit (BBU) discovery, and complex behavior analysis, as shown in Fig. 2 (see [10] for a detailed description of relationships between the four blocks enclosed in the dashed box).

**Behavior modeling**. We need to define, characterize, and represent the behaviors of interest in terms of three factors: physical (spatiotemporal) features; the relationship between these behaviors; and the relationship between the animal and its environment. This step interacts with the other three modules. The behavior characterization can then drive the task of feature extraction for basic and complex behaviors (or behavior pattern) recognition, which may in turn help the interpretation of behaviors. Furthermore, another important component in this block is the internal generative model driving the behaviors of an animal [11], which can be helpful in behavior recognition or prediction.

**Feature extraction**. To be able to distinguish behaviors, we need to be able to extract sufficient spatiotemporal physical features of the object from video sequences that represent different behaviors. The features may include: the object's position, posture, speed, contour or region pixels, kinematics and dynamics, motion patterns, etc. We may also need to extract features of the environment, and calculate any other features that can be calculated from these basic features. This process usually requires the ability to detect and track objects from video sequences. Feature dimension reduction and fusion may be necessary when the feature dimensionality is high and features come from more than one sensors.

**Discovery of basic behavior units (BBUs), or behavioral segmentation**. BBUs are the behavior primitives and higher level analysis will be carried out in terms of these. A BBU can be defined as an activity that remains consistent within a period of time, and that can be represented by a set of spatiotemporal variables or features. This step is based upon successful feature extraction. For a mouse-in-cage example, the BBUs of a mouse in a cage can be resting, exploring, eating, grooming, etc. The process of BBU extraction involves mapping the extracted physical features to distinctive behavior units, hence classifying subsequences of the video frames into a sequence of BBUs. BBUs of interest are usually defined for specific applications. The choice of BBUs are completely application dependent.

**Complex behavior analysis**. A complex behavior consists of multiple BBUs with spatial or temporal relationships between them. It is in a higher level of behavioral hierarchy. Once basic behaviors are discovered, complex behaviors can be constructed and analyzed based upon the relationship between the animal's basic behaviors, the interactions of the animal with environment, and with other animals.

In this paper, we concentrate on the feature extraction and BBU discovery modules. We present and discuss feature fusion from one and multiple cameras for BBU discovery. The rest of the paper is organized as follows: Section 3 describes the related work; Section 4 presents the vector fusion method for BBU discovery; Section 5 presents the affinity graph method for BBU discovery and its extension to multiple cameras; Section 6 presents our experiments and shows the results; finally, conclusions are drawn in Section 7.

## 3. Related work

In the visual surveillance literature, most of the existing techniques extract basic behaviors (or actions) directly based upon one or more features extracted (trajectory, motion, posture, etc.) from the detection and tracking results. Pattern recognition techniques (template matching, clustering analysis) are used to classify the video sequence into actions or behavior units, as discussed in the survey papers [1–4]. These methods are effective in their specific applications. The idea is to utilize all the available distinguishing features to perform classification.

Recently, new approaches based on data (or feature) variance or similarity analysis have been developed for discovering BBUs: PCA-related techniques [7,9], and affinity graph-based techniques [13,11,14]. The former capture the variance in a dataset in terms of principle components, and the latter utilize the degree of similarity between the data elements. The commonality of these two approaches lies in the fact that, first a covariance matrix (for PCA) or affinity matrix (for affinity method) is constructed, then Singular Value Decomposition (SVD) is performed to derive eigenvalues and eigenvectors. Segmentation is performed upon the eigenvector corresponding to the largest eigenvalue.

**PCA-related techniques**. Jenkins [7] employs a spatiotemporal nonlinear dimension reduction technique (PCA-based) to derive action and behavior primitives from motion capture data, for modularizing humanoid robot control. They first build spatiotemporal neighborhoods, then compute a matrix $D$ of all pairs' shortest distance paths, and finally perform PCA on the matrix D. Barbic et al. [9] propose three PCA-based approaches which cut on where the intrinsic dimensionality increases or the observed distribution of poses changes, to segment motion into distinct high-level behaviors (such as walking, running, punching, etc.).

**Affinity graph method**. The affinity graph method has mostly been applied in image segmentation, as summarized in [15]. Recently, this method has been applied to event detection in video [13,14]. Though not exactly the same approach, the concept of similarity matrix for classification is applied in gait recognition [16] and action recognition [17].

Different affinity measures have been proposed to construct the affinity matrix. In image segmentation, distance, intensity, color, texture and motion have been used [18]. In video-based event detection, as in [13], a statistical distance measure between video sequences is proposed based on spatiotemporal intensity gradients

at multiple temporal scales. [14] uses a mixture of object-based and frame-based features, which consist of histograms of aspect ratio, slant, orientation, speed, color, size, etc., as generated by the video tracker. Multiple affinity matrices are constructed based on different features, and a weighted sum approach is utilized for constructing the final affinity matrix.

The most closely related methods to our work are [13,14]. [13] constructs an affinity matrix from temporal subsequences using a single feature, while [14] constructs the affinity matrices for each frame based upon weighted multiple features.

We are particularly interested in discovering animal behaviors from video sequences. We propose a framework for discovering basic behaviors from temporal sequences based on multiple spatiotemporal features. In our approach, we combine the advantages of the approaches from [13,14]: (1) We use a classification tree approach with the affinity graph method. (2) We construct the affinity matrix on a subsequence of the frame features (multiple-temporal scale), instead of on one frame. Thus we can encode the time trend feature into the problem, and capture the character of the gradual temporal changes. (3) We apply the affinity graph technique to multiple cameras. Multiple cameras have been used in human posture classification [19,20], where either multiple 2D information fusion or reconstructed 3D information is used. Approaches other than the affinity graph method are used. In our work, we use the multiple camera image information in the simplest way to demonstrate the effectiveness of multiple cameras.

In most of the BBU segmentation methods, the feature data usually have a large dimension, which usually makes the algorithms computationally expensive. Hence feature dimension reduction is often applied before applying BBU segmentation algorithms. The vector fusion method [12,21], is inherently one such technique: it reduces an arbitrarily large dimension to a two dimensional space, which can help discovering the underlying structure of the data. This method was originally proposed as an aid for visualizing the structure of multiple dimensional data, and has also been applied in characterizing and measuring data. Here we propose to explore its applicability in grouping behavioral data.

## 4. The vector fusion algorithm for BBU segmentation

In this section, we describe Johnson's vector fusion method (denoted as **SBP** − **S**ingle-point **B**roken-line **P**arallel-coordinate in [12,21]) and how we apply it in BBU discovery.

The vector fusion method is a vectorized generalization of the parallel coordinates [22] method for visualizing multi-dimensional datasets, which allows one to see any number of dimensions concurrently by arranging the coordinates parallel to each other. The vector fusion method maps a multi-variate vector into a 2D vector, by adding each element of the row (the multi-variate vector) rotated by some angle to the prior one, and summing the whole row to a single-end-point resultant, as expressed in Eq. (1).

$$\begin{aligned}\mathbf{w} &= w_1 e^{i\theta_1} + w_2 e^{i\theta_2} + \cdots + w_d e^{i\theta_d} \\ &= \sum_d w_i \cos(\theta_i) + i \sum_d w_i \sin(\theta_i) \\ &= (w_{\text{sum } x}, w_{\text{sum } y}) \\ &= (SBPx, SBPy) \end{aligned} \quad (1)$$

where

$\theta_i = (i-1)180°/d$

$d$ is the dimension of the multi-variate vector

$w_i$ is the feature value of the $i$th dimension.

This concept is further demonstrated in Fig. 3, which shows how the 4 dimensional vector is fused to form a two-dimensional vector (coordinate). By fusing each element vector of the data, and plotting the final coordinate sequence, this method is able to reveal some underlying structure within the data. The advantage of this method is its simplicity in representing the multiple-dimension vectors. However many dimensions the data element may have, it reduces it to a two dimensional coordinate in SBP space. Johnson has demonstrated its effectiveness in several applications, such as spectral signature identification, medical data analysis, etc. [12,21].

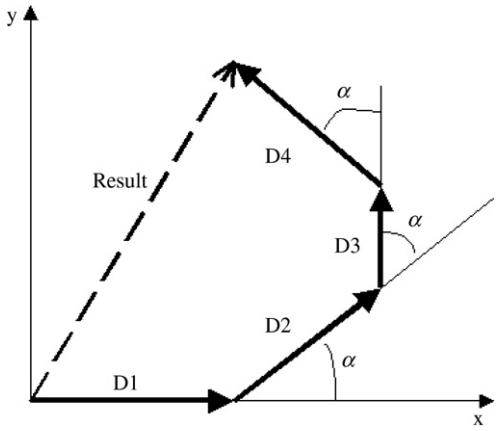



**Fig. 3.** Vector fusion demonstration (vector of 4 dimensions, $\alpha = 45°$).

We are interested in BBU segmentation of visually captured data. The data we have are multiple dimensional sequential feature points, either extracted from video sequences, or calculated analytically. By applying the vector fusion method, the multiple dimensional data is reduced to two-dimensional points in SBP space. We analyze the 2D SBP points in two ways: one is to directly find the spatial structure of the sequence in the SBP space, i.e., identifying clusters of SBP points; the other is to analyze the temporal properties in the SBP space, and discover motion patterns for different BBUs. This can be considered the training process. Then we can group BBUs based upon the spatial and temporal properties of the SBP points.

In Section 6, we present and discuss the results of applying this approach to different dataset, which are based on simulations of a physical system, and an artificial mouse that mimicks the behaviors of a real mouse in a cage scenario.

## 5. BBU discovery with multiple cameras

### 5.1. The affinity graph method

We propose to use the affinity graph method, an unsupervised learning method to discover basic behavior units. Firstly, the spatiotemporal features are extracted from video frames, as in the Feature Extraction block, shown in Fig. 2. Then we take a subsequence (of length $T$) of the features extracted from video images as an element, and calculate the affinity measure between each pair of elements to construct the affinity matrix. Each element overlaps with the next element by a couple of frames, as shown in Fig. 4, like a sliding window.

This is done by choosing an *element* for consideration. Next a matrix is constructed in which each $(i, j)$ entry gives an affinity (or similarity) measure of the $i$th and $j$th elements. The eigenvalues and eigenvectors of the matrix are found, and the eigenvalues give evidence of the strength of a cluster of similar elements. As described in [18,23], if we maximize the objective function $w_n^T \mathcal{A} w_n$ with affinity matrix $\mathcal{A}$ and weight vector $w_n$ linking elements to the $n$th cluster, and requiring $w_n^T w_n = 1$, then the Lagrangian is:

$$w_n^T \mathcal{A} w_n + \lambda(w_n^T w_n - 1)$$

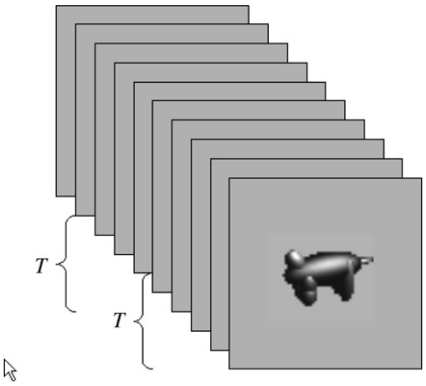



**Fig. 4.** Demonstration of video image subsequence.

where $\lambda$ is Lagrangian multiplier. Differentiation of this formula and dropping a factor of two leads to solving $\mathcal{A}w_n = \lambda w_n$. Therefore, $w_n$ is an eigenvector of $\mathcal{A}$. The eigenvector corresponding to the largest eigenvalue is used to partition the data into two clusters. Then we can iteratively partition the eigenvector corresponding to the next significant eigenvalue until there are no more major clusters [18].

After the eigenvector is generated by Singular Value Decomposition (SVD), a thresholding technique is applied to partition the eigenvector. In [23], manual threshold selection is used, while in [24], the median or a threshold (by search) that minimizes the CUT or NCUT value (see [24]) is used. Here we take a different approach. We first calculate the accumulative histogram of the eigenvector, and smooth it with a Gaussian kernel, and then find the first threshold value that has gradient value smaller than a certain percentage of the number of bins, say 10%. This seems to be effective for our experiment.

The affinity measure we use is the exponential function as used in [18,23,24]:

$$aff(e1, e2) = \exp\{-((f(e1) - f(e2))^{\mathrm{t}}(f(e1) - f(e2))/2\sigma_I^2)\}.$$

Our approach differs from the closest literature [13,14] as described in the related work in four aspects: (1) We construct one affinity matrix based on a feature vector consisting of a set of *weighted* features, instead of calculating affinity matrices for each feature. The combined features provide us with more information. (2) We propose a sequential hierarchical BBU segmentation based upon the distinguishing power of the features. We first use this method to split the video sequences into static and dynamic groups, and then further split each of the static and dynamic groups into BBUs. (3) We construct the affinity matrix on a *subsequence* of the frame features (multiple-temporal scale), instead of on one frame. Selecting the optimal affinity measure, and time scale (length of the subsequence) is our next step. (4) We also apply this approach to multiple camera scenarios.

### 5.2. Affinity graph method for single and multiple cameras

For the one camera case, each *element* consists of a stack of spatiotemporal features extracted from a subsequence (of length $T$) of video images. Here we denote each element as $E[T][D]$ ($D$ is the feature dimension). For multiple cameras that capture the video synchronously, we simply construct the affinity matrix based on elements that concatenate features from the multiple cameras: e.g., the length of the new feature vector for each image is doubled or tripled and so on. So each element is now $E[T][n * D]$ ($n$ is the number of cameras). This is simple, but we are going to show that it is effective.


**Fig. 5.** Feature extraction steps.

### 5.3. Feature extraction and selection

As in the framework shown in Fig. 2, features need to be extracted and selected prior to performing BBU discovery. Basically, our methodology [25] starts from BBUs to find the *intrinsic* variables (based on the notion of intrinsic images in [26]) that can characterize and distinguish them, and then find the corresponding best suitable spatiotemporal features to use for BBU discovery. Several critical questions need to be answered:

(1) What are the intrinsic variables for BBU discovery?
(2) What video features allow recovery of the intrinsic variable values?
(3) What methods to use to extract those features?
(4) How does feature error relate to BBU error?

To answer these questions, the feature extraction and selection module needs to be implemented in the following steps, shown in the Fig. 5.

Generally, a human or animal behavior can be characterized in terms of variables in global motion, local motion, posture, dynamics, orientation, shape, substructure, contexts, etc. depending upon specific BBUs. Let's consider as an example a synthetic mouse scenario consisting of resting, exploring, eating, and grooming BBUs described in Section 6.

The first question is how to determine the *intrinsic* variables. The following criteria are the general guidelines in finding the *intrinsic* variables and video features:

- *Complete.* A sufficient number of intrinsic variables need to be found to ensure the full recovery of the BBUs.
- *Independent.* It is desirable that these variables are independent from each other, hence their distinguishing power fully utilized.
- *Minimal.* It is desirable that the set of intrinsic variables are minimal, hence less redundancy, which would reduce the search in feature space.

For the synthetic mouse, the three variables are: (1) Global motion (speed of the mouse body). The *explore* behavior can be distinguished. (2) The local motion pattern (kinematics of the head or limbs of the mouse). This can distinguish *grooming* behavior from the other BBUs. (3) The posture of the mouse (orientation) and its changing pattern, and the distance of the mouse to the food tank. These variables can distinguish the *eat* behavior from the rest of BBUs. The global and local motion variables can single out the *rest* behavior. The value of these variables can be directly derived from the simulation process. We add noise to these variable values to see how feature noise affects the BBU discovery result.

Next, we find the corresponding features from the synthetically generated mouse video. (1) For global motion variable, we can calculate the speed of the centroid of the bounding box of the detected mouse. (2) For the mouse body posture, we can compute the orientation of the mouse and the eccentricity of its bounding box. (3) The local motion variable of the head and limbs of the mouse can not be directly obtained from the video, but we can approximate this variable by means of calculating the change pattern of the optical flow or the motion history image (MHI) [27]. To extract these features, we need to detect and track the animal silhouette, and calculate these features.

Each intrinsic variable can be translated into more than one video features. Here a feature selection algorithm can be applied. The impact of feature errors on BBU errors can be simulated by degrading the feature values with additive Gaussian noise, as discussed in [25].

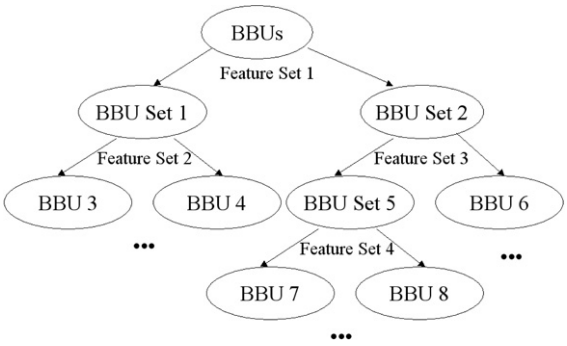



**Fig. 6.** Hierarchical BBU segmentation.

### 5.4. The classification tree approach to BBU discovery

The one-vs-all approach has been popular in the literature. Here we propose to use the classification tree approach (sequential hierarchical classification) with the affinity method, as shown in Fig. 6:

(1) Select the feature set with most distinguishing power, and perform the affinity graph method with these features. This segments the image sequence into several segments.

(2) Select the next feature set with most distinguishing power, and perform BBU segmentation with these features on the segments produced by the previous step.

(3) Repeat step (2) with the rest of the features.

This hierarchical approach is advantageous in utilizing domain knowledge, and is computationally more efficient.

## 6. Experimental results

### 6.1. Vector fusion for BBU discovery

We have experimented with the vector fusion method with data derived from two cases: (1) a bouncing ball, and (2) an artificial mouse.

#### 6.1.1. Bouncing ball

**Data**: In this case, a ball falls down and bounces back, assuming no friction. A temporal sequence of the ball position and speed is generated by simulation, as shown in Fig. 7. The BBUs to be distinguished are 'falling down', 'bounce', and 'rising up'. We use the position and velocity of the ball as input feature data (2D), with the length of 100.

**Result**: The result of applying vector fusion method to the bouncing ball is shown in Fig. 8. Note that, in this figure, as well as in the Figs. 11–17, the horizontal axis is the $SBP_x$ coordinate, and the vertical axis is the $SBP_y$ coordinate. In the bouncing ball example, the point where the ball reaches its highest position corresponds to the rightmost point (denoted as $P1$) in Fig. 8, the point where the ball has the lowest position corresponds to the upper-left-most point (denoted as $P2$) in Fig. 8, and the point immediately after the lowest position corresponds to the bottom-left-most point (bouncing point, denoted as $P3$) in Fig. 8. The 'falling down' BBU corresponds to the section of curve between the $P1$ and $P2$, 'bounce' corresponds to the transition from $P2$ to $P3$, and 'rising up' corresponds to the curve from $P3$ to $P1$.

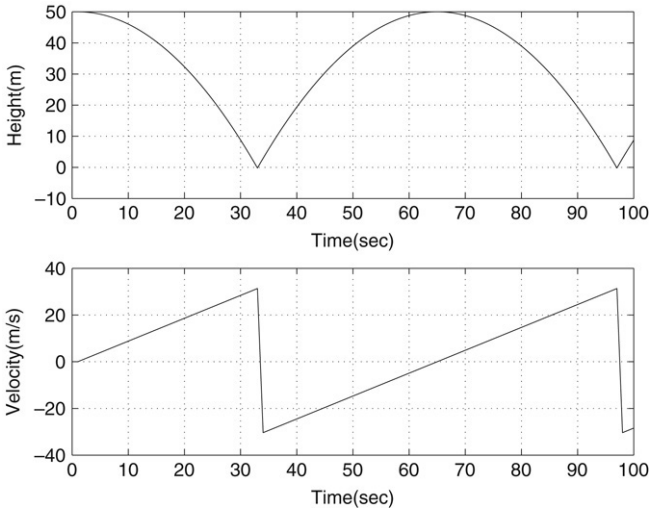



**Fig. 7.** Bouncing ball example (position and speed).

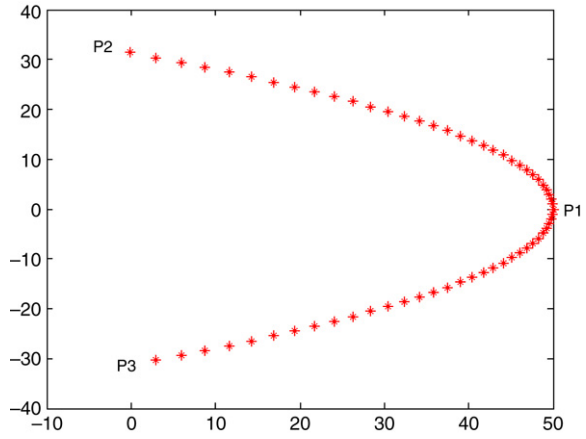



**Fig. 8.** Vector fusion result for bouncing ball (position and speed). Horizonal axis: $SBP_x$; vertical axis: $SBP_y$.

#### 6.1.2. Artificial mouse video data

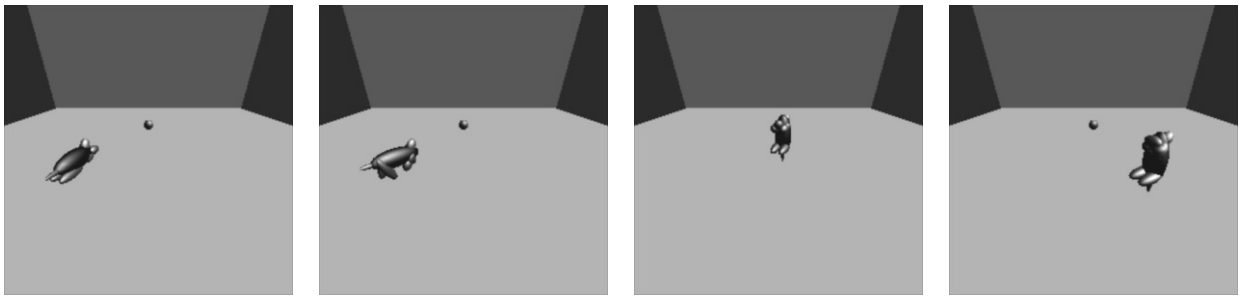
We synthesized several clips of the mouse-in-cage scenario, where the artificial mouse is constructed with ellipsoids. There are four behaviors simulated in this video, shown in Fig. 9:

- **Resting**. No movement. The body and limbs do not move.
- **Exploring**. The body moves in random directions, while the limbs move in such a fashion: the front right and back left legs move at the same pace (same rotating angle), and the front left and back right legs move at the same pace.
- **Eating**. Reaching up to the 'food' above (represented as a little sphere), and getting down, and repeat up and down.
- **Grooming**. Standing on tail with two front legs brushing the head with slight body motion.

This 2000-frame synthetic video sequence consists of 8 rest segments, 4 segments of reaching up, 2 grooming segments, and the rest are exploring segments, as shown in Fig. 10.

**Data.** The feature data are obtained in the following two ways:

- **Extraction from synthetic video data**: First, the artificial mouse blob is tracked and extracted from each frame by simple background subtraction method. Then we calculate the following features: the speed (x,y), aspect ratio, filling ratio, the orientation of the extracted bounding box of the synthetic mouse blob, and the orientation of the mouse. Here the orientation can be simply approximated by its angle from horizontal line. Each feature element is a 5-D vector.



(a) Rest. (b) Explore. (c) Eat. (d) Groom.

**Fig. 9.** Synthetic mouse-in-cage scenario video clips.

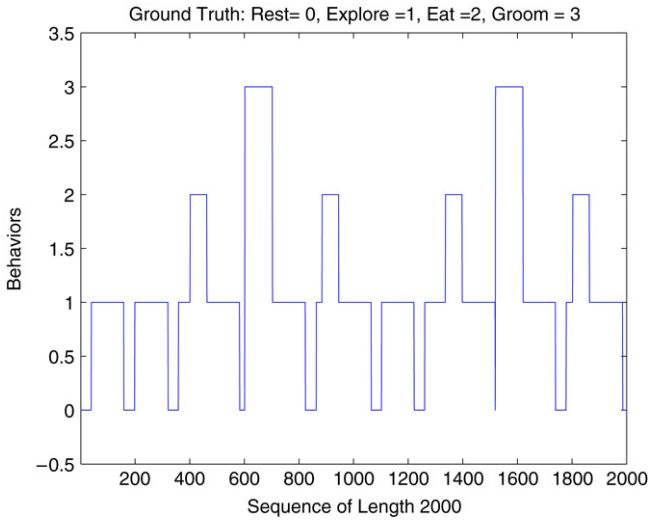



**Fig. 10.** Behaviors in the synthetic video sequence. Rest = 0, Explore = 1, Eat = 2, Groom = 3.

- **Direct analytical data from simulation**: We use a selection of the following features that are calculated analytically or recorded during simulation: position $(x, y, z)$, speed $(v_x, v_y, v_z)$, orientation $(\theta_x, \theta_y, \theta_z)$, and orientation change $(\mathrm{d}\theta_x, \mathrm{d}\theta_y, \mathrm{d}\theta_z)$ of the body and four limbs of the artificial mouse simulation. Both position and orientation are derived analytically from the simulation. Each feature element is a 60-dimensional vector.

**Results.** In all the experiments, each selected feature has the same weight. For the feature data extracted from the synthetic video, the results are shown in Figs. 11 and 12. The four BBUs are not clearly separated.

For the analytical artificial mouse data, if we use all 60-dimensional feature data, the vector fusion result does not distinguish the behaviors either. Fig. 13 uses absolute position of the mouse body and limbs, the orientation is in radians $(0–2\pi)$. Fig. 14 shows the result using relative position of the limbs (relative to the mouse body), and the orientation is in radians. The result of using relative position using radians starts to show some kind of pattern for different BBUs, comparing to using absolute positions. This is reasonable, since the relative motion of the limbs best distinguishes the four BBUs. Also, we found that proper normalization is needed for each dimension of the feature data. Otherwise, the result would not be meaningful. Here in these experiments, we normalize each feature by its mean. An alternative could be $z$-scaling, i.e., use the difference with mean divided by standard deviation.

Based upon the previous results and the analysis of motion pattern (Explore, Eat, and Groom also exhibit some periodic limb motion) for each BBU for the analytical data, we changed to use


**Fig. 11.** Vector fusion result for mouse BBU.

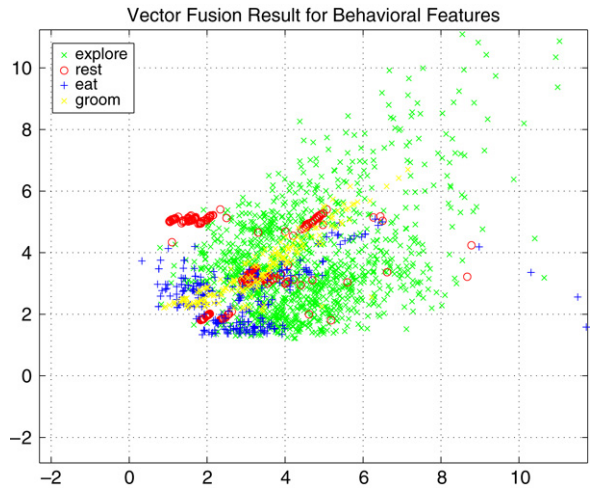



**Fig. 12.** Vector fusion result for mouse BBU Zoom In. '•'—Rest, Green 'x'—Explore, '+'—eat, Yellow 'x'—Groom. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

only artificial mouse limb orientation (rotation angles relative to the mouse body—local motion). Each dimension of the feature data is normalized to the range of 0–1. This time we get much better results, as shown in Fig. 15. The result using artificial mouse limb orientation (four limbs) $(\theta 1, \theta 2, \theta 3)$ and the body speed $(\mathrm{d}\mathbf{x})$ is shown in Fig. 16, comparable to Fig. 15. Now we can easily distinguish the BBUs, by fitting lines or ellipses to the data.

Fig. 17 shows the vector fusion result for each BBU, where the *SBPx* and *SBPy* coordinates of each BBU sequence are plotted (in the vertical axis) against the time step (in the horizontal

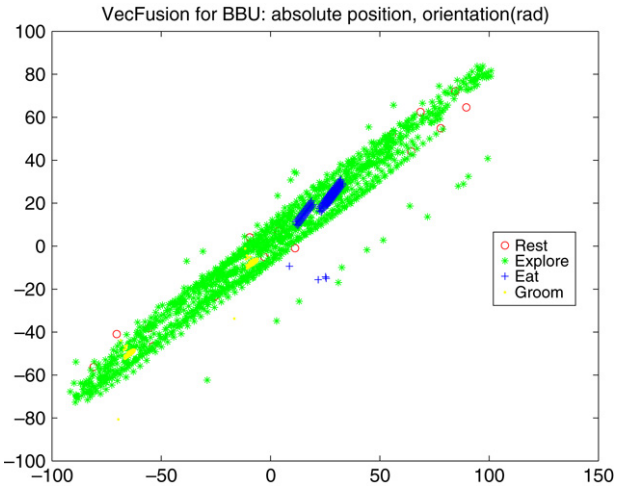



**Fig. 13.** Vector fusion result for mouse BBU — Absolute limb position, total 60 dimensions. '○'—Rest, '∗'—Explore, '+'—eat, '•'—groom.

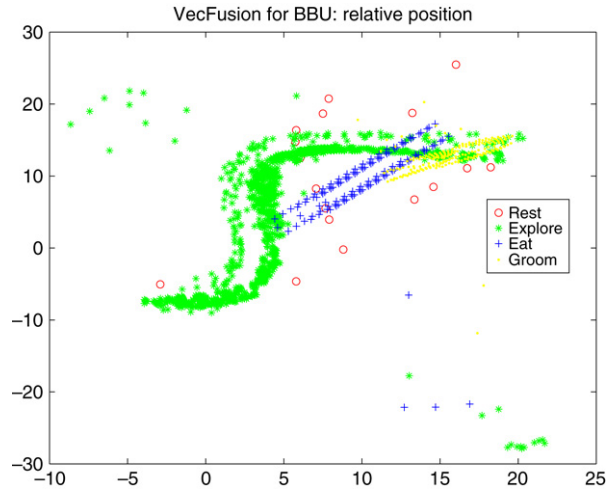



**Fig. 14.** Vector fusion result for mouse BBU — Relative Limb Position, Total 57 Dimensions. '○'—Rest, '∗'—Explore, '+'—eat, '•'—Groom.

axis). The *SBPx* coordinate of each BBU sequence is plotted in the top figure, and the *SBPy* coordinate of each BBU sequence is plotted in the bottom figure. We can see that the SBP coordinate sequence for each BBU exhibits either a stationary or periodic pattern. By making movies of how the *SBPx*, *SBPy* coordinates (or the SBP point in the SBP space) change over time for each BBU, we can observe more clearly the temporal patterns of each BBU (see http://www.cs.utah.edu/~xwxue/vectorFusion/ for the movies): The rest BBU is basically a stationary point, the explore, eat, and groom BBUs show obvious periodic motion along different lines. Hence we can easily distinguish each BBU in the sequence.

### 6.2. BBU discovery results with single and multiple cameras

Here we present BBU discovery results on single and multiple cameras. We use the synthetic mouse data for BBU discovery with one, two and three cameras. For multiple cameras, we simply record the video in multiple locations and record the sequences. The three images captured by three cameras are shown in Fig. 18.

We experimented with the following features extracted from the silhouette of the artificial mouse, as the result of contour tracking or background subtraction: position (centroid of the blob), speed (of the blob centroid), orientation (principle axis of the blob), orientation change, aspect ratio (width/height), aspect

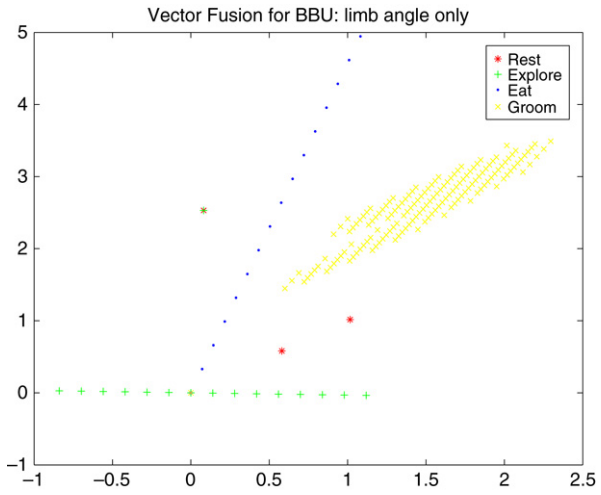



**Fig. 15.** Vector fusion result for mouse BBU — Normalized limb angles only total 8 dimensions. '*'—Rest, '+'—Explore, '•'—eat, 'x'—Groom.

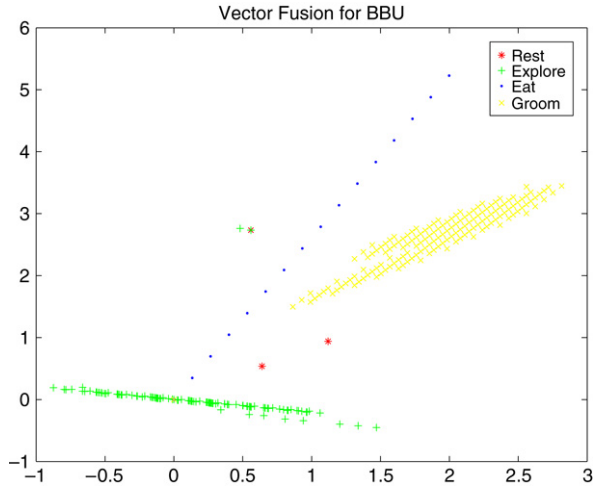



**Fig. 16.** Vector fusion result for mouse BBU — normalized limb angle and body speed, total 9 dimensions. '*'—Rest, '+'—Explore, '•'—eat, 'x'—Groom.

ratio change, and similar features of the motion history image (MHI) [27]. We used a subsequence of length 10 ($T = 10$) and slides one frame at a time in the experiments.

We have tried two approaches: one using combined weighted features in the BBU detection step, the other using a sequential inference approach. The experiment results show that the global motion (i.e., the speed) of the blob is a good feature for segmenting out the frames with no or slight motion. The orientation and its change, and features of MHI are good to separate the grooming (slight global motion, with locomotion) from resting behavior, and separate the reaching up behavior from the exploration behavior. Based upon this observation, here we take a sequential hierarchical BBU segmentation approach with the affinity method, as described in Section 5. We first segment the video into static and dynamic sequences using the affinity measure on the speed feature in step 1. Then the rest of the features are used to segment the *grooming* behavior from the *resting* behavior, and segment the *eating* behavior from the *exploring* behavior.

In our experiment, the BBU segmentation results using multiple cameras achieves better detection accuracy than using only a single camera. We have run 5 experiments with one, two and three cameras, with each experiment having a random variable controlling the moving speed and direction of the artificial mouse.

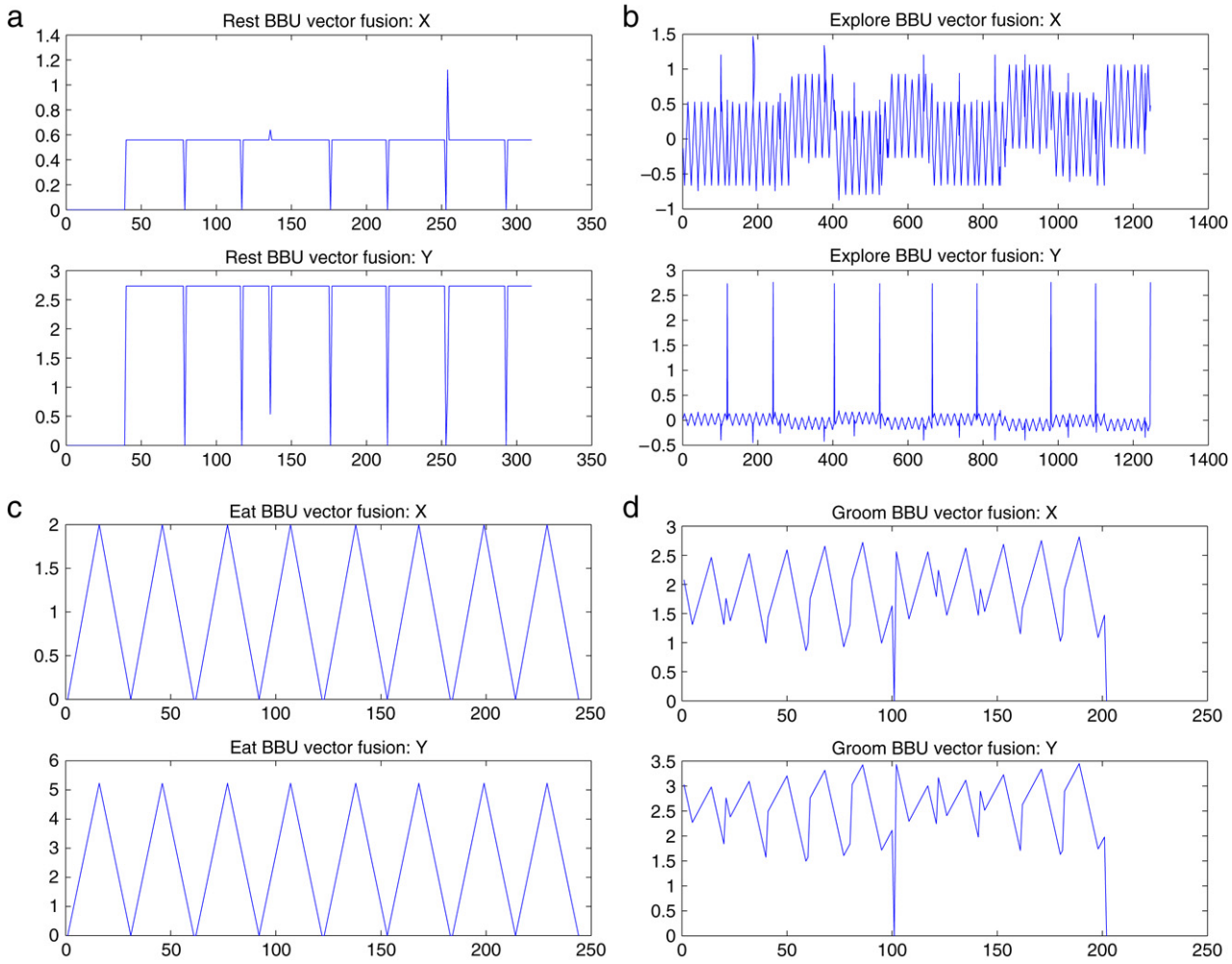
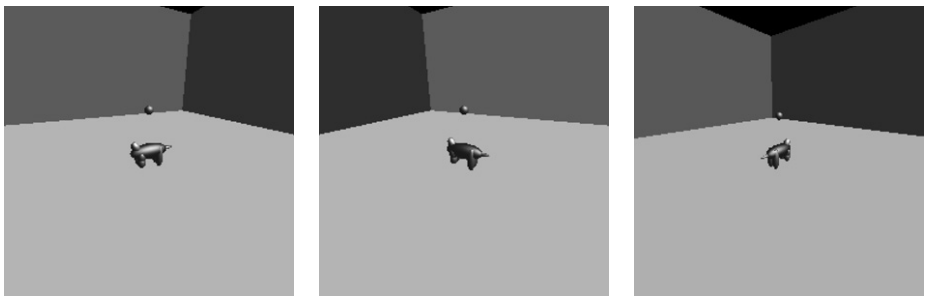



**Fig. 17.** Vector fusion result for each artificial mouse BBU using normalized limb orientation and body speed: (a) Rest, (b) Explore, (c) Eat (d) Groom. For each BBU, the top figure plots the temporal $SBP_x$ coordinate, and the bottom figure plots the temporal $SBP_y$ coordinate.



(a) Left. (b) Middle. (c) Right.

**Fig. 18.** Images captured by three cameras.

The results shows unanimous better results with more cameras. The average error rates are about 10%, 8% and 6% for single, two and three cameras, respectively (this does not include the errors in the interval between each behavior transition, to account for the size of the subsequence window). Fig. 19 compares the static frame discovery results between ground truth, and the best results of single camera, stereo, and three cameras. Fig. 20 shows the best BBU result of the corresponding cases among the 5 experiments.

In the computational aspect, constructing the affinity matrix and SVD process are two major computation components. The computation time for constructing the affinity matrix is proportional to the square of the number of elements $n$ ($n = nFrames/T$). In our experiment for the 2000-frame synthetic sequence ($T = 2000/10$), it takes about 115 s and 3 s, respectively to compute these two components and overall about 2 min in Matlab on a 1.6 GHz laptop with 768 MB RAM.

The errors come from two major sources, one is the selection of features. In the BBU detection, the distinguishing power of the features is essential. Better spatial-temporal features need to be further explored. The other is the choice of affinity measure and the optimal selection of parameters (such as subsequence length, skip length, weights of features, value of sigma in affinity measure, and the threshold selection for bipartition the eigenvector, etc.), which is the next step of this research.

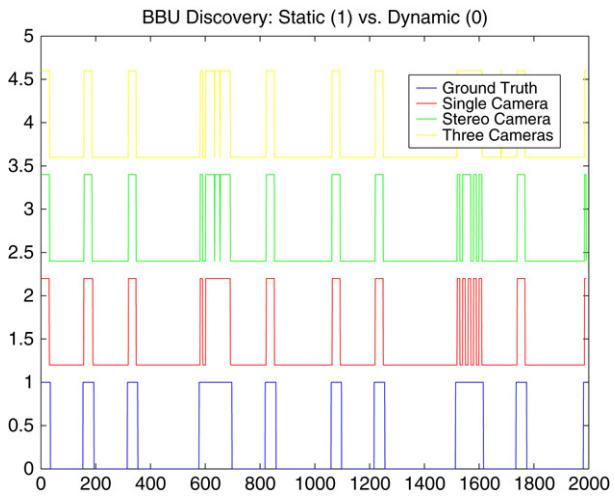
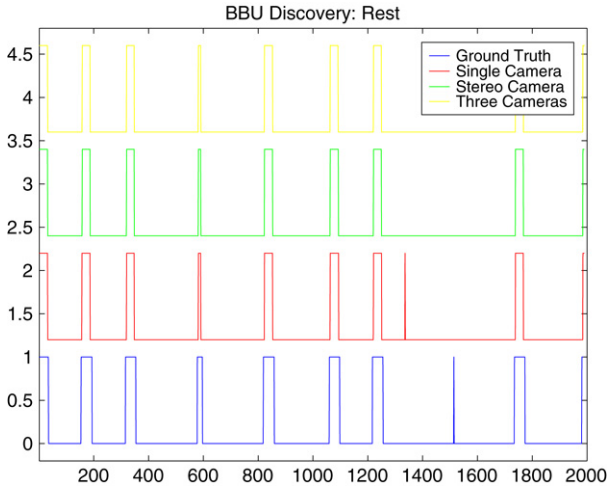



**Fig. 19.** Discovery of static frames: top row: three cameras; second row: stereo cameras; third row: single camera; bottom row: ground truth.

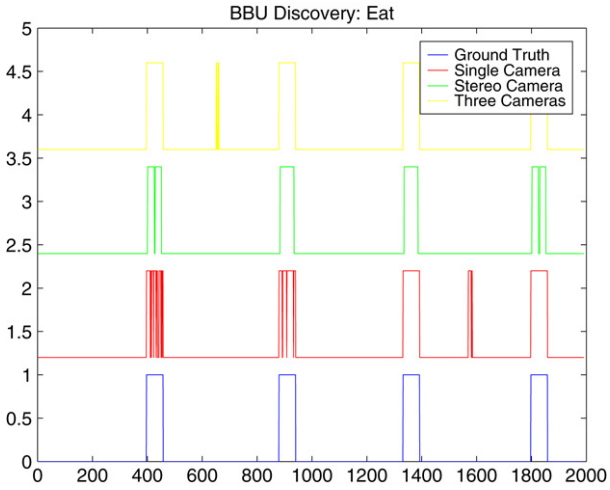
## 7. Conclusions

We propose a framework for video based animal behavior analysis, and concentrate on feature fusion methods for BBU discovery. We have explored the vector fusion method for its application in object basic behavior unit segmentation in a temporal sequence, and presented results on a physical system and a synthetic mouse-in-a-cage scenario. The vector fusion method reduces multiple dimensional data into the 2D SBP space, and the spatial and temporal analysis in SBP space provides a good distinction and interpretation for the bouncing ball example and the analytical data from the synthetic video simulation upon certain selected features.

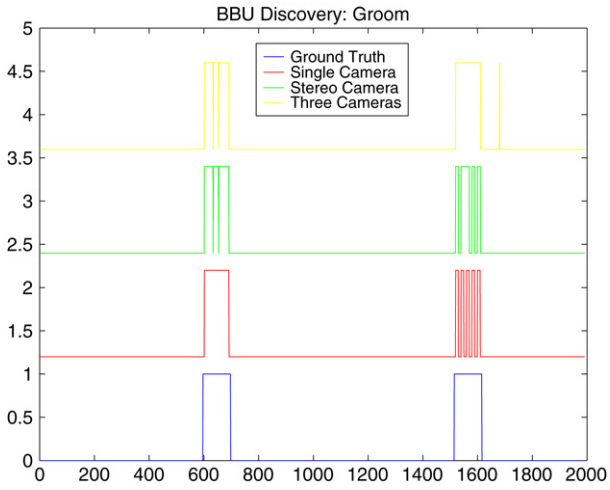
Our experiments show that several factors influences the effectiveness of the vector fusion method in BBU segmentation. First, proper features with enough BBU distinguishing power needs to be selected, just as in other BBU segmentation methods. Second, the weights of each feature element in the multiple-dimensional feature space play an important role, hence, each feature element needs to be properly normalized to account for the different value range (hence different weight) for each feature element, and the distinguishing power of the features. The result of the temporal analysis in SBP space suggests it can be very powerful for BBUs consisting periodic motion [28], and may be potentially a good




(a) Rest.




(b) Eat.




(c) Groom.


(d) Explore.

**Fig. 20.** BBU Discovery result (a) *Resting* (b) *Eating* (c) *Grooming* (d) *Exploring* Top row: three cameras; second row: two cameras; third row: single camera; bottom row: ground truth.

approach for motion capture data analysis (where joint angles can be easily calculated). Its great simplicity (reducing multi-dimensional feature space to the 2D SBP space) is a great advantage over the more complex methods.

We applied the affinity graph method and classification tree approach to perform BBU discovery using features extracted from single, stereo and multiple cameras. The simple feature concatenation fusing method is shown to be effective in the experimental results on synthetic mouse video. The results are encouraging and promising.

Meanwhile, we have noticed that in applying the affinity method in BBU discovery, optimal feature (spatio-temporal features) and parameter (size of subsequence, and number of frames to skip) selection is critical for the successful behavior clustering.

Mutual information feature selection and other feature ranking algorithm could be explored in finding the distinguishing power of the features for BBU discovery. In addition, a probabilistic approach to BBU discovery on top of these methods would be an interesting future research, as well as the study for connecting both SBP and affinity in BBU grouping.

Finally, we are going to apply this method to the real mouse video. Our next step will be conducting complex video animal behavior analysis and uncovering underlying behavior models. For multiple camera cases, where the cameras shall be deployed to get optimal information [29], and how the more complicated information fusion techniques can be applied here will also need to be studied in the future.

## Acknowledgments

The authors thank Bob Johnson for helpful discussions on the vector fusion method.

## References

[1] J. Aggarval, Q. Cai, Human motion analysis: A review, Computer Vision and Image Understanding 73 (3) (1999).
[2] L. Wang, W. Hu, T. Tan, Recent developments in human motion analysis, Chinese Journal of Computers 25 (3) (2002) 225–237.
[3] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics 34 (3) (2004) 334–351.
[4] T. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer Vision and Image Understanding 81 (3) (2001) 231–268.
[5] P. van Lochem, M. Buma, J. Rousseau, L. Noldus, Automatic recognition of behavioral patterns of rats using video imaging and statistical classification, in: Measuring Behavior, Groningen, The Netherlands, 1998.
[6] L. Noldus, A.J. Spink, R.A. Tegelenbosch, Ethovision: A versatile video tracking system for automation of behavioral experiments, Behavior Research Methods, Instruments, & Computers 33 (3) (2001) 398–414.
[7] O.C. Jenkins, M.J. Mataric, Deriving action and behavior primitives from human motion data, in: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS, Lausanne, Switzerland, 2002, pp. 2551–2556.
[8] A. Fod, M.J. Mataric, O.C. Jenkins, Automated derivation of primitives for movement classification, Autonomous Robots 12 (1) (2002) 39–54.
[9] J. Barbic, A. Safonova, J.-Y. Pan, C. Faloutsos, J.K. Hodgins, N.S. Polland, Segmenting motion capture data into distinct behaviors, in: Proc. Graphics Interface 2004, GI'04, London, Ontario, Canada, May 2004.
[10] X. Xue, T.C. Henderson, Video-based animal behavior analysis, University of Utah, TechReport UUCS-06-006, June 2006.
[11] T.C. Henderson, X. Xue, Construct complex behaviors: A simulation study, in ISCA 18th Intl. Conf. on Computer Applications in Industry and Engineering, CAINE, Hawaii, Nov. 2005.
[12] R. Johnson, Visualization of multi-dimensional data with vector fusion, in: IEEE Proc. Visualization, 2000, pp. 297–302.
[13] L. Zelnik-Manor, M. Irani, Event-based analysis of video, in: Proc. IEEE CVPR, Hawaii, 2001.
[14] F. Porikli, T. Haga, Event detection by eigenvector decomposition using object and frame features, in: Workshop on Event Mining, IEEE CVPR, Washington DC, 2004.
[15] Y. Weiss, Segmentation using eigenvectors: A unifying view, in: Proc. IEEE Int. Conference on Computer Vision, Kerkyra, Corfu, Greece, 1999, pp. 975–982.
[16] C. BenAbdelkader, R.G. Cutler, L.S. Davis, Gait recognition using image self-similarity, EURASIP Journal on Applied Signal Processing 4 (2004) 572–585.
[17] A.A. Efros, A.C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: IEEE Int. Conference on Computer Vision, Nice, France, 2003, pp. 726–733.
[18] D. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, Upper Saddle River, NJ, 2003.
[19] R. Cucchiara, A. Prati, R. Vezzani, Posture classification in a multi-camera indoor environment, in: Proc. IEEE Int. Conference on Image Processing, ICIP, vol. 1, Genoa, Italy, 2005, pp. 725 – 728.
[20] S. Pellegrini, L. Iocchi, Human posture tracking and classification through stereo vision, in: Proc. Intl. Conf. on Computer Vision Theory and Applicartions, VISAPP, Setubal, Portugal, 2006.
[21] R. Johnson, Relational data analysis: Characterizing and measuring data to discover relationships in that data. http://www.n-dv.com Research Papers, 2006.
[22] A. Inselberg, B. Dimsdale, Parallel coordinates, a tool for visualizing multivariate relations, in: Human-Machine Interactive Systems, Plenum Press Publishing, New York, 1991.
[23] P. Perona, W. Freeman, A factorization approach to grouping, in: Proc. 5th European Conference of Computer Vision, ECCV, Freiburg, Germany, 1998, pp. 655–670.
[24] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[25] X. Xue, T.C. Henderson, Feature extraction and selection for behavior identification from video sequences, unpublished.
[26] H. Barrow, J.M. Tenenbaum, Recovering intrinsic scene characteristics from images, Computer Vision Systems (1978) 3–26.
[27] J.W. Davis, A.F. Bobick, The representation and recognition of action using temporal templates, in: Proc. IEEE CVPR, San Juan, Puerto Rico, 1997.
[28] R. Cutler, L. Davis, Robust real-time periodic motion detection, analysis, and applications, IEEE Transactions on PAMI 22 (8) (2000) 781–796.
[29] S. Abrams, P.K. Allen, K.A. Tarabanis, Dynamic sensor planning, in: Proc. IEEE International Conference on Robotics and Automation, 1993.

**Xinwei Xue** is currently working with Fair Isaac Corporation as an Analytic Science Scientist and he receives his Ph.D. degree in Computer Science from School of Computing, University of Utah in 2008. He got his and B.S. and M.S. degrees in Precision Instruments from Tianjin University in 1997 and 2000 respectively. His research interest includes image processing, computer vision, video-based object behavior analysis, artificial intelligence and machine learning.

**Thomas C. Henderson** received his B.S in Math with Honors from Louisiana State University in 1973 and his Ph.D. in Computer Science from the University of Texas at Austin in 1979. He is currently a full Professor in the School of Computing at the University of Utah. He has been at Utah since 1982, and was a visiting professor at DLR in Germany in 1980, and at INRIA in France in 1981 and 1987, and at the University of Karlsruhe, Germany in 2003. Prof. Henderson was chairman of the Department of Computer Science at Utah from 1991–1997, and was the founding Director of the School of Computing from 2000–2003. Prof. Henderson is the author of Discrete Relaxation Techniques (University of Oxford Press), and editor of Traditional and Non-Traditional Robotic Sensors (Springer-Verlag); he served for 15 years as Co-Editor-in-Chief of the Journal of Robotics and Autonomous Systems and was an Associate Editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Robotics and Automation. His research interests include autonomous agents, robotics and computer vision, and his ultimate goal is to help realize functional androids. He has produced over 200 scholarly publications, and has been principal investigator on over $8M in research funding. Prof. Henderson is a Fellow of the IEEE, and received the Governor's Medal for Science and Technology in 2000. He enjoys good dinners with friends, reading, playing basketball and hiking.