An AI Approach to Measuring Resident-on-Resident Physical Aggression In Nursing Homes

Datong Chen[1], Howard Wactlar[1], Ashok Bharucha[2], Ming-yu Chen[1], Can Gao[1] and Alex Hauptmann[1]

[1] Carnegie Mellon University
Computer Science Department
417 S. Craig St. Pittsburgh PA 15213
[2] School of Medicine
University of Pittsburgh
3471 Fifth Avenue Pittsburgh, PA 15213
{datong, wactlar, mychen, cgao, hauptmann }@cs.cmu.edu, bharuchaaj@upmc.edu

Abstract

Video surveillance is an alternative approach to staff or self-reporting that has the potential to detect and monitor aggressive behaviors more accurately. In this paper, we propose an automatic algorithm capable of recognizing aggressive behaviors from video records using local binary motion descriptors. The proposed algorithm will increase the accuracy for retrieving aggressive behaviors from video records, and thereby facilitate scientific inquiry into this low frequency but high impact phenomenon that eludes other measurement approaches.

Introduction

Resident-on-resident physical aggression (RRPA) is a highly neglected clinical phenomenon in long-term care settings that has clear implications for public health policy (Shinoda 2004; Lachs 2007). RRPA, as previously defined, is a major public health concern as highlighted in a federal legislative report. Aggregating both RRPA and staff-on-resident aggression (verbal, physical or sexual) using data from the Online Survey, Certification, and Reporting (OSCAR) and Nursing Home Complaints Databases, the report found almost 9,000 abuse violations in more than 5,200 nursing homes nationally over a two year period from January 1999 to January 2001. Over 2,500 of these abuse violations were described as, “serious enough to cause actual harm to the residents or to place them in immediate jeopardy of death or serious injury.” The report however neither attempted to estimate the actual number of residents who were victims of aggression, nor disaggregate RRPA from staff-on-resident aggression. The actual prevalence of aggression is thought to be much higher since over 40% of these abuse violations did not come to light during annual state nursing home inspections but rather were discovered during complaint investigations. Echoing these findings, the Alzheimer’s Association of Central Indiana’s study, *Violence & Dementia*, also concluded that RRPA in particular is significantly underreported for reasons that include: fear of liability, fear of being cited for violations by state health officials, and factors related to suboptimal staffing and training in the management of aggression in nursing homes (Johnson, 2002). In spite of these barriers to truthful disclosure, long-term care ombudsman programs nationally received 3,000 complaints of resident-on-resident aggression in the year 2000, up from approximately 2,500 complaints in 1996. Moreover, since a greater proportion of younger disabled persons (e.g., traumatic brain injuries) and those with chronic mental illnesses who develop severe cognitive deficits are expected to require increased levels of care, the incidence of RRPA will likely continue to rise (Sherrell 1998).

The literature to date has overwhelmingly focused on physical aggression that is directed by residents towards staff during intimate care in private spaces (Hoeffer 2006; Farrell-Miller 1997). In contrast, little is known about the prevalence, phenomenology and longitudinal predictors of RRPA. We postulate that RRPA is significantly under-reported, is more likely to occur during congregate activities, and has the potential to inflict serious physical and psychological harm on frail residents who may be less well able than staff to protect themselves. Moreover, timely and appropriate management of RRPA is a moral imperative since safety is an indispensable aspect of quality-of-life. To date,_the limited data from these retrospective analyses provide only a preliminary estimate of the actual prevalence, and no information about predictors of the phenomenon or its longitudinal course

(Shinoda 2004; Lachs 2007). Thus, the current knowledge base is grossly inadequate for the development of prevention and intervention strategies.

In this paper, we propose an AI algorithm to automatically recognize and classify RRPA behaviors from continuous video records collected in a long-term care nursing home. A dementia unit within the nursing home has been instrumented with unobtrusive ceiling-mounted digital cameras and microphones that directly capture data onto computer hard drives. Consenting residents' activities and behaviors were continuously recorded in real-time 24 hours a day for 26-day periods

In order to analyze the recorded digital data, human coders viewed video segments of the digital recordings using an adjustable speed computer interface and complete an electronic checklist consisting of nearly 80 non-duplicate directly observable aggressive, non-aggressive and positive behaviors. This list was compiled from a review of over 50 instruments that measure dementia related behaviors. Typical aggressions include **spitting, grabbing, banging, pinching/squeezing, punching, elbowing, slapping, tackling, using object as a weapon, taking from others, kicking, scratching, throwing, knocking over, pushing, pulling/tugging, biting, hurting self, obscene gesture, and physically refusing care or activities**. This manual coding is a very expensive human task. The key to enabling a video-based approach for wider application amongst institutions and for longer observation periods is machine-based algorithms that can recognize human activities automatically from video data.

Video-based human action recognition addresses the problem of classifying simple human behavior units from video scenes. The biggest classification challenge is the fact that observed video appearances for each human action contain large variances stemming from body poses, non-rigid body movements, camera angles, clothing textures, and lighting conditions. There are two main approaches to analyzing human motions and actions: model-based and appearance-based.

A model-based approach employs a kinematic model to represent the poses of body parts in each snapshot of body action. A recognition algorithm first aligns the kinematic model to the observed body appearance in each video frame and then codes the motion of the body parts with the model transformations. Most kinematic models are closely related to the physical structure of the human body. Akita (Akita 1984) decomposed the human body into six parts: head, torso, arms and legs, and built a cone model with the six segments corresponding to counterparts in stick images. Hogg (Hogg 1983) used an elliptical cylinder model to describe human walking. A Hidden Markov Model (HMM) was used to recognize tennis actions (Yamato 1983). Yamato, *et al.* extract symbol sequences from image sequences and build an HMM to model the tennis actions. Bregler (Bregler 1997) further extended HMM to dynamic models which contain spatial and temporal blob information extracted from human bodies. Lee, *et al.* (Lee 2002) applied a particle filter on a set of constraints on body poses. Finally, Deutscher, *et al.* (Deutscher 2000) propose an annealed particle filter method that uses simulated annealing to improve the efficiency of searching. Model-based approaches require reliable analytical body part detection and tracking, a complex computer vision problem that continues to merit further exploration.

An appearance-based method builds classifiers to directly remember the appearance of actions in each class without explicitly representing the kinematics of the human body. A good example is template matching, which is widely used as an appearance-based action recognition algorithm. Polana, *et al.* (Polana 1994) computed a spatio-temporal motion magnitude template as the basis for activities recognition. Bobick, *et al.* (Bobick 2001) constructed Motion-Energy Images (MEI) and motion history images as temporal templates and then searched the same patterns in incoming test data. Appearance models can be generally extended to detect various actions without introducing knowledge on constructing domain specific models. However, appearance-based methods require more training examples to learn appearances under different body poses and motions compared with model-based methods. Many appearance-based methods also rely deeply on adequate actor segmentations that are difficult to guarantee.

In recent years, a branch of appearance-based approaches called *part-based approaches* has been attracting interest. A part-based method decomposes the entire appearance of an actor into a set of small, local spatio-temporal components, and applies statistical models to map these local components to actions. It has adequate scalability and does not require constructing specific models as is the case with model-based approaches.. It is also more robust under varying translations, background noise, 2D rotations, and lighting changes than appearance-based methods that require global appearances. Local features in the space-time representation have been applied to human action recognition with an SVM classifier (Schuldt 2004). As an alternative, Dollár, *et al.* (Dollár 2005) proposed to detect sparse space-time interest points using linear filters. Niebles, *et al.* (Niebles 2006) considered an unsupervised learning method to categorize and localize human actions with a collection of spatial-temporal interest points. Ke, *et al.* (Ke 2005) proposed volumetric features to describe events. The features are extracted from optical flow and are represented as combinations of small volumes.

Proposed approach

We propose to characterize human behaviors in surveillance video though the use of local binary motion descriptors (LBMD) as shown in Figure 1. We detect points that are crucial in describing scenes and motions in video and extract spatio-temporal video cubes. A spatio-temporal video cube is a small, short and local video

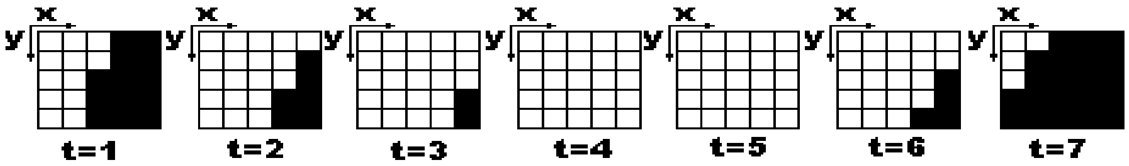

Figure 2: An illustration of local binary motion descriptors

sequence extracted from an interest point to capture small but informative motions in the video. These small motions can be finger raising, knee bending, or lips moving. We then compress a spatio-temporal video cube into a local binary motion descriptor, which can capture local appearance of a combination of these different types of movements and is robust or invariant to global appearance, posture, illumination, occlusion, etc. A video codebook is learned from a large number of LBMDs via a clustering algorithm to merge similar LBMDs together. We then represent each behavior as a bag-of-video-words and build recognizers to classify aggressions from non-aggressive behaviors.

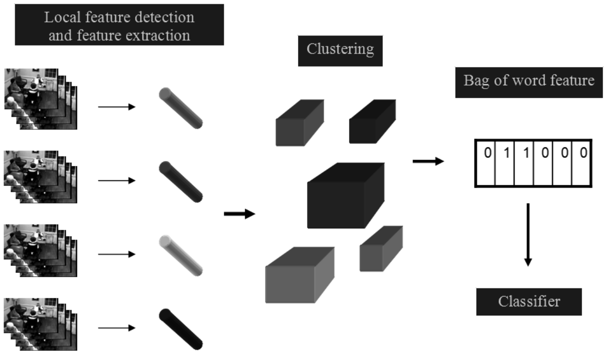

Figure 1: Framework of the proposed human behavior recognition. It includes 3 major stages: (1) feature point detection and local feature extraction, (2) clustering and bag-of-words representation based on video codebooks, and (3) classification to achieve behavior recognition.

Detection of Points of Interest

Local representations are usually extracted from certain interesting points instead of all the image pixels in a video. Typically, an interest point is extracted as a local response maxima pixel that corresponds to a predefined response function. In 2D images, such a response function could be a corner detector. In video, a spatio-temporal corner can be defined as a spatial corner that contains non-constant movements. Laptev, *et al*. (Laptev 2003) extended the Harris interest point detector to extract spatial-temporal corners in a video sequence. Spatial-temporal corners are spatial interest points corresponding to the moments with non-constant motion. In other words, a spatial-temporal corner is a region with strong local gradients in orthogonal directions along *x, y*, and *t*, i.e., a spatial corner or edge whose velocity vector is changing. In practice, true spatial-temporal corners are quite rare. This proved to be a challenge in detection and recognition tasks observed by Lowe (Lowe 2004). Therefore, another local spatial-temporal interest point detector is proposed to detect periodic movements (Dollár 2005). It applies 1D Gabor filters temporally and attempts to capture periodic motions. This provides a richer set of features, but it remains to be seen whether complex actions can be represented by periodic motions alone.

Our proposed interest point detection is based on the Harris corner detector. Instead of corner points in spatial positions, we extract points along edges with velocity vectors by simply replacing the 2nd moment gradient matrix with gradient magnitudes of *x, y,* and *t*. The goal is to find high contrast points both in space and time together. For example, the points are along edges in a video image and contain velocity vectors. In comparison to using the Harris detector and the temporal information (e.g. background subtraction), this algorithm provides dense rather than sparse features. It also contains points with a range of different types of motions, not just periodic motions.

The proposed formula for interest point calculation is as follows:

$$L(x,y,t,\Sigma) = I(x,y,t) * g(0,\Sigma)$$
$$R(x,y,t) = \sqrt{\left(\frac{\partial L}{\partial x}\right)^2 + \left(\frac{\partial L}{\partial y}\right)^2 + \left(\frac{\partial L}{\partial t}\right)^2} \qquad (1)$$

L denotes a smoothed video, which is computed by a convolution between the original video *I* and a Gaussian smoothing kernel *g*. To simplify the computation, we only keep the diagonal values of the covariance matrix Σ and use the variances in *x*, *y*, and *t* dimensions independently to control smoothing scales in space and temporal sequence. The response function *R* combines the magnitudes in the space and temporal dimensions. We calculate the response function value for each pixel and extract local-maxima pixels as interest points. The gradient over time performs a similar function as background subtraction to remove static background and preserve moving objects. We calculate approximate gradients with Sobel operators instead of true gradients to speed up the algorithm.

Local Binary Motion Descriptor (LBMD)

At each interest point, a cube C_{xyt} is extracted which contains the spatio-temporally windowed pixel values in the video. The window size is normally set to contain most of the volume of data that contributed to the response function. We first convert the cube C_{xyt} to be binary cube B_{xyt} by thresholding pixels in the cube with one threshold τ.

The threshold τ is determined by performing a class variance algorithm (Wang 2000) on the first frame of the cube. Formally, the discrete probability distribution function of the intensity of pixels in a cube C_{xyt} is $p(i)$. The probability of pixels below the threshold τ can be expressed as:

$$P(\tau) = \sum_{i=0}^{\tau-1} p(i) \tag{2}$$

A class variance algorithm classifies pixels into two classes B_0 and B_1 with means ($_0$, $_1$) and variances ($\sigma_{0,}$ σ_1), by minimizing the ratio between the within class variance σ_W and between class variance σ_B,

$$\tau^* = \arg\min_{\tau} \frac{\sigma_W(\tau)}{\sigma_B(\tau)} \tag{3}$$

where the means are given by:

$$\mu_0 = \frac{\sum_{i=0}^{\tau-1} ip(i)}{P(\tau)}, \ \mu_1 = \frac{\sum_{i=\tau}^{N} ip(i)}{1-P(\tau)} \tag{4}$$

and N is the maximum intensity value of the pixels. The variances of the two classes are

$$\sigma_0 = \frac{\sum_{i=0}^{\tau-1} (i-\mu_0)^2 p(i)}{P(\tau)}, \ \sigma_1 = \frac{\sum_{i=\tau}^{N} (i-\mu_1)^2 p(i)}{1-P(\tau)} \tag{5}$$

The within class variance σ_W is defined by

$$\sigma_W^2 = P(\tau)\sigma_0^2 + (1-P(\tau))\sigma_1^2 \tag{6}$$

and the between class variance σ_B is

$$\sigma_B^2 = P(\tau)(1-P(\tau))(\mu_0-\mu_1)^2 \tag{7}$$

We choose only the first frame to determine the threshold because an edge passes the center of the first frame of the cube according to our definition of the interest point. We assume that one side of the edge belongs to the actor's body and the other side belongs to the background. Most likely, the two sides contain a similar number of pixels. We expect an adequate threshold to be found by solving a binary classification problem as in the class variance algorithm. In other frames, there may be a very unbalanced number of pixels between the body and background regions due to the actor's motion, where an adequate threshold may be difficult to guarantee.

A local binary feature $BF(S,M)$ is computed from a binary cube B_{xyt}, which consists of shape feature S and motion feature M. The shape feature S is the first frame of the binary cube. We characterize this frame by modeling the "0" and "1" regions with two Gaussians cross the spatial dimensions respectively.

$$S = \left(\mu_x^0, \sigma_x^0, \ {}_y^0, \sigma_y^0, \ {}_x^1, \sigma_x^1, \ {}_y^1, \sigma_y^1\right) \tag{8}$$

The motion feature is defined as a vector which records the motions of the geometric means of the "0" and "1" regions between frames.

$$M = (\Delta x_2, \Delta y_2, \ldots, \Delta x_t, \Delta y_t) \tag{9}$$

The first frame has no motion features. If one of the regions moves out of the cube at frame t, we record motion features in frame i frame $i+1$ as "*NULL*".

LBMD has many advantages in representing local appearances of behaviors. As many other local descriptors, LBMD is invariant to global translations. In comparison to pixel or template representations (Ke 2005, Chen 2008), it compressed the dimension of the feature with a factor of 10. Varying lighting conditions may scale the contrast in a cube and render the grayscale of a cube pixel in very different values. By converting the cube to be binary, we only preserve the shapes of the strong edges in each cube image. Therefore, a resulting binary cube is more robust than the grayscale cube under lighting changes. Specially, the shape and motion descriptors are separated in the LBMD. This makes the LBMD more robust to 3D rotations and movements. Figure 2. displays frames of a binary 5x5x7 cube. We can see that an object moves out of the cube window from the right bottom and moves back.

Feature codebook

Local motion features extracted from the same body part contain similar motion information. Therefore we can cluster them to reduce the feature space into a fixed size of feature codebook.

We apply a modified K-Means algorithm to perform this spatial-constraint clustering. The detailed algorithm was proposed by (Chen 2008). It uses a graphic model to group local object appearance features into clusters under the constraint that spatially nearby local features should most likely be grouped into the same cluster. We replace their 2D appearance features by the proposed local binary motion features and train the clusters with the EM process in the algorithm.

K-Mean clustering. K-Means is a traditional clustering algorithm which iteratively partitions the dataset into K groups. The algorithm relocates group centroids and re-partitions the dataset iteratively to locally minimize the total squared Euclidean distance between the data points and the cluster centroids. Let $X = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^m$ be the set of data points. Let n denotes the total number of data points in dataset and *m* means the dimensionality of feature for data points. We denote $U = \{u_i, \ldots, u_K\}, u_i \in \mathbb{R}^m$ as centroids of clusters and K is the number of clusters. $L = \{l_1, \ldots, l_n\}, l_i \in \{1, \ldots, K\}$ denotes cluster label for each data point in *X*. The K-Means clustering algorithm can be formulized to locally minimize the object function as follow:

$$J_{k-mean} = \min_L \sum_{x_i \in X} D\left(x_i, u_{l_i}\right)$$

$$D\left(x_i, u_{l_i}\right) = \left\| x_i - u_{l_i} \right\|^2 = \left(x_i - u_{l_i}\right)^T \left(x_i - u_{l_i}\right) \tag{10}$$

where J is the objective function of K-Means and *D* denotes a distance function which is L_2 norm distance. The EM algorithm can be applied to locally minimize the object function. In fact, K-Means can be seen as mixture of K Gaussians under the assumption that Gaussians have identity covariance and uniform priors. The objective function is the total squared Euclidean distance between the data point to its center point. There are three steps to achieve K-Means by an EM process: initialization, E-step and M-step. It first initializes K centroids in the feature space and then starts to execute E-step and M-step iteratively till the value of the objective function converges or it reaches maximum iteration. In the E-step, every point is assigned to the cluster that minimizes the sum of the distance between data points and centroids. The M-step updates centroids based on the grouping information computed in the E-step. The EM algorithm is theoretically guaranteed to monotonically decrease the value of objective function and to converge to a local optimal solution. As we mentioned before, the centroids initialization could sometimes decide the local optimal solution and the clustering result.

EM clustering with pair-wised constraints. Our interest point detector tends to detect dense interest points from the outline of moving objects. Therefore, we may extract interest points from positions in the video which are both spatially and temporally nearby. By visualizing our clustering results, we discover that clustering algorithm is sometimes too sensitive. It sometimes separates continuous components into difference clusters. These components are the same part along a time sequence, and people will intuitively expect they are clustered into the same group. This mainly results from two factors. First, the method we use to detect interest points tends to extract rich features. Ideally, we decide to extract points from moving edges and to select representative points by local maxima in the area. However, it extracts a very rich number of video cubes and some of them only have small differences in feature space. When this goes to a clustering process, the small difference can cause conceptually similar cubes to be pushed into different clusters due to sensitivity of the clustering algorithm. The second factor comes from the center point initialization and the distance function in the clustering algorithm. These two factors can also change the clustering result a lot. Center point initialization makes the clustering result unstable because initial center points may not be suitable for the current dataset and thus force the clustering result to fall into a local optimal solution which doesn't help the recognition task. In high dimensional feature space, the distance metric can change the shape of cluster's boundary and the clustering result too. In our proposed method, we believe the spatial-temporal nearby components should be clustered into the same cluster. Therefore, we introduce a pair-wised constraint clustering algorithm to force video cubes which are spatial-temporal related to be clustered into the same cluster if possible during clustering process.

In the original K-Means algorithm, data points are independent of each other. However, in our proposed method, video cubes could have either spatial or temporal dependency between them. Our intuitive idea is to add constraints to video cubes which are both spatial and temporal nearby. Although, we don't really do tracking in our framework, we tend to pair video cubes which are the same component over time and hope they are clustered to the same cluster.

A semi-supervised clustering algorithm tries to employ some data labels to the clustering process and significantly improves the clustering performance. Basu et al. (Basu 2006) propose to add pair-wised constraints in a clustering algorithm to guide the algorithm toward a better grouping of the data. The algorithm aims to manually annotate data and apply this information to the clustering process. They have two different types of relationships between data: must-link pair and cannot-link pair. The idea is very intuitive. The penalties will be added to an objective function if two data points which are labeled as must-link belong to different clusters during the clustering process. If two points are labeled cannot-link but belong to the same cluster during the clustering process, penalties will be added too. In our proposed method, we will only penalize pairs which are spatial and temporal nearby (which we consider as potential continuous components) that belong to different clusters. It's the same as the must-link relation in Basu's method. We do not really manually label the data points. The pairs we generate are purely from data and therefore they are pseudo-labels in our framework.

Therefore, we revise the objective function of clustering as follows:

$$J_{const} = \min_L \left[\sum_{x_i \in X} D\left(x_i, u_{l_i}\right) + \sum_{(x_i, x_j) \in N} \frac{1}{D'\left(x_i, x_j\right)} \delta\left(l_i \neq l_j\right) \right] \tag{11}$$

$$\delta(true) = 1, \delta(f\,alse) = 0$$

The first term of the new objective function remains the same as K-Means. The second term represents our idea to penalize pairs which are considered as continuous components but do not belong to the same cluster. N denotes the set which contains spatial and temporal nearby pairs. δ function equals to one if two data points are not in the same cluster. In the second term, we can discover that penalty is correlated to reversed distance between two data points. Theoretically, continuous components should be very similar in feature space because they could be the same motion unit overtime. In this assumption, the penalty is high if they don't belong to the same cluster. However, two exceptions may happen. The motion is too fast or the motion is changing. If the motion is too fast, we may link different parts together no matter how we define "spatial-temporal nearby". We can try to set up a soft boundary instead of hard boundary to release the strict definition. In practice, we extract hundreds of thousands of video cubes from our data set. It's not tractable given that n-square pairs are involved in the EM process. Therefore, we may mis-label two different cubes as must-link and penalize them if they are not in the same cluster. The other reason we may mis-label data pairs comes from motion changing. Since we try to make spatial and temporal nearby cubes as pairs, we have a good chance to link two cubes from two different actions which are continuous. Since we do not track interest points nor analyze points' relationship in location, we can't avoid these exceptions when we try to connect video cubes as clustering constraints. However, we can reduce the penalty for these mis-labeled pairs. In both exceptions, we believe these pairs have a large difference in feature space. It means the distance between two video cubes should be large and it turns to a small penalty in our penalty function. Therefore, the objective function will be penalized more given that similar pairs in feature space are not in the same cluster. The objective function will be penalized less if the pair is actually quite different in feature space which means the pair does come from continuous motion.

In our work, we replace Euclidean distance in K-Means by the Mahalanobis distance to achieve a Gaussian assumption for partitioning data points. The Mahalanobis distance function is:

$$D\left(x_i, u_{l_i}\right) = \left\|x_i - u_{l_i}\right\|^2_{A_{l_i}} = \left(x_i - u_{l_i}\right)^T A_{l_i}\left(x_i - u_{l_i}\right) \qquad (12)$$

A_{l_i} is a m x m is a diagonal matrix called covariance matrix. The distance function $D'\left(x_i, x_j\right)$ between two points mixes distance metrics from both Gaussians to be adequate:

$$D'\left(x_i, x_j\right) = \left\|x_i - x_j\right\|^2_{A_{l_i}} + \left\|x_i - x_j\right\|^2_{A_{l_j}} \qquad (13)$$

The optimization process still goes though an EM algorithm. The only difference is M-Step. In M-Step, we not only update centroids but also update covariance matrices for clusters.

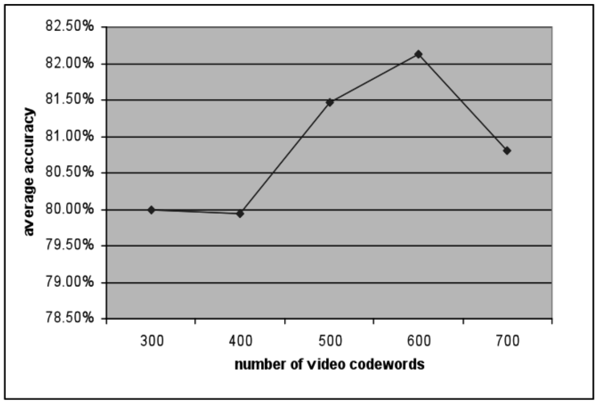

Figure 3: Classification performance using varying sizes codebooks in the KTH dataset.

The size of the codebook is determined by cross validation on the KTH human action dataset. The KTH human motion dataset is widely used to evaluate event detection and recognition (Schuldt 2004). It's also the largest available video dataset of human actions for researchers to evaluate and compare with. The dataset contains six types of human actions (walking, jogging, running, boxing, hand waving, and hand clapping) performed by 25 different persons. Each person performs the same action four times under four different scenarios (outdoor, outdoor with different scale, outdoor with camera moving, and indoor). We performed leave-one-subject-out cross-validation to evaluate the size of the codebook. Figure 3 shows the recognition performance of using different sizes of video codebooks. The result shows there is a peak to achieve best performance (600 in KTH dataset). Too many video code words or too few video code words will all hurt the recognition performance.

Behavior classification

Human behaviors can vary greatly in global appearance. We may therefore extract a different number of video cubes from behavior sequences. This is a challenging problem in building behavior descriptor access by machine learning algorithms. The video codebook allows us to borrow the idea from document classification in building behavior descriptors. For each code in the video codebook, we can treat it as a word in documents. In text classification, documents with different lengths are represented by a bag-of-words, which contains the frequencies of each word within a limited-size vocabulary. In our case, we can map extracted video cubes to their closest code word.

A behavior is represented by a histogram of all local binary features within a region of interest. The histogram is generated on the basis of the codebook, where code words are used as bins. Each local binary feature is mapped to its closest code word and added into the associated bin. We eventually normalize the counts in bins into frequencies.

This descriptor does not consider the spatial correlations among local features, because the spatial information has somehow been used in the clustering step. A behavior descriptor is treated as a vector with the same size as the codebook.

Due to the rarity of aggressive behaviors in real life in comparison to normal behaviors, we use a one-center SVM to train a model for all normal behaviors and detect aggressive behaviors as outliers.

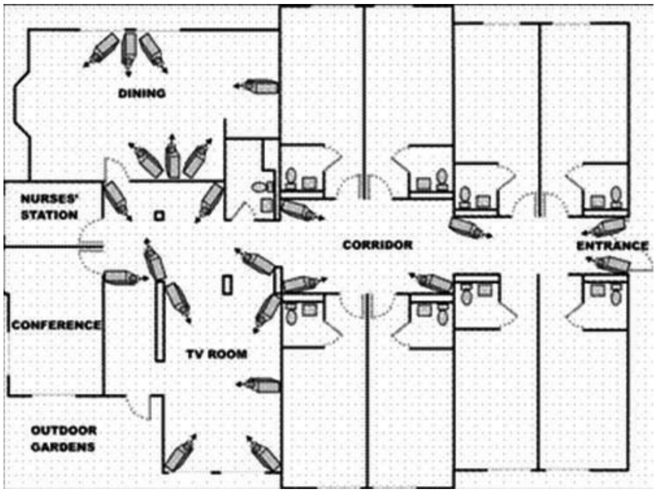

Figure 4: Camera placement in the nursing home

Experiments

We evaluate our algorithm using the CareMedia aggression dataset (Caremedia 2005), that was collected from a real world surveillance video application. CareMedia dataset is a surveillance video data collection from a dementia unit within a nursing home. The unit has 15 residents served by 4 nurses and a number of assistants. We placed 24 cameras in public areas such as the dinning room, TV room and hallways. Figure 4 shows the camera set up in the nursing home. We recorded resident life 24 hours daily for 26 days via those 24 cameras. The recording set up is 640x480 resolution and 30 frames per second of MPEG2 recording. We collected over 13,000-hours of video which was about 25 terabytes.

We demonstrate the robustness of our algorithm in recognizing aggressive behaviors in the CareMedia dataset. Forty-two physically aggressive behavior video clips and 1074 physically non-aggressive behavior video clips recorded in a dining room with multi camera views were labeled for training and testing. We used 1000 non-aggressive behavior video clips for training and the remaining 116 (42 + 74) clips for testing.

We smoothed input videos by a Gaussian filter with zero mean and variances (5, 5, 10) and extracted 5x5x10 video cubes from the interest point. Each video cube was first converted into a binary cube and then represented by local binary features. ROIs in the Caremedia dataset were labeled manually. We created a local binary behavior descriptor for each ROI in each video clip using a 600-word codebook.

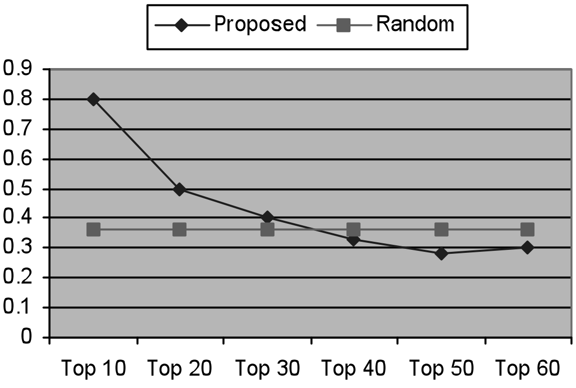

Figure 6. Aggression retrieval accuracy.

Figure 6 shows the performance of the proposed algorithm in recognizing aggressive behaviors. The top 10 retrieval results include about 80% aggressive behaviors, which is much better than the random accuracy 36.2%.

Figure 7 shows some frames extracted from the top 10 retrieval results. These behaviors involve large and colorful objects such as chairs and signs and can be well recognized by the proposed algorithm. Figure 8 shows some examples from the last 20 retrieval results. We can see that aggressive behaviors here are either occluded or only involve small objects that are difficult to notice even for humans. We also observed that many "aggressive behaviors" would not have been truly aggressive if they did not involve an object, i.e., spoon or chair. Recognizing subtle forms of aggressive behavior will require more than human kinemics models alone. Our approach, on the other hand, is able to model the action of the arm, body, and the object together.

Conclusions

We have demonstrated the feasibility of using an AI-based approach and digital video to automatically classify RRPA behaviors in a long-term care nursing home facility. In contrast to the current self and caregiver aggression reporting and tabulating methods, our approach has many advantages in observing incidents over extended periods in multiple areas, in enabling the recall and examination of individual aggression incident details, in determining antecedent and consequent events, in reducing the workload of caregivers and improving the overall quality of care and safety of the residents. In the coming future, we will extend the LBMD method into multiple spatial and temporal scales.

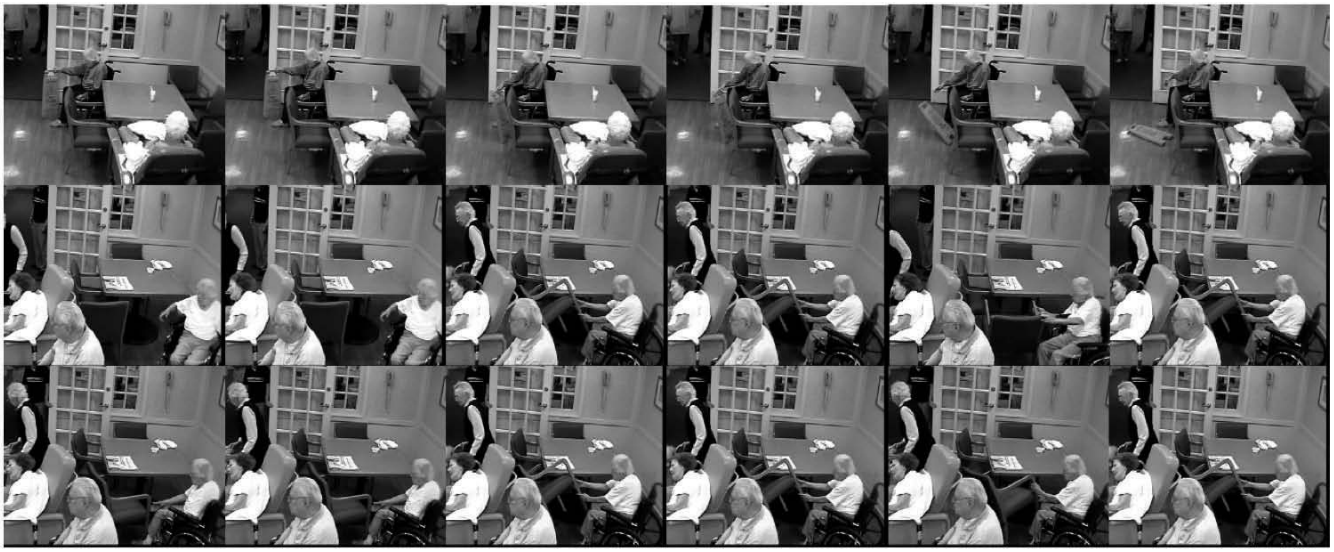

Figure 7. Some aggressions in the top 10 retrieval results

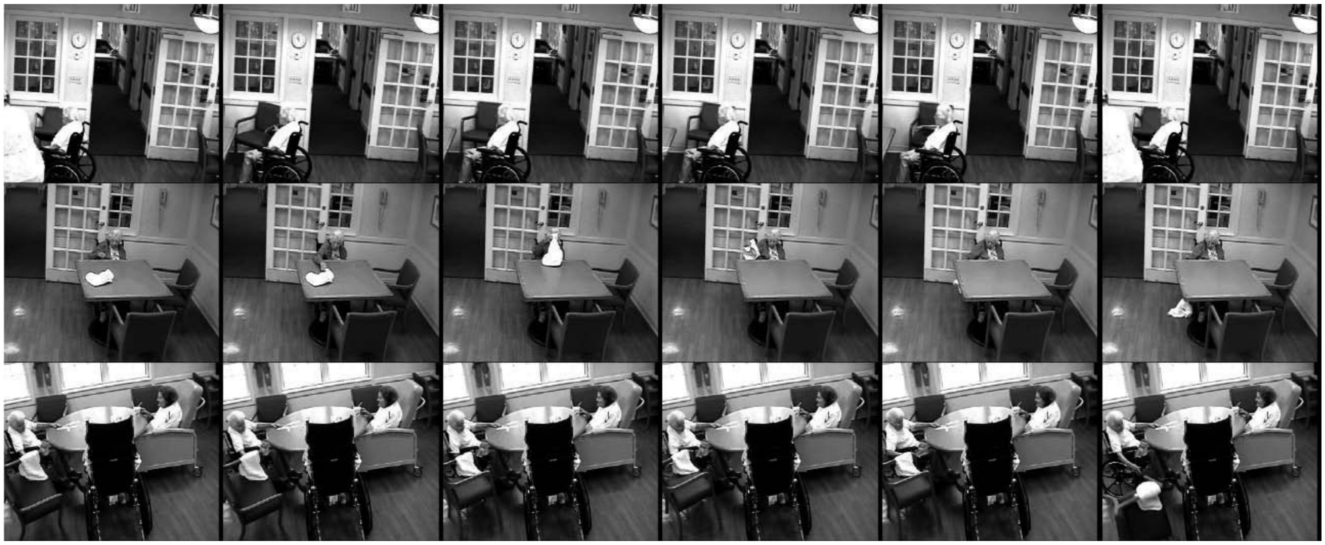

Figure 8. Some aggressions in the last 20 retrieval results.

References

Agarwal, S., Awan, A., and Roth, D. 2004, Learning to detect objects in images via a sparse, part-based representation, PAMI, November 2004

Akita, K. 1984, Image sequence analysis of real world human motion, Pattern Recognition, 17(1):73-83, 1984

Hogg, D. 1983, Model-based vision: a program to see a walking person. Image and Vision Computing, 1(1):5-20,

Basu, S., Bilenko, M., Banerjess, A. and Mooney, R.J. 2006, Probabilistic Semi-Supervised Clustering with Constraints, In Semi-Supervised Learning, MIT Press, 2006.

Bobick, A.F. and Davis, J.W. 2001, The recognition of human movement using temporal templates. IEEE Trans. PAMI, 2001

Bregler, C. 1997, Learning and recognizing human dynamics in video sequences, In CVPR, San Juan, Puerto Rico, June 1997

Caremedia http://www.informedia.cs.cmu.edu/caremedia.

Chen, M. Long Term Activity Analysis in Surveillance Video Archives, TR Language Technologies Institute, Carnegie Mellon University, March 2008

Dollár, P., Rabaud, V, Gottrell, G. and Belongie, S. 2005. Behavior Recognition via Sparse Spatio-Temporal Features, In VS-PETS 2005, page 65-72

Deutscher, J., Blake, A. and Reid, I. Articulated body motion capture by annealed particle filtering. In *IEEE CVPR*, volume 2, pages 126–133, 2000.

Farrell Miller M. Physically aggressive resident behavior during hygienic care. J Gerontol Nurs 1997;23:24-39.

Fergus, R., Perona, P., and Zisserman, A. 2003, Object class recognition by unsupervised scale-invariant learning, *In CVPR,* 2003

Hoeffer B, Talerico KA, Rasin J, Mitchell M, Stewart BJ, McKenzie D, Barrick AL, Rader J, Sloane PD. Assisting cognitively impaired nursing home residents with bathing: effects of two bathing interventions on caregiving. Gerontologist 2006;46:524-532.

Hu, W., Tan, T., Wang, L., and Maybank, S. A Survey on Visual Surveillance of Object Motion and Behaviors, IEEE Trans. SMC 3(34), Aug. 2004

Johnson E. Violence in nursing homes a growing concern. Available at: http://www.myinky.com/ecp/news/article/0,1626,ECP_734_1581923,00.html, December 2, 2002.

Ke, Y., Sukthankar R., and Hebert, M. 2005, Efficient visual event detection using volumetric features. In ICCV, p. 166-173, 2005

Lachs M, Bachman R, Williams CS, O'Leary JR. Resident-to-resident elder mistreatment and police contact in nursing homes: findings from a population-based cohort. J Am Geriatr Soc 2007;55:840-845.

Laptev, I. and Lindeberg, T. 2003, Space-time interest points, In ICCV, p. 432-439, 2003

Lee, MW, Cohen, I. and Jung, SK. Particle filter with analytical inference for human body tracking. In *IEEE Workshop on Motion and Video Computing*, pages 159–165, 2002.

Lowe, D.G., 2004, Distinctive image features from scale invariant key points, IJCV, November 2004

Niebles, J.C., Wang, H., Li,. F. Unsupervised learning of human action categories using spatial-temporal words. 1983

Polana, R., and Nelson, R. 1994, Low level recognition of human motion (or how to get your man without finding his body parts). In Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects, p. 77-82, Austin TX, 1994

Schuldt, C., Laptev, I., and Caputo, B. Recognizing human actions: A local SVM approach. In ICPR, pp: 32–36, 2004. BMVC 2006.

Sherrell K, Anderson R, Buckwalter K. Invisible residents: the chronically mentally ill elderly in nursing homes. Arch Psychiatr Nurs 1998;12:131-139.

Shinoda-Tagawa T, Leonard R, Pontikas J, McDonough JE, Allen D, Dreyer PI. Resident-to-resident violent incidents in nursing homes. JAMA 2004;291:591-598.

Yang, J., Jiang Y.G., Hauptmann, A. and Ngo, C.W. 2007, Evaluating bag-of-visual-word representation in scene classification, MIR'07 ACMMM, September 2007

Wang, X., Wu, C. Approach of automatic multithreshold image segmation, in the 3rd World Congress on Intelligent Control and Automation, June 2000.

Yamato, J., Ohya, J., and Ishii, K. 1992, Recognizing human action in time-sequential images using Hiden Markov Model, In CVPR, p. 379-385, Champaign, IL, June 1992