Dynamics Based Approach for Human Activity Understanding

A Thesis Presented
by
Teresa Yu Mao
to
The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of
Master of Science
in
Electrical and Computer Engineering
in the field of
Control Systems and Signal Processing

Northeastern University
Boston, Massachusetts
December 2011
In recent years, there has been an increasing interest within computer vision in the analysis of human activity for surveillance applications. These efforts are motivated by ubiquity of surveillance cameras and the need for security in large public spaces. The goal of human activity recognition from video is to classify an activity in a given video as one of several activities learned from training data. A related problem, event and anomaly detection, flags a behavior or event as abnormal when it deviates from previous available data. In this case, the activity is not known a priori. Instead, the goal is to look for something that has not been seen before.

In this thesis, we propose a new approach to exploit the temporal information embedded in video data to address problems in human activity analysis. The main idea is to model human behaviors as output of unknown dynamical systems while the initial conditions are unknown. We use Mixture of Gaussian to determine outliers, which are labeled as anomalies. We will introduce this approach in the context of activity recognition, event detection and anomaly detection.
Acknowledgements

I would like to express my gratitude to my advisor, Prof. Octavia Camps for her efforts in advising me on this thesis, research, academia and life. Prof. Octavia Camps’ encouragement, patience, and ideas have been invaluable in my research. Her extensive wealth of knowledge and enthusiasm for research has inspired me and taught me how to do research efficiently.

I would also like to acknowledge Prof. Mario Sznaier and Prof. Jennifer Dy who inspired ideas about research. This thesis would not have been possible without their help.

Finally, I would like to thank the students in the Robust System Lab: Binlong Li, Fei Xiong, Mustafa Hacettepeli, Necmiye Ozay and Caglayan Dicle provided me valuable suggestions when I was working on my thesis.

Boston, Massachusetts

Teresa Mao
# Table of Contents

Abstract

Acknowledgements

1 Introduction 1
  1.1 Introduction 1
  1.2 Thesis overview 2
  1.3 Previous work 4

2 Feature Selection 7
  2.1 Histogram of Gradients 7
  2.2 HOG Implementation Details 9
  2.3 Space-Time Interest Points 12

3 Hankel Matrix 15
  3.1 Dynamic Systems 15
  3.2 Linear Time Invariant Systems 15
  3.3 Hankel Matrix 16
  3.4 Rank 17

4 Dynamic Subspace Angles 19
  4.1 Overview 19
  4.2 Theoretical Details 19

5 Mixture of Gaussians 23

6 Human Activity Recognition 27
  6.1 Introduction 27
  6.2 Human Detection 28
  6.3 Constructing Feature Vector 31
  6.4 Supervised Learning Method 35
7 Anomaly Detection 37
  7.1 Introduction 37
  7.2 Datasets 39
  7.3 Experiments 40
  7.4 Remarks 45

8 Future work 47

Bibliography 49
Chapter 1

Introduction

1.1 Introduction

Visual surveillance devices have long been in use to gather information and to monitor human actions, events and activities. The three most widely used devices in the visual surveillance market are CCD and CMOS cameras, thermal cameras and night vision. Surveillance systems are prevalent in many homeland security sponsor scenarios such as boarder security, perimeter security and airport security. Unfortunately, the manpower required to monitor and analyze surveillance videos is often expensive. Therefore, these surveillance videos are usually not monitored, they are kept as archives. Surveillance cameras can be better used. They can detect events requiring attention as they happen and prompt authorities to take action in real time. The main goal of visual surveillance is to develop intelligent visual surveillance to replace the ineffective passive video surveillance. Visual surveillance should be actively and effectively performing surveillance task as automatically as possible.

As shown in the figure below, the first level of general video surveillance system, features (i.e. areas of motion) are extracted. Based on these extracted features, object can be detected and tracked. At the second level, events in which the
objects participate can be recognized. A selected representation of events is used to define concepts and relations in the context of human activity monitoring.

1.2 Thesis Overview

In the recent years, there has been an increasing interest within computer vision in the analysis of human activity for surveillance applications. These efforts are motivated by ubiquity of surveillance cameras and the need for security in large public spaces. This thesis is on building robust surveillance system, it will cover human activity recognition, event detection and anomaly detection. My research started with human activity recognition, experimenting with simple activities in
the KTH dataset (walking, running, jogging, waving, clapping and boxing). From that I worked on event detection, experimented with CAVIAR dataset (fighting, left bag, split from groups... etc). There exist restrictions with activity recognition and event detection in the context of surveillance applications. For example, we will not always have a model for every possible behavior or event, not all behaviors and events are recognizable. Therefore, it is difficult to evaluated if an activity or event in the context of public safety. The goal of surveillance systems is to detect dangerous or interesting activities instead of normal activities. Not to say that dangerous activities can’t be consist of normal activities, but normal environment analysis isn’t the goal here. Therefore it is more important to be able to detect anomalies. So lastly, we will discuss my research in anomaly detection, UMN crowd panic dataset.

In Chapter 2, I will go over popular human activity recognition features such as
HOG and popular spatio-temporal features such as STIP. In Chapter 3, provides an algorithm overview and how system dynamic can be used to solve activity recognition, event detection and anomaly detection problems. In Chapter 4, I will introduce and explain the importance of subspace angles and how it works with Hankel matrices. In Chapter 5, I will introduce Mixture of Gaussians and how it can be used for anomaly detection. In Chapter 6, experimental results: human activity recognition. In Chapter 7, experimental results: anomaly detection. In Chapter 9, conclude this thesis with conclusions and future work.

1.3 Previous Work

1.3.1 Human Activity Recognition

Human activity recognition from video has been central not only to surveillance applications but also assisted living for elderly and human computer interfaces. Currently there are three major approaches to modeling and recognizing actions of single actors: nonparametric, volumetric, and parametric time series approaches.

Nonparametric methods depend on features extracted at the frame level, then matched to the existing templates. Volumetric approaches extract volume of pixels and use 3D local features from 2D features such as corners and spatial-temporal filter responses. However, non-parametric and volumetric approaches are restricted by the “local” nature of the features use and lack information among features across frames. In contrast, the parametric time series approach
(i.e. hidden Markov models and linear dynamical systems) use dynamical models of motion to exploit the temporal relationships in video sequences.

### 1.3.2 Anomaly detection

There has been wide interest in surveillance application and computer vision to analyze densely crowded environments. Most of these efforts are motivated by the ubiquity of surveillance cameras, and the importance of crowd monitoring in various applications. The goal is not to analyze normal crowd behavior but to detect deviations from the normal events or behaviors.

Trajectory based anomaly detection comprises of tracking each object in the scene and learning model. Basharat uses tracks of every object in the scene as input for their framework while Siebel and Maybank uses three part fusion tracker: active shape tracker, region tracker and head detector for initialization. And Zhang and Lu, Cluster each class of trajectories with their spatial distribution (trajectories’ parameters and directions). It is important to note that trajectory based modeling is difficult and impractical for densely crowded scenes.

To avoid having to deal with tracking, various authors have proposed motion representation for anomaly detection. The most popular motion based approach is some form of spatio-temporal gradients (i.e. dense optical flow). Adam et al. uses histograms to find probabilities of optical flow in local regions. Kim and Grauman uses mixture of probabilistic PCA models to model local optical flow patterns. They also use a Markov Random Field (MRF) to enforce global
consistency. Mehran et al. draw inspiration from classical studies of crowd behavior that characterize crowd behavior using social force concepts.
Chapter 2

Feature Selection

2.1 Histogram of Oriented Gradients

In 2005, Dalal and Trigg reviewed the existing edge and gradient based descriptors for human detection and found that grids of histograms of Oriented Gradients (HOG) descriptors significantly outperforms the other feature sets for human detection at the time. Fine-scale gradients, fine orientation binning, relatively coarse spatial binning and high-quality local contrast normalization in overlapping descriptor blocks are important for good results. HOG not only outperforms wavelets [17,22], it is good at dealing with variable appearance and poses, cluttered backgrounds and illumination variations.

HOG descriptors reminiscent of edge orientation histograms [4,5], SIFT descriptor [12] and shape contexts [1]. The idea is that the local object’s appearance and shape can be characterized well by examining distribution of local intensity, gradients or edge direction without precise knowledge of the corresponding gradient or edge position. This method evaluates well-normalized local histograms of image gradient orientations in a dense grid. Similar features have seen increasing use over the past decade [4,5,12,15].
In practice this is implemented by dividing the image window into small spatial regions “cells”, for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell; the combined histogram entries form this presentation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram “energy” over a larger spatial regions “blocks” and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as Histogram of Oriented Gradient (HOG) descriptors.

For human detection, Tiling the detection window with a dense (in fact, overlapping) grid of HOG descriptors and using the combined feature vector in a conventional SVM based window classifier gives our human detection chain.

Comparing with Precursors

SIFT-style approaches perform remarkably well in this application [12,14]. The use of orientation histograms has many precursors [13,4,5], but they only work well when combined with local spatial histogramming and normalization in Lowe's Scale Invariant Feature Transformation (SIFT). [12], in which it provides the underlying image patch descriptor for matching scale-invariant keypoints. The Shape Context work [1] studied alternative cell and block shapes, albeit initially using only edge pixel counts
without the orientation histogramming that makes this representation effective. The success of these sparse feature based representations overshadows the power and simplicity of HOG's as dense image descriptors. In particular, experiments suggest that even the best current keypoint based approaches are likely to have false positive rates at least 1.2 orders of magnitude higher than our dense grid approach for human detection, mainly because none of the keypoint detectors that we are aware of detect human body structures reliably.

The HOG/SIFT representation has several advantages.

It captures edge or gradient structure that is very characteristic of local shape, it does so in a local representation with an easily controllable degree of invariance to local geometric and photometric transformations: translations or rotations make little difference if they are much smaller that the local spatial or orientation bin size.

For human detection, rather coarse spatial sampling, _ne orientation sampling and strong local photometric normalization turns out to be the best strategy, presumably because it permits limbs and body segments to change appearance and move from side to side quite a lot provided that they maintain a roughly upright orientation.

### 2.2 HOG Implementation Details

The main conclusions are that for good performance, one should use _ne scale derivatives (essentially no smoothing), many orientation bins, and moderately sized, strongly normalized, overlapping descriptor blocks. (fig 4)

1. Gamma/Colour Normalization
After evaluating several input pixel representation, RGB and LAB color spaces give comparable results, but grayscale reduces performance by 1.5%. Square root gamma compression at each color channel improves performance by 1%, but log compression is too strong and worsens it by 2%.

2. Gradient Computation

Detector performance is sensitive depending on how gradients are computed and the simplest scheme turns out to be the best. Simple 1-D [-1, 0, 1] masks at sigma = 0 work the best. Larger masks always decrease performance and smoothing damages it significantly. For Gaussian derivatives, moving from \( \sigma = 0 \) to \( \sigma = 2 \) reduces the recall rate from 89% to 80%. At \( \sigma = 0 \), cubic corrected 1-D width 5
_lters are about 1% worse than [1; 0; 1], while the 2_2 diagonal masks are 1.5% worse. Using uncentered [1; 1] derivative masks also decreases performance (by 1.5%) presumably because orientation estimation suffers as a result of the x and y _lters being based at different centers. For color images, we calculate separate gradients for each color channel, and take the one with the largest norm as the pixel's gradient vector.

3. Spatial/Orientation

Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centred on it, and the votes are accumulated into orientation bins over local spatial regions that we call cells.

Cells can be either rectangular or radial (log-polar sectors). For unsigned gradient, the orientation bins are evenly spaced over 0_180_, for signed gradient, the orientation bins are evenly spaced over 0_360_. To reduce aliasing, votes are interpolated bilinearly between the neighbouring bin centres in both orientation and position.

In practice, the vote is a function of the magnitude gives the best result.

In practice, using the magnitude itself gives the best results. Taking the square root reduces performance slightly, while using binary edge presence voting decreases it significantly by 5%. Fine orientation coding turns out to be essential for good performance, whereas (see below) spatial binning can be rather coarse.
For humans, the wide range of clothing and background colors presumably makes the signs of contrasts uninformative. However note that including sign information does help substantially in some other object recognition tasks, e.g. cars, motorbikes.

4. Normalization and Descriptor Blocks

Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization is essential for good performance.

We evaluated a number of different normalization schemes. Most of them are based on grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window.

In fact, we typically overlap the blocks so that each scalar cell response contributes several components to the final descriptor vector, each normalized with respect to a different block. This may seem redundant but good normalization is critical and including overlap significantly improves the performance.

2.3 Space-Time Interest Points

As shown by Lazebnik et al [9], a coarse description of the spatial layout of the scene can improve recognition results. Successful extensions of this idea includes the optimization
of weights for the individual pyramid levels and the use of more general spatial grids. Lapev build on these ideas and go a step further by building space-time grids.

**Space-time features**

Sparse space-time features have recently shown good performance for action recognition [3,6,13,15]. They provide a compact representation and robust to background clutter, occlusions and scale changes. These features are inspired by [7], detects interest point using a space-time extension of the Harris operator. Use a multi-scale approach and extract features at multiple levels of spatio-temporal scales. Instead of select a specific scale in [7], we use a multi-scale approach and extract features at multiple levels of spatio-temporal scales. Not only does this reduce computational complexity, the independence from the scale selection artifacts and evidence of good recognition using dense scale sampling.

To characterize motion and appearance with these local features, we compute histogram descriptors of space-time volumes in the neighborhood of detected points. The size of each volume is subdivided into a grid of cuboids; for each cuboid we computer coarse histogram of gradient (HoG) and optic flow (HoF). There Hog and HoF descriptors are normalized histograms that are concatenated into vectors, and are similar in spirit to the well known SIFT descriptor.

Interestingly, HoG performs better than HOF for almost all of the action in the “real-world action” set. The inverse is true for the KTH action set. This shows that the
context and the image content play a large role in realistic setting, while simple action can be well characterized by their motions only.
Chapter 3

Hankel Matrix

3.1 Dynamic Systems

The advantage of working with videos is that they are temporally ordered data, therefore dynamical systems are a powerful tool when working with videos. Using dynamical systems for temporally ordered data has been used in several applications such as activity recognition, tracking, dynamic textures and other computer vision applications.

The main idea is to use a dynamical system to do dimensional reduction. We model temporal changes of a measurement vector as a function of a low dimensional state vector that changes over time. This means dynamical model can use as generative model (to predict future data) and nominal model (to recognize and classify data). In this thesis, I will use dynamical model a nominal model to recognize activity, events and anomalies.

3.2 Linear Time Invariant (LTI) Systems
The simplest dynamical model is a Linear-Time Invariant (LTI) system. Let’s first consider a single-input single output (SISO), LTI dynamic system. The state spaced model is defined as:

\[
\begin{align*}
\begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & C \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \\
&= Ax(t) + Bu(t) \\
y(t) &= Cx(t)
\end{align*}
\]  

(1)

with initialization \( x(0) \)

where \( A \) is \( N \times N \) matrix, \( B \) is \( Nx1 \) matrix, \( C \) is \( 1xN \) matrix.

A and \( C \) are constant over time, and \( Bu(t) \) is uncorrelated zero mean Gaussian measurement noise. This is state-space model is both controllable and observable, which means it has the same input-output behaviors as the transfer function of the system. This is called the minimal realization of the transfer function. The dimension of the state vector is the order of the system and also the measure of its complexity.

One important limitation of models in form (1), one must assume or estimate the dimensions and values of \( A, C, \) and \( x(o) \). To solve for \( A, C, \) and \( x(o) \) is a non convex problem. To avoid dealing with having to solve non convex problem. We will not work directly with the model in form (1), instead we will work with block Hankel Matrices.

### 3.3 Hankel Matrix

Given a system’s output \( (y_0, y_1..) \), it’s associated Hankel matrix is
In state-space system identification theory, the Hankel matrix appears prior to model realization. Traditionally, one identifies from the Markov parameters of input-output data. The Hankel Matrix comprises of the Markov parameters arranged in a specific Toeplitz pattern. Much efforts have been placed on the problem of obtaining the Markov parameters from input-output data by time or frequency domain approaches. Once the Markov parameters are determined, they become entries in the Hankel matrix for state-space identification. This approach is effective in detecting the order of the system, which means it is capable of producing relatively low-dimensional state-space model.

3.4 Rank (Order of System)

While the rank of the covariance matrix plays a central role in many statistical methods. The rank of a Hankel matrix has similar significance in model identification problems in system theory and signal processing.

The minimization of the rank of a Hankel matrix is particularly useful in designing a low-order LTI (linear, time invariant) system directly from convex specification on its impulse response. For example, we would like to find a linear system with
the lowest order that fits upper and lower bounds on \( n \) samples of a step response.

\[
\text{Rank}[H_f] < n + \text{no}, \quad n > \text{no}
\]

Moreover, if the impulse response excited all the modes of the system and \( n >> \text{no} \), the equality holds.

We can readily extend this problem to Multiple Input Multiple Output (MIMO) problem by using block Hankel matrices. It is well known that the rank of the Hankel matrix is the order of the system. Extension of work of Ho and Kalman (2). In state-space realization methods, Hankel matrix plays a critical role because order of the model can be obtained from singular value decomposition (SVD) of the Hankel matrix. With perfect noise-free data, the minimum order realization can be easily obtained by keeping only the non-zero Hankel singular values. However, in real or noise-contaminated data, The Hankel matrix tends to be full rank, thus making problems of determining minimum-order state space model difficult. In this case, we can general observe that there is a significant drop in singular values that represents to rank of the system.
Chapter 4

Dynamic Subspace Angles

4.1 Overview

The subspace angle of two subspaces is the amount of new information between subspace A and subspace B while not associated with statistical errors of fluctuations. For our application, we define each subspace as the following.

1) \([U \ D \ V] = \text{svd}(\text{Hankel})\)

2) \(\text{subspace}_a = U(1:\text{rank},:)\)

\(\text{subspace}_b = U(1:\text{rank},:))\)

If the angle between the two subspaces is small, the two spaces are nearly linearly dependent.

4.2 Theoretical Details

I. Introduction

Let \(F\) and \(G\) be subspaces of unitary space \(E\), assume that

\(p = \text{dim} \ (F) > \text{dim} \ (G) = q > 1\)
The smallest angle between F, G $\in [0, \pi/2]$

This angle is the smallest angle between the orthogonal complement of F with respect to $u_1$, and G with respect to $v_1$

$$\cos \theta_i = \max_{u \in F} \max_{v \in G} \frac{u^H v}{||u||_2 ||v||_2}, \quad ||u||_2 = 1, \quad ||v||_2 = 1.$$  

II. Canonical Correlations

Note: $R(A) = \text{range of } A$ & $N(A)$ is the nullspace of A

In the problem of canonical correlations we have $F = R(A)$, $G = R(B)$. Where A and B are rectangular matrices. The canonical correlation is to do cosine on the angle. The canonical correlations greater or equal to zero are basically the eigenvalues of $y_k, z_k$. Where $u_k = A y_k$ and $v_k = B z_k$.

When A and B have full column rank, the canonical correlation is computed by $A^H A$, $B^H B$, $A^H B$ and perform the Choleski decompositions.

III. Solution Using Singular Values

In most applications, each subspace is defined as range or some variation of range of a matrix. In this case, a unitary basis for the subspace can be computed by well-known methods for QR-decomposition.

Given a $m \times n$ matrix A, where $m > n$, a decomposition
Recently, an efficient and numerically stable algorithm for computer singular value decomposition (SVD) of matrix has been developed. We use this to compute principle angles and vectors.

\[ A = (Q' | Q'')(S | 0)(p \times n)(m - p) \times n \]

IV. Numerical methods

In this section, we assume that the columns of matrix A and B are linearly independent. To get the orthogonal bases for F and G, we need the to do QR decomposition of matrices A and B. There are two methods to do this: 1) Householder triangularization (HT) 2) modified Gram-Schmidt (MGS) method. MGS has an advantage over HT, the total number of multiplications is less than that of HT.

\[ \text{MGS: } p^2 m, \quad \text{HT: } 2p^2(m - p/3). \]
if only the principle angles are wanted, then the number of multiplications in SVD is \(2q^2(p - q/3)\). Therefore when \(m >> p\), MGS requires only half as much as HT.

However, MGS does have its problems. When \(A = B\),

\[
\| I - \bar{Q}_A^T \bar{Q}_A \|_2 \leq 2p(p + 1)\kappa(A)2^{-1}.
\]

singular values of \(M = QaQa\) may not be near one when large. Only \(A\) not exactly equal to \(B\), then the rounding error comput \(Qa\) and \(Qb\) will not be correlated and in ill-conditioned cases, we will not get all angeles near zero either with HT or MGS.

V. The Singular Case

When \(A\) and/or \(B\) does not have full column rank, small perturbations in \(A\) and \(B\) will change rank of \(A\) and/or \(B\). The main difficulty is finding the correct rank for \(A\) and \(B\). The best way of solving this problem is to use singular vector decompoision (SVD)

We can also use SVD for non-singular cases, but the computation cost of SVD is more expensive than computing the corresponding QR-decomposition. To use QR-methods in the singular case, the column pivoting must be used.
Chapter 5

Mixture of Gaussians

5.1 Stauffer and Grimson’s Mixture of Gaussians

Mixture of Gaussian is a probabilistic method for classifying normal and abnormal events. It involves modeling subspace angles over time as a mixture model. This method is stable and robust enough for real-time applications. Mixture of Gaussians only requires two parameters, \((o, T)\), which are robust to different cameras and different scene settings.

Mixture of Gaussian was originally proposed by Stauffer and Grimson a method for background subtraction. One may think of anomaly detection as a background subtraction problem. The background is like the normal events, which we are not interested in. The foreground is like the anomalies, which we are interested in and are the information we extract as useful from the video sequence.
Use Mixture of Gaussian to monitor activity of a site over extended period of time. This detects patterns of the motion and interaction demonstrated by objects in the site

1) Provide statistical description of typical activity pattern (normal behavior)

2) Detects unusual events, by spotting activity that is very different from normal activity pattern.

3) Should detect unusually interactions between objects.

We model recent history as a mixture of K Gaussian distributions. The probability of observing the current value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \ast \eta(X_t, \mu_{i,t}, \Sigma_{i,t}),$$

where $K$ is the number of distributions, $\omega_{i,t}$ is an estimate of the weight (the portion of the data accounted for by this Gaussian) of the $i$th Gaussian in the mixture at time $t$, $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and covariance matrix of the $i$th Gaussian in the mixture at time $t$, and where $\eta$ is a Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^\frac{1}{2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)}.$$ 

It is recommend to use $K= 3$ to $5$
To avoid costly matrix inversion, we assume that R, G, B pixel values are independent and have the same variance.

The distribution of recently observed values is characterized by mixture of Gaussian. A new value will usually be represented by one of the major component of the mixture model and used to update the model.

Expectation Maximization (EM) is commonly used for maximizing the likelihood of the observed data. Every new value, $X_t$ is checked against existing $K$ Gaussian distribution until a match is found as a pixel value within 2.5 standard deviation of a distribution.

If none of the $K$ distributions match the current pixel value, the least probable distribution is replaced with a distribution with current value as its mean, an initially high variance, and low prior weight.

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}).$$

where alpha is the learning rate and $M$ is 1 for the model which matched and 0 for the remaining modles. After this approximation, the weights are renormalized. $1/\alpha$ defines the time constant which determines change. $W$ is effectively the causal low-pass filtered average of the posterior probability that pixel values have matched model $k$ given observation from time 1 through $t$. 

$$\Sigma_{k,t} = \sigma^2_k I.$$
\[ \mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \]

\[ \sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho (X_t - \mu_t)^T (X_t - \mu_t), \]

where

\[ \rho = \alpha \eta(X_t | \mu_k, \sigma_k) \]

is the learning factor for adapting current distributions.
Chapter 6

Human Activity Recognition

This paper presents an algorithm for human activity recognition using multiclass SVM. Given a video with a person performing an activity, such as clapping, boxing, waving or walking, our supervised learning algorithm can recognize which activity this person is doing with relatively good results. For each frame, we use the distance from the center of the frame to the contour of the person at different angles. And for each video, we incorporate the relationship between the frames by using Hankel matrix to represent each video sequence as a dynamic system. We then use the SVD of this Hankel matrix as the input to the Support Vector Machine.

6.1 Introduction

We want machines to be able to recognize or understand what activities humans are doing. Human activity recognition is important for applications such as surveillance in public spaces, pedestrian on the streets [2], fall detection for older people [3]...etc. Most of current algorithms are carried out within one frame and neglect the data association between the frames in the time domain.
Generally there are three algorithms used for data associations: nearest neighbor, multiple hypothesis tracking and joint probability data association. In this project, we will use Hankel matrix for data association between the frames.

We will explore human activities such as clapping, waving, boxing and walking from the KTH dataset. The KTH dataset is the standard dataset used to compare human activity recognition algorithms within the research arena. Each video in this dataset contains a different person doing one of the four activities. Each person does each activity differently. We label, train and test the activities and not based on the particular person.

### 6.2 Human detection

The KTH dataset contains video of a person doing an activity in front of a relatively non-complex background. The goal of this step is to separate the human and the background, more specifically, we want to have a binary image where the white pixels represents the human and the black pixels represents the non-human or the background.
6.1.1 Background and Foreground

When it comes to separating foreground and background, many people use background subtraction. Simple background subtraction doesn’t work well with this dataset because we do not have the background image without something in the foreground and it assumes the parts of human that are stationary as part of the background. For example, if we use background subtraction for boxing, it results with only the arms and the other parts of the body are assumed to be parts of the background since they are the same between frame n and frame n+1. And if we use edge detection, we end up with more edges than we need.

The background in this dataset is not complex, but the pattern of the grass sometimes produces edges we don’t want. So to solve this problem we convolve the Gaussian filter with the Sobel edge detector so that only edges with very high gradients will be detected (Fig. 1). Gaussian filter smooths the image by
weighted average of pixels in the neighborhood. Sobel edge detects the edges in the image by calculating the derivatives. With the convolution of these two filters, we get a better result of having only the edges we need to be able to identify the foreground and the background.

Figure 1. (left most) original image, (left) vertical and horizontal filter results, (center), sum of vertical and horizontal filter results, (right most) binary image of the contour.

6.2.2 Find Contour with Snakes

The binary image we produced in the previous step gives some details other than the most outer contour of the person, we can get rid of the details with active contours / snakes. (Fig. 2) We only want the most outer contour of the person because this makes the implementation for measuring length of center to contour much easier later.

We start off by putting a bounding box around the binary image (Fig. 1) in
the previous step [3, 5]. We use this bounding box to initialize our snake. The framework behind Snakes is to minimize an energy associated to the current contour as a sum of an internal and external energy. The external energy is minimized when the snake is at the boundary position / place of highest gradients. The internal energy is minimized when the shape is as smooth as possible. Think of active contours as an elastic band that uses energy minimization to find the contour of the object.

\[ E = \sum_{i=1}^{N} a_i E_c(p_i) + b_i E_s(p_i) + c_i E_g(p_i) \]

Ec is continuity, distance between actually point and the average distance between points. Es is smoothness, the second derivative . Eg is edgeness, magnitude of the gradient. ai, bi, ci are weight for continuity, smoothness and edgeness.

\[ E_c(p_i) = (d - |p_i - p_{i-1}|)^2 \]
\[ E_s(p_i) = |p_{i-1} - 2p_i + p_{i-1}|^2 \]
\[ E_g(p_i) = -| \nabla I(p_i) | \]

Figure 2. (left) filtered image. (right) binary image

6.3 Constructing the Feature Vector
6.3.1 Length from reference point to contour

The idea of using the distance from reference point to contour was inspired by Wenrui Ding’s paper on Unsupervised Spatio-temporal Multi-Human Detection and Recognition in Complex Scene. In his paper, he proposed to use the length from the reference point (head) to the contour at varies angles [10,11,12] (Fig.3).

In this paper, for each frame of the video, we set the reference point as the
center of the image. And we measure the length of the ray from the reference point to the contour at every 30 degrees. In the actual implementation, we rotate each image 30 degrees and measure the length from reference point to the contour at 0 degrees. It scans from right to left, from the image border which is usually a black pixel. It keeps checking the pixel value and stops at the first white pixel it encounters. We then measure the distance between this pixel and the reference pixel, this distance is our length between the reference point to the contour. Since we measure the length at each 30 degrees, for each image, we end up a vector containing 12 numbers.

6.3.2 Hankel matrix and SVD

In this project, each video is made up of 30 consecutive images. From the previous step, we get a vector of 12 numbers for each image. To use these vectors to represent the video sequence, we use the Hankel matrix. It’s a square matrix with constant, positive sloping skew-diagonals [9]. The Hankel matrix allows you to arrange the length vector for each frame into partially overlapping segments and rearranging them into a matrix. This is especially useful because the Hankel matrix represents the dynamic order of the video sequence. In our case, for 30 frames, each with 12 length measurements, the hankel marix is 180 x 15 for each video.

$$H_f = \begin{bmatrix} f_1 & f_2 & \cdots & f_{n/2} \\ f_2 & f_3 & \cdots & f_{n/2+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n/2} & f_{n/2+1} & \cdots & f_{n-1} \end{bmatrix}$$
where \( f_n \) is the vector containing length information for frame \( n \).

The Hankel matrices are formed when given a sequence of output data and a realization of an underlying state-space is need. In other words, we are using the Hankel matrix to represent a dynamic system. The SVD of the Hankel matrix can still provide the dynamic order of the Hankel matrix [13].

\[
M = U \Sigma V^*
\]

Where \( \Sigma \) is a diagonal matrix containing the singular values. We then normalize \( \Sigma \), use this as the input to the SVM.
6.4. Supervised Learning Method

Human activity recognition using SVM is a new method in this area of research. SVM is based on the study of limited sample learning theory [1]. By the theory of structural risk minimization, it performs well for classification of limited sample size. SVM is released to establish an optimal hyperplane under the linear separable condition. As for nonlinear separable conditions, it uses a kernel [7] (i.e. quadratic, polynomial, and rbf) to map the problem from low dimension to higher dimensional feature space (Fig. 4). This optimal hyperplane leaves the largest possible fraction of points of same class on the same side and maximizes the distance of either class from the hyperplane.

Figure 4. use kernel to map from low dimension (right) to high dimensional feature space (left)

6.4.2 Multiclass SVM

There are two simple approaches to multiclass SVM, One vs. All and One vs. One. The idea of One vs. All multiclass SVM is: Given N classes, build N different binary classifier. For the ith classifier, let the positive examples be all the points in the class i and let the negative examples be all the points not in class i. The idea
of One vs. One multiclass SVM is to build $N (N - 1)$ classifiers, one classifier to distinguish each pair of classes $i$ and $j$. Let $f_{ij}$ be the classifier where class $i$ were positive examples and class $j$ were negative. We will look at both of these approaches in our experiments.

### Results

**Previous work:**

<table>
<thead>
<tr>
<th></th>
<th>HOG3D</th>
<th>HOG/HOF</th>
<th>HOG</th>
<th>HOF</th>
<th>Cuboids</th>
<th>ESURF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris3D</td>
<td>89.0%</td>
<td>91.8%</td>
<td>80.9%</td>
<td>92.1%</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Cuboids</td>
<td>90.0%</td>
<td>88.7%</td>
<td>82.3%</td>
<td>88.2%</td>
<td>89.1%</td>
<td>–</td>
</tr>
<tr>
<td>Hessian</td>
<td>84.6%</td>
<td>88.7%</td>
<td>77.7%</td>
<td>88.6%</td>
<td>–</td>
<td>81.4%</td>
</tr>
<tr>
<td>Dense</td>
<td>85.3%</td>
<td>86.1%</td>
<td>79.0%</td>
<td>88.0%</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Our method: tested clapping, waving and boxing. Using only decent segmentations.

**Average accuracy rate = 96.3%**
Chapter 7

Anomaly Detection

7.1 Introduction

Since the ubiquity of cameras, we are using cameras in wide variety of locations and applications, such as live traffic monitoring, parking lot surveillance, inside vehicles and intelligent spaces. These cameras offer data on a daily basis that help analysis behaviors and events that eventually protect us from danger or analyze past situations.

Unfortunately, most visual surveillance still depend on the human operator is monitor the surveillance video. It is tedious and tiring to monitor for potentially dangerous or interesting events that very rarely happens. There is also the problem that the human operator falls asleep and unable to detect these events when they do happen. Therefore it is necessary to be able to automate this process as much as possible to assist operators.

In the recent years, abnormal event detection has become increasingly popular due to it is critical roles in surveillance applications. The research issues focus on the following areas: 1) how to shorten training period 2) how to reduce computational complexity
Automatic behavior understanding from videos is a very challenging problem. 

- extraction of relevant visual information
- suitable representation of information
- interpretation of visual information for behavior learning and recognition

Further complicated by the variability and the unconstrained environments such as specific time, place, or activity scenario. Either someone defines the events of interest for a particular application or use machine learning techniques to automatically construct activity models, which is better suite for online analysis because it is supported by real data.

In general, previous approaches for abnormal event detection falls into two categories: tracking based and motion-based approaches. The tracking based approaches focus on the trajectories of moving objects. However, in complicated scenes, real-time tracking of all moving objects is too difficult (in terms of accuracy and speed) to achieve in real world scenarios. Therefore, many propose to use motion based approaches to address this problem.

Motion based approaches can be classified into two groups based on how the motion features are extracted. The first approach is the background subtraction based. The second approach is optical-flow based.

Homeland security and crime prevention are two major topics that would benefit from indoor and outdoor monitoring of critical infrastructures, highways, parking garages and other public spaces. With rich activity space, it is difficult to have general procedure that works well over a wide range of scenarios. Therefore we take this problem in a different angle. We want to detect events or events that are simply different from what
has been going on in the scene. Event and anomaly detection, flags a behavior or event as abnormal when it deviates from previous available data. In this case, the activity is not known a priori. Instead, the goal is to look for something that has not been seen before.

7.2 Datasets

UMN dataset

The normal and abnormal crowd videos from University of Minnesota. The dataset comprises the videos of 11 different scenarios of an escape event in 3 different indoor and outdoor scenes. Figure below shows sample frames of these scenes. Each video consists of an initial part of normal behavior and towards the end of the sequences the abnormal behavior occurs.
CAVIAR Dataset

For the CAVIAR project, there are videos recorded acting out different scenarios of interest. These include walking alone, meeting others, window shopping, entering and exiting shops, fighting and passing out and leaving a package in public place.

7.3 Experiments

We could do abnormal event detection following similar techniques as we used for human activity recognition. However, it is obvious that segmentation was extremely important when the distance features used in activity recognition. Event detection involves interaction between human and another human or object. This means it will besides segmentation, tracking will also be important. Segmentation and tracking are two of the most fundamental and hardest problems in computer vision. Therefore we will use Histogram of Gradients (HOG) to avoid complications of both of these problems.

7.3.1 Abnormal Event Detection

For abnormal event detection we want to detect when an abnormal event has occur, note that we do not what event occurred. The CAVIAR dataset does specify what event occurred however in this experiment, we will not identify the specific events. We simple want to know when something abnormal happened, doesn’t matter what it is.
For abnormal event detection, first we first extract HOG features from each frame of the video, then organize these features into a block Hankel matrix.

\[ H_y = \begin{bmatrix}
  y_0 & y_1 & y_2 & \cdots & y_m \\
  y_1 & y_2 & y_3 & \cdots & y_{m+1} \\
  y_2 & y_3 & y_4 & \cdots & y_{m+2} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  y_{m-1} & y_m & y_{m+1} & \cdots & y_{2m-1} \\
  y_m & y_{m+1} & y_{m+2} & \cdots & y_{2m}
\end{bmatrix} \]

Note: each of the \( y \) represents a feature vector for each frame.

\[ [U \ D \ V] = \text{SVD}(H_y) \]

To create subspaces of the features, we take the U component from SVD of the block Hankel matrix. Generally, the data is full rank but there’s a jump between the first and the second singular value of the SVD, therefore we assume rank 1 and define the first column of the U component as our subspace. The block Hankel matrix is created with 30 frames, Hankel matrix at time \( t \) and Hankel matrix at time \( t+1 \) contain overlapping frames. Think in term of sliding window, this way out subspace angle change will be gradual and not sudden. We want to see event build up.

Once we have the subspaces of \( t \) and \( t+1 \), we then take the subspace angle between these subspaces. The subspace angle between two video intervals calculate the amount of new information between A and B. The hypothesis that during normal intervals of the video, the subspace angles remains about the same; when abnormal event occurs, the subspace
becomes significantly different than it was during the normal interval of the video, therefore the subspace angle increases dramatically.

As you can see from the figure above, at time 42, the subspace angle peaks (this is when two people are fighting). At time 56, the subspace angles peaks again (this is because two people are running away).

Note: This work was done during my work at Mitre Corporation. Therefore only events of their interests were used for experiments.

7.3.2 Anomaly Detection for Crowded Scenes
For anomaly detection in crowded scenes, we use a more advanced HOG feature. The newer feature is described in detail in chapter 2. Space Time Interest Point (STIP), these

```
$ teebeasley:~$ ./stip-2.0-linux$ ./bin/stipdet -help
This program detects space-time interest points at multiple space-time scales and computes corresponding descriptors. See README for the type of point detectors/descriptors that are currently implemented.

Usage:

Input/Output options:
- `i`: input file with sample names; format:
  - `sample-name start-frame end-frame`
  - `** sample-name must be without any extension (e.g. .avi, .txt) **`
  - `** start/end-frame is optional **`
- `path`: path to video sequences
- `ext`: video extension e.g. avi, mpeg, etc. (default is avi)
- `fpath`: path to pre-detected feature files; file (e.g. sample-name.txt) format:
  - `point-type x y t sigma2 tau2 detector-confidence`
- `o`: file name for saving detected features/descriptors; feature format:
  - `point-type x_norm x_norm t_norm y x t sigma2 tau2 descriptor`

Detection options:
- `det`: feature detector to be used (default=harris3d)
  - `harris3d`: Harris3D detector
  - `dense`: Regular dense sampling of features in space and time
  - `sosverlap`: spatial overlap (default=50 percent)
  - `toverlap`: temporal overlap (default=50 percent)
- `nlev`: number of levels in spatial frame pyramid (default=3)
  - `factor 2 subsampling is used; for each pyramid level
    points are detected at four combinations of spatial
    and temporal scales obtained by Gaussian smoothing
    with spatial variance sigma2=[4.0,8.0] and
    temporal variance tau2=[2.0,4.0]`
- `plev0`: initial level of spatial frame pyramid (default=0)
- `kparam`: K parameter in Harris function (default=0.0005)
- `thresh`: threshold for omitting weak points (default=1.000e-09)
  - `** to get all interest points set to zero **`
- `border`: reject interest points within image boundary (default=5)

Descriptor options:
- `dscr`: type of descriptor [hoghof,hogf] (default=hoghof)
- `szf`: factor used to compute descriptor spatial patch size (default=9.0)
- `tszf`: factor used to compute descriptor temporal patch size (default=4.0)
  - `patch size along spatial/temporal dimensions is defined as
  patch_dim=szsf^2*sqrt(Gauss variance_dim)`

Other options:
- `h`: shows this message
- `vis`: [yes|no] visualization (default=yes)
- `stdout`: [yes|no] stdout output (default=no)
- `mode`: feature detection & description mode (default=0)
  - `0`: feature detection & description (harris3d or dense)
  - `1`: feature detection for external points (switch -fpath is required)
```
are basically temporal patches of HOG descriptor and/or HOF descriptor. From experiments, HOG descriptor works better than HOF descriptor. This is because people are constantly moving, the general optical flow is complicated to categorize. Whereas orientation is more detail in describing similar general motions.

Now similar to our techniques in abnormal event detection. First we first extract STIP features from each frame of the video, then organize these features into a block Hankel matrix.

To create subspaces of the features, we take the U component from SVD of the block Hankel matrix. The block Hankel matrix is created with 30 frames, Hankel matrix at time t and Hankel matrix at time t+1 contain overlapping frames. Think in term of sliding window, this way out subspace angle change will be gradual and not sudden. We want to see event build up.

Once we have the subspaces of t and t+1, we then take the subspace angle between these subspaces. The subspace angle between two video intervals calculate the amount of new information between A and B. The hypothesis that during normal intervals of the video, the subspace angles remains about the same; when abnormal event occurs, the subspace becomes significantly different than it was during the normal interval of the video, therefore the subspace angle increases dramatically.

Now that we have the subspace angles. We will model these subspace angle changes with mixture of Gaussians. Use Mixture of Gaussians to monitor activity of a site over
extended period of time. This detects patterns of the motion and interaction demonstrated by objects in the site

1. Provide statistical description of typical activity pattern (normal behavior)

2. Detects unusual events, by spotting activity that is very different from normal activity pattern.

3. Should detect unusually interactions between objects.

\[ P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \ast \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) , \]

The results are good. Average over 3 scenes, 22 possible detections. Overall accuracy is 70% with false positive rate at 16% and false negative rate: 15%.

7.4 Remarks

It is interesting to note that the our algorithm does a lot better on UMN dataset than the CAVIAR dataset. The reason has to do with amount of movement in each dataset. The CAVIAR dataset are videos of one or two people, whereas the UMN dataset are videos involving a group of people. Group of people creates more movement (optical flow) than one or two people. Therefore the subspace angle response is stronger with movement by group of people than movement by two people. This may not directly have to do with the number of people in the video, but the amount of change proportional to the frame size.
We can tell the strength of the subspace response by looking at the y-axis. The UMN dataset max at 0.4 where as the CAVIAR dataset max at $3.5 \times 10^{-3}$ (which is still very close to zero). From this we can tell that the UMN dataset has stronger subspace angle response than the CAVIAR dataset.
Chapter 8

Future Work

8.1 Biomedical Applications

Behavior understanding and anomaly detection is not only useful for surveillance application, it is also very useful for biomedical applications such as cell to cell/substance interaction, mitosis, cancer metastasis and atherosclerosis. Most of anomaly detection in medical scenario are focused on studying organ that are more prone to certain type of cancers: breast, lungs and brain.

For example, I use my anomaly detection algorithm to do detect apoptosis (cell suicide). The extension here is that I was not only able to detect when cell suicide happens, I was also able to locate where the cell suicide occurs by using subimages.
This is extremely useful since localization is usually done by tracking, which is a very difficult problem.

The following shows the result of anomaly detection in apoptosis.

8.2 Improving Surveillance Systems
We can always improve the accuracy, speed, robustness of our algorithm. As mentioned above, it would be good to be able to localize the anomalies. Real-time application would be extremely useful in case of surveillance systems. The robustness of our algorithm can be improve with experimenting with more difficult datasets.
Bibliography


