Progress Report of Thesis

# Abstract

The thesis focuses on the research of abnormal activities detection in video by using a space-time Markov Random Field (MRF) model. To build the model, I divide each frame into x by y overlapping local regions and compute corresponding optical flow features. Furthermore, the optical flow is applied to a Mixture of Probabilistic Principal Component Analyzers to learn a generative model for local activity patterns (normal or abnormal). For every new frame, the space-time MRF will update by MAP computation. In addition, a method based on Gaussian Mixture Model (GMM) is achieved to detect movement fields. By combining the above two methods, I hope the research can detect the abnormal activities better. Of cause, this needs the further validation of experiments. By now, the optical flow computing has been achieved with a multi-scale block-based matching. Optical flows obtained at each scale are summed into a final flow vector, and generate a 9-dimensinal vector (8 orientations and 1 speed) for every pixel. Moreover, the movement fields detection based on GMM also finished, which has a robust performance. Next, I will concentrate on the achievement of the MPPCA and space-time MRF model, and add the movement fields’ detection to enhance the result of detecting abnormal activities.

# Introduction

Intelligent visual surveillance has got more research attention and funding due to increased global security concerns and an ever increasing need for effective monitoring of public places such as airports, railway stations, shopping malls, crowded sports arenas, military installations, etc., or for use in smart healthcare facilities such as daily activity monitoring and fall detection in old people’s homes. Often times, the objective is to detect, recognize, or learn interesting events which contextually may be defined as “abnormal behavior/activity”, “anomaly”, etc.

The three video surveillance research directions are detection and tracking, human motion analysis, and activity analysis (parsing temporal sequences of object observations to produce high-level descriptions of agent actions and multi agent interactions). And activity analysis will be the most important area of future research in video surveillance. This projection appears no less true today as research publications in this field over the last decade show. The use of closed-circuit television (CCTV) cameras to capture and monitor scenes by human agents has become ubiquitous. Although video footage capturing devices are more affordable and popular in today’s world, available human resources to monitor and analyze the footage are quite limited and sometimes not cheap. In many situations where surveillance cameras are used, it is common to find poor monitoring due to human factors like fatigue. The CCTV operators suffer boredom because in most cases, nothing “strange” or something that catches the attention occurs in the scene.

It is desirable to have systems that perform intelligent real time detection of “interesting behavior” to the human agent. The challenge, however, is that these events are rare and occur relatively infrequently (sometimes with very undesirable negative consequences). To aid human agents, efforts are being made to design intelligent surveillance systems that are capable of learning what normal behavior is and are able to distinguish between what is normal or abnormal within the context (because a normal behavior in one context may be abnormal in another).

# Finished Research Work and Result

## Optical Flow Features

Optical flow is the pattern of apparent [motion](http://en.wikipedia.org/wiki/Motion_(physics)) of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an [eye](http://en.wikipedia.org/wiki/Human_eye) or a [camera](http://en.wikipedia.org/wiki/Camera)) and the scene. The concept of optical flow was introduced by the American psychologist [James J. Gibson](http://en.wikipedia.org/wiki/James_J._Gibson) in the 1940s to describe the visual stimulus provided to animals moving through the world. James Gibson stressed the importance of optic flow for [affordance perception](http://en.wikipedia.org/wiki/Affordance), the ability to discern possibilities for action within the environment. Followers of Gibson and his [ecological approach to psychology](http://en.wikipedia.org/wiki/Ecological_Psychology) have further demonstrated the role of the optical flow stimulus for: the perception of movement by the observer in the world; perception of the shape, distance and movement of objects in the world; and the control of locomotion. This feature has since been co-opted by roboticists, who use optical flow techniques (including motion detection, object segmentation, time-to-contact information, focus of expansion calculations, luminance, motion compensated encoding, and stereo disparity measurement) for image processing and control of navigation.

The relationship between the optical flow in the image plane and the velocities of objects in the three dimensional world is not necessarily obvious. We perceive motion when a changing picture is projected onto a stationary screen, for example. Conversely, a moving object may give rise to a constant brightness pattern. Consider, for example, a uniform sphere which exhibits shading because its surface elements are oriented in many different directions. Yet, when it is rotated, the optical flow is zero at all points in the image, since the shading does not move with the surface. Also, specular reflections move with a velocity characteristic of the virtual image, not the surface in which light is reflected. For convenience, we tackle a particularly simple world where the apparent velocity of brightness patterns can be directly identified with the movement of surfaces in the scene.

Differential methods of estimating optical flow, based on partial derivatives of the image signal and/or the sought flow field and higher-order partial derivatives, such as:

[Lucas–Kanade[6] method](http://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method) regard image patches and an affine model for the flow field; [Horn–Schunck[3] method](http://en.wikipedia.org/wiki/Horn%E2%80%93Schunck_method) optimize a functional based on residuals from the brightness constancy constraint, and a particular regularization term expressing the expected smoothness of the flow field.

## MPPCA

Principal component analysis (PCA) (Jolliffe 1986) has proven to be an exceedingly popular technique for dimensionality reduction and is discussed at length in most texts on multivariate analysis. Its many application areas include data compression, image analysis, visualization, pattern recognition, regression and time series prediction.

The most common definition of PCA, due to Hotelling, is that, for a set of observed d-dimensional data vectors {tn},, the q principal axes w­j, , are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors w­j are given by the dominant eigenvectors of the sample covariance matrix such that and where the sample is mean. The vector, where, is thus a q-dimensional reduced representation of the observed vector tn.

Probabilistic principal component analysis (PPCA) [11] shows how the principal subspace of a set of data vectors can be obtained within a maximum-likelihood framework. Next we extend this result to mixture models, and outline an efficient EM algorithm for estimating all of the model parameters in a mixture of probabilistic principal component analysis (MPPCA) [12]. The partitioning of the data and the estimation of local principal axes are automatically linked.

The association of a probability model with PCA offers the tempting prospect of being able to model complex data structures with a combination of local PCA models through the mechanism of a mixture of probabilistic principal component analysis (Tipping and Bishop 1997). This formulation would permit all of the model parameters to be determined from maximum-likelihood, where both the appropriate partitioning of the data and the determination of the respective principal axes occur automatically as the likelihood is maximized.

## Learning Model

We use optical flow as a low-level measure of activity in local regions. We compute the flow with a multi-scale block-based matching between adjacent frames. Optical flows obtained at each scale are summed into a final flow vector, from which we compute a 9-dimensional optical flow vector (8 orientations + 1 speed) for every pixel. To construct a feature descriptor representing the atomic activity in each local region (corresponding to each node), we divide the region L into u by v sub-regions; each sub-region is represented by a 9-d vector obtained by summing the flow from all pixels within it. Finally, we concatenate the flow vectors of each sub-region into a 9uv dimensional activity descriptor for local region L. The number of sub-regions (i.e., u and v) is determined depending on how finely we want to capture the motion details.

After extracting descriptors for all local regions in the initial training video, we apply the Mixture of Probabilistic Principal Component Analyzers (MPPCA) algorithm to learn a generative model for local activity patterns. The dimensionality reduction offered by MPPCA gives us a compact representation of the high-dimensional descriptors. An MPPCA model is defined as follows:

(1)

where t is an activity descriptor, is a probability density function of mixture component i, and and denote the covariance matrix and mean vector of component i, respectively. The variable is a mixing coefficient for component i. Expectation-Maximization (EM) is used to compute all MPPCA parameters.

Rather than fit one model per local region, we construct a common MPPCA over all local regions. This is due to the fact that some local regions do not have enough samples in the initial video to allow stable convergence in EM; that is, most of the observations are motion-free at some local regions. Essentially, the mixture model probabilistically encodes the “vocabulary” of low-level motions. From the learned MPPCA, we compute two histograms: a frequency histogram at each node, and a co-occurrence histogram at each link. The frequency histogram represents how often each MPPCA component is observed at each node; the co-occurrence histogram records how often two MPPCA components co-occur at neighboring nodes. Together these empirical distributions describe the typical local activities and their interactions, and are used to establish the space-time MRF (to be defined in the following section).

Let Hi denote the frequency histogram at node i, and let Hi,j denote the co-occurrence histogram for neighboring nodes i and j, computed as follows:

(2)

(3)

where denotes the lth bin of Hi, and denotes the (l,m)th bin of Hi,j . The terms and are the posterior probabilities of the occurrence of MPPCA components l and m respectively, given activity descriptors ti,k and tj,k at nodes i and j at the kth frame. Thus, Hi(l) accumulates the posterior probability of component l over all previous activity descriptors observed at node i, thereby representing the likelihood of that low-level motion “type” occurring in that region of the video. Similarly, Hi,j(l,m) represents the likelihood that components l and m co-occur at neighbor nodes i and j, thereby capturing the common interactions between nearby regions, whether spatially or temporally. The posteriors are defined as follows:

(4)

(5)

## GMM for Movement Fields Detection

Rather than explicitly modeling the values of all the pixels as one particular type of distribution, we simply model the values of a particular pixel as a mixture of Gaussians [8]. Based on the persistence and the variance of each of the Gaussians of the mixture, we determine which Gaussians may correspond to background colors. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it. Our system adapts to deal robustly with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow-moving objects, and introducing or removing objects from the scene. Slowly moving objects take longer to be incorporated into the background, because their color has a larger variance than the background. Also, repetitive variations are learned, and a model for the background distribution is generally maintained even if it is temporarily replaced by another distribution which leads to faster recovery when objects are removed. Our background generation method contains two significant parameters - a, the learning constant and T, the proportion of the data that should be accounted for by the background. Without needing to alter parameters, our system has been used in an indoors, human-computer interface application and, for the past 16 months, has been continuously monitoring outdoor scenes.

In practice, the illumination in the scene could change gradually (daytime or weather conditions in an outdoor scene) or suddenly (switching light in an indoor scene). A new object could be brought into the scene or a present object removed from it. In order to adapt to changes we can update the training set by adding new samples and discarding the old ones. We choose a reasonable time period T and at time t we have. For each new sample we update the training data set XT and estimate again. However, among the samples from the recent history there could be some values that belong to the foreground objects and we should denote this estimate as. We use GMM with M components:

Where are the estimates of the means and are the estimates of the variances that describe the Gaussian components. The covariance matrices are assumed to be diagonal and the identity matrix I has proper dimensions. The mixing weights denoted by are non-negative and add up to one. Given a new data sample at time t.

Where . Instead of the time interval T that was mentioned above, here constant describes an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation having in mind that approximately. For a new sample the ownership is set to 1 for the 'close' component with largest and the others are set to zero. We dene that a sample is 'close' to a component if the Mahalanobis distance from the component is for example less than three standard deviations. The squared distance from the m-th component is calculated as: . If there are no 'close' components a new component is generated with where is some appropriate initial variance. If the maximum number of components is reached we discard the component with smallest.

The presented algorithm presents an on-line clustering algorithm. Usually, the intruding foreground objects will be represented by some additional clusters with small weights. Therefore, we can approximate the background model by the first B largest clusters:

If the components are sorted to have descending weights we have:

where cf is a measure of the maximum portion of the data that can belong to foreground objects without influencing the background model. For example, if a new object comes into a scene and remains static for some time it will probably generate an additional stabile cluster. Since the old background is occluded the weight B+1 of the new cluster will be constantly increasing. If the object remains static long enough, its weight becomes larger than cf and it can be considered to be part of the background. If we look at (4) we can conclude that the object should be static for approximately frames. For example for cf = 0:1 and = 0:001 we get 105 frames.

# Further Research Work and Shedule

## Space-Time MRF Model

Whenever a new video frame comes in, we construct a space-time MRF in an online manner using the new frame and a fixed-length history of recently seen frames (we use 10 in our experiments). The MRF is defined in terms of two functions: the node evidence and the pair-wise potentials. We compute them both in terms of the learned MPPCA model defined above. Ultimately, inference on the graph will yield the maximum a posteriori (MAP) labeling that specifies which nodes are normal or abnormal, as computed by maximizing the following:

(6)

where n(·) is the node evidence function, and p(·, ·) is a pair-wise potential function. The value λ is a constant to weight the node evidence, and xi denotes the label telling whether node i is normal or abnormal. (xi = 0 signifies node i is abnormal; xi = 1 means it is normal.) The node evidence function itself consists of two terms: a frequency term nf (·) and a suitability term ns(·). The frequency term measures how often an activity pattern (i.e., a MPPCA component) similar to the current activity descriptor at the given node has been observed before at that node. The suitability term evaluates how likely it is that the current activity descriptor was generated by the existing MPPCA model.

The frequency term imposes a relational constraint on each node-component pair. Simply speaking, if the activity descriptor detected at node i belongs to one of the frequently observed components for node i, the value of nf (xi = 1) becomes higher (or conversely, for a rarely observed component, it becomes lower). Complementarily, nf (xi = 0) = 1 − nf (xi = 1). We compute the frequency term from each node’s histogram Hi:

(7)

Where Hi(c) is a (normalized) frequency histogram for node i defined by Eq. (2)(3), and p(c|ti) is the posterior probability of component c given activity descriptor ti, as defined in Eq. (4)(5). The function Tk(·) is a transformation function to control the degree of sensitivity to abnormalities, and is defined as:

(8)

Lower values of the control parameter k will lead to fewer abnormal activity detections (i.e., less sensitivity to deviations from the model). This function is similar to those used for outlier rejection in robust statistics. In sum, nf (xi = 1) is the (transformed) normalized correlation between the frequency histogram Hi and the probability distribution of MPPCA components for a node i’s current activity descriptor.

The suitability term reflects how well the current MPPCA model explains the new activity descriptor t. We compute it as: , where the term p(t) denotes the p(t) given in Eq. (1). For numeric stability we directly use the Mahalanobis distance to evaluate it. Thus, the suitability is defined as follows:

(9)

Where is the Mahalanobis distance between activity descriptor ti and the MPPCA component c, normalized to be in [0, 1]. The normalization function, Fc(·) is the cumulative distribution of distances at the component c over all previous observations, which we implement using a cumulative histogram of the distances for all previous descriptors. Tk(·) is defined as above, and ns(xi = 1) = 1 − ns(xi = 0).

Finally, we have the complete node evidence function:

(10)

where is a weighting constant and is always set with > 0.5. Essentially, this serves to down-weight the frequency term should the activity descriptor at node i deviate significantly from the current MPPCA model (i.e., ns(xi = 0) > 0.5), which is important since the frequency term assumes that the observation can be explained well by the current model. Otherwise, we weight the frequency more than the suitability, as it is more discriminative in detecting abnormality as long as the activity descriptor’s Mahalanobis distance is low. In short, the node evidence function measures the normality of an activity descriptor at each node, and it balances the frequency and suitability terms depending on how well the descriptor can be explained by the existing MPPCA model.

The pair-wise potential function, \_(·, ·), consists of two terms: a co-occurrence frequency term \_f (·, ·) and a smoothness term \_s(·, ·). The co-occurrence frequency term evaluates how often we have observed twoMPPCA components co-occurring at neighboring nodes i and j. If xi = 1 and xj = 1, then

(11)

Otherwise, . This definition has a similar form to the frequency term of Eq. (7), except it uses the normalized co-occurrence histogram Hi,j defined in Eq. (2)(3). Here p(ci|ti) and p(cj |tj) denote the posterior probabilities of components ci and cj given activity descriptors ti and tj at nodes i and j, respectively. This term will measure how normal it is for two motions to co-occur at neighboring nodes.

The smoothness term imposes smoothness on label assignments between neighboring nodes based on their motion similarity: more similar motions lead to more smoothing, which is based on the fact that similar motions at neighboring nodes are more likely to be involved in a common activity, so they have higher probability of the same labeling being assigned. We compute this term based on the normalized correlation between the two activity descriptors:

(12)

## Incremental update of Space-Time MRF Model

Having built an MPPCA model using a small amount of initial training video, we can continuously update its parameters using the new activity descriptors extracted at every video frame. All the histograms (i.e., frequency histograms, co-occurrence histograms, and cumulative histogram of Mahalanobis distances) and MRF parameters are straightforward to adjust according to the updated MPPCA parameters.

To update the MPPCA parameters given a new activity descriptor, we first pick the most likely component for the descriptor, and then update the covariance matrix C and mean vector μ of that component cmax using the algorithm given in [7]. The mixing coefficients \_i for all components are also adjusted:

in which Nt and Nt+1 are the total numbers of activity descriptors observed until times t and t + 1, and Nt,i and Nt+1,i are the total numbers of activity descriptors belonging to the component i until times t and t + 1, respectively, and πt+1,i is the updated mixing coefficient of the component i at time t + 1.

Our incremental algorithm is quite simple and easy to implement. However, we should note one necessary approximation that it makes: we assume that the posterior probability of each component is unchanged once the descriptor is inserted into the model. Since the MPPCA parameters change whenever a new input comes in, the posterior probabilities of all previous descriptors should also change in response. However, re-calculating all the posteriors would mean touching every previous observation, thus defeating the purpose of an incremental update. (The method given in [7] incrementally adjusts a single component, without such a backward computation.) This is a well-known issue with incremental learning and mixture models. Following [4], we assume that the posterior probability of an activity descriptor is fixed to the value computed at the time when the descriptor was first introduced to the model.

To choose the number of MPPCA components automatically, we empirically identify the minimum number of components that appear to account for most of the initial dataset. Starting with a single component, we increase the number of components until it happens that some trivial component is formed that accounts for only a very small number of activity descriptors (e.g., less than 5% of overall training data). In future implementations, model selection techniques for EM (such as [2]) could be used.

## Schedule

April — May: achieve the whole algorithm of the model.

June — July: evaluate the model by experiments and improve.

August — September: write the thesis.

# Conclusion

The thesis uses a space-time MRF for detecting abnormal activities that combines the advantages of both local and global approaches. In addition, Gaussian Mixture Model is applied to detect the movement fields, which is robust in complex environment. In the future, the algorithm is going to concentrate on the combination of the above two methods.

# Reference

1. Chen J, Tang C K. Spatio-temporal markov random field for video denoising[C]//Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007: 1-8.
2. Figueiredo M A T, Jain A K. Unsupervised learning of finite mixture models[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002, 24(3): 381-396.
3. Horn B K P, Schunck B G. Determining optical flow[J]. Artificial intelligence, 1981, 17(1): 185-203.
4. Jepson A D, Fleet D J, El-Maraghi T F. Robust online appearance models for visual tracking[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2003, 25(10): 1296-1311.
5. Kim J, Grauman K. Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 2921-2928.
6. Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision[C]//Proceedings of the 7th international joint conference on Artificial intelligence. 1981.
7. Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2008, 77(1): 125-141.
8. Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking[C]//Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. IEEE, 1999, 2.
9. Shum H Y, Ikeuchi K, Reddy R. Principal component analysis with missing data and its application to polyhedral object modeling[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1995, 17(9): 854-867.
10. Sun D, Roth S, Lewis J, et al. Learning optical flow[J]. Computer Vision–ECCV 2008, 2008: 83-97.
11. Tipping M E, Bishop C M. Probabilistic principal component analysis[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 1999, 61(3): 611-622.
12. Tipping M E, Bishop C M. Mixtures of probabilistic principal component analyzers[J]. Neural computation, 1999, 11(2): 443-482.
13. Williams O, Isard M, MacCormick J. Estimating disparity and occlusions in stereo video sequences[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 2: 250-257.
14. Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction[C]//Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. IEEE, 2004, 2: 28-31.