PROCEEDINGS

The HKBU 5th Computer Science Postgraduate Research Symposium

January 11, 2007

PG Day 2007

Department of Computer Science
Hong Kong Baptist University

The 5th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

<table>
<tr><th colspan="3">January 11 Thursday, 2007</th></tr>
<tr><th>Time</th><th colspan="2">Sessions</th></tr>
<tr><td>09:30-09:50</td><td colspan="2">On-site registration (LMC514)</td></tr>
<tr><td>09:50-10:00</td><td colspan="2">Welcome: Prof. Yiu Wing Leung, Acting Head of Computer Science Department (LMC 514)</td></tr>
<tr><td>10:00-11:00</td><td colspan="2">Invited Talk: Prof. Anil K. Jain, Michigan State University (LMC 514)</td></tr>
<tr><td>11:00-11:10</td><td colspan="2">Tea Break</td></tr>
<tr><td rowspan="2">11:10-13:10</td><td colspan="2">Session A: (Chair: KaHo Chan) (LMC 514)
Pattern Recognition</td></tr>
<tr><td colspan="2">• Class-Distribution Preserving Transformation for Face Biometric Data Security
YiCheng Feng
• Human Motion Segmentation and Tracking using Boosted Attention-based Features for Surveillance Applications
Chang Liu
• Face Recognition Based on Wavelet Decomposition and LDA
LiMin Cui
• A Method of Handwritten Chinese Character Recognition
JianJia Pan</td></tr>
<tr><td>13:10-14:00</td><td colspan="2">Noon Break</td></tr>
<tr><td rowspan="2">14:00-15:30</td><td>Session B1: (Chair: Xiaofeng Zhang) (LMC 514)
Intelligent Informatics</td><td>Session C1: (Chair: ManChung Yeung) (LMC 511)
Networking</td></tr>
<tr><td>■ Learning Nominal Data Similarities for Kernel Methods
Victor Cheng
■ Orthogonal NMF based POMDP Compression
Xin Li
■ Maximum Entropy in Regularized Classifier Ensemble
ZhiLi Wu</td><td>■ Lightweight Piggybacking for Packet Loss Recovery in Internet Telephony
WingYan Chow
■ Performance Evaluation of IEEE 802.11 DCF with Internal UDP Traffic
Yong Yan
■ Design of Proportional Delay Guarantee Controller of a Cluster-Based Web Server
KaHo Chan</td></tr>
<tr><td>15:30-15:40</td><td colspan="2">Tea Break</td></tr>
<tr><td rowspan="2">15:40-17:40</td><td>Session B2: (Chair: Xin Li) (LMC 514)
Intelligent Informatics</td><td>Session C2: (Chair: Junyang Zhou) (LMC 511)
Networking</td></tr>
<tr><td>■ Agent based testbed for relax criteria negotiation
KaFung Ng
■ Creating Service Flows Using Semantic Approach
Kai Kin Chan
■ Data Hiding on 3D Geometry:A Perspective from ICA to Orthogonal Transformation
HaoTian Wu
■ An Maximum Weighted Likelihood Estimation for Unsupervised Model Selection and Feature Weighting on Gaussian Mixture
Hong Zeng</td><td>■ Privacy-Preserving Location-based Queries in Mobile Environments
Jing Du
■ Wireless LAN Positioning : Studies on Asymmetrical Signal Strength
Wilson M. Yeung</td></tr>
<tr><td>18:30</td><td colspan="2">Best Paper & Best Presentation Awards Announcement via Email</td></tr>
</table>

TABLE OF CONTENTS

Session A: Pattern Recognition

Session B1: Intelligent Informatics

Session C1: Networking

Session B2: Intelligent Informatics

Session C2: Networking

Class-Distribution Preserving Transformation for Face Biometric Data Security

Yicheng Feng

Abstract

This paper has proposed a new scheme to protect face biometric data against attack. After feature vectors are extracted from biometric data, a transformation scheme based on distance to "distinguish points" is proposed to transform the original feature vectors to binary strings with Hamming distance, which preserves the distribution of these feature vectors. After this they are encoded by the error-correcting coding process with BCH code, and then encrypted for protection. The binary transformation scheme makes the error-correcting coding approach feasible to the original face recognition scheme. The scheme is proved to be feasible and has good performance, which can enhance a security of about 78∼126 bits while the accuracy of the system is affected no more than 0.8%.

1. Introduction

In developing real world biometric applications, protection of biometric data (security) [1, 8, 7] is one of the main concern. While most of the current research works focus on the recognition/verification performance of biometric authentication under different conditions and in large databases, biometric security has received less attention. This paper focuses on the protection algorithm for face biometric.

The biometric cryptographic systems using public-key architecture are not secure enough because of open public key. An attacker may use this key to encrypt his own biometric data and build a fake token to access the system if it uses the public key for encryption. If the system uses the private key for encryption, open public key can be used for decryption thus the biometric data is totally insecure. Therefore, in matching process the biometric data should be compared in encrypted space to get higher security. However, matching in encrypted space is sensitive to image variations. Error correcting coding scheme is used to address this problem. The error correcting codes compensate the variation of biometric data before authentication, and the encryption functions can be applied. It has been applied to protect fingerprint and iris biometrics.[4, 3, 5, 6] Surprisingly, as far as we know, there are not many research articles on protecting face biometric. One related article is to protect face photo on ID cards. [9] Another approach, which computes cryptographic key from face images and thus enhances some security, gets a closest thinking to ours. [11] An cryptography-based scheme with Reed-Solomon codes [2] was proposed in 2006.

In our research we mainly concentrate on the security of face biometric data. We apply the error-correcting coding approach and choose the BCH codes because it has strong error-correcting ability and computational simple. However, those common error-correcting codes including BCH codes can only correct variation in vectors with Hamming distance while the variation in face feature vectors are different. Other error-correcting codes, such as the bounded distance coding algorithm applied in Davida's off-line scheme [4] are not suitable because face feature vectors can have large variation. A binary string transformation is done before coding to solve this problem. Because the transformed vectors become binary strings with Hamming distance, BCH codes are suitable.

This paper is organized as follows. Section 2 gives the description of our proposed scheme. Experimental results and the Security analysis will be given in Section 3 and Section 4, and at last is the conclusion.

2. Class-distribution preserving transformation

2.1. Basic idea

To solve the variation problem in the error-correcting biometric cryptosystem we propose a special binary string transformation: the Class-distribution preserving transformation. It is such a function that transforms the original face feature vectors to binary strings, making feature vectors from the same class transformed to similar binary strings and feature vectors from different classes transformed to binary strings much different from each other. The transformed binary strings are treated as the new representations of biometric data and used for authentication, which is shown in figure 1. The basic idea of this scheme is: the Euclidean distances from a random vector (or point) to feature vectors in the same class will be near the same, while distances to feature vectors in different classes may be much different. After quantization the distances can be

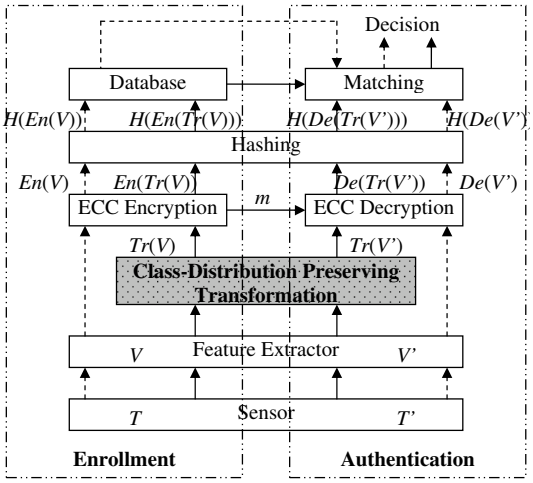


Figure 1. The error-correcting coding approach with class-distribution preserving transformation, in which "ECC" means "error-correcting codes". The dashed arrows show the flowchart of the authentication process without the transformation, while the real arrows show the flowchart with the transformation.

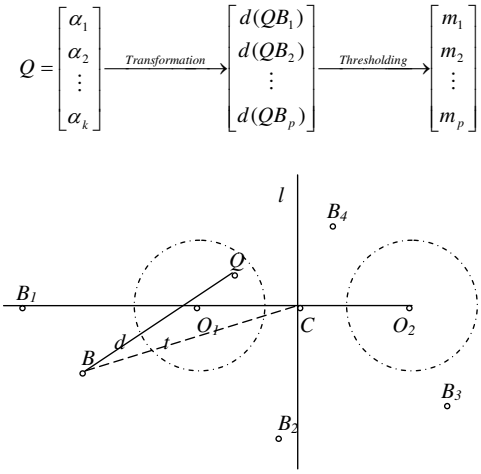


Figure 2. The simple two classes condition for the class-distribution preserving transformation.

transformed to 0 or 1, thus, for each random point, we get a distance to a certain feature vector and so as a bit. If there are many random points, we can get a binary string of that feature vector.

Consider the simplest condition, which has only two classes (class 1 & 2) of feature vectors with vector length 2. Treat these feature vectors as points in a plane. Assume the means of these two classes are separately O_1 and O_2, and a random point B lies beyond them. Point C is the center of line O_1O_2. P is some point in the two classes. The whole condition is shown in figure 2. Use the distance (denoted as d) from B to P and a threshold (denoted as t) to quantize point P to a bit:

$$m_B = \begin{cases} 0 & \text{if } d \le t \\ 1 & \text{if } d > t \end{cases}$$

Then we can classify the points by its transformed bit. If the bit is 0, it belongs to class 1, otherwise belongs to class 2. It is obvious that when B lies near the extension line of O_1O_2 (position of B_1) and t is set to $|BC|$, this bit-based classification will own an almost optimal performance. And if B lies near the midperpendicular l of line O_1O_2 (position of B_2), the performance will be very bad. We call the point B used for classification as "distinguish point" and the location that get good/bad performance as "good/bad position". We can even use more than one distinguish point. Just as figure 2 shows, distinguish points B_1, B_2, B_3 and B_4 are used and four bits are extracted, constructing a binary string $m_1m_2m_3m_4$.

Treat the original feature vectors with length k as points in k-dimensional space, we can still use the same way to classify these points. Distinguish points are generated and distances are computed. If more points are generated for classification, it is more possible that a pair of classes can find a distinguish point which lies on good position, thus the classifying ability will be better. For each feature vector v, compute its distance d to each generated distinguish point B_i. A threshold t_i is used to quantize the result to bit m_i. After the distances to all distinguish points are computed and quantized, we get a binary string $m_1m_2m_3m_4m_5 \ldots m_p$, in which p is the number of distinguish points. Two binary strings transformed from vectors of the same class should certainly be similar. For two feature vectors in different classes, it is much possible that the transformed binary strings are very different when p is very large, such as 200∼1000. As a result, the transformed binary strings can represent the distribution of the original feature vectors.

2.2. Variation considering

The variation problem is actually highly weakened after vector transformation because the original feature vectors are quantized to binary strings. After quantization little variation will be eliminated. But if feature vectors v_1 and v_2 belongs to the same class and their distances to a distinguish point B are separately larger and smaller than threshold t, v_1 and v_2 will be treated from different classes. The quantization can not solve this problem. To solve this problem, a variation range r is defined and the judge rule for transformation is modified:

$$m_i = \begin{cases} 0 & \text{if } d < t - r/2 \\ 1 & \text{if } d > t + r/2 \\ \phi & \text{if } t - r/2 \le d \le t + r/2 \end{cases}$$

The transformed vector is then not binary string, but a binary string with some bits of value ϕ. In comparison of two binary strings, if some bit of one string is ϕ, even the corresponding bit in the other string is not ϕ, this bit pair will not be counted in the Hamming distance. Two corresponding bits of v_1 and v_2 are counted in the Hamming distance if and only if they are separately "0" and "1", that is, the distance from v_1 to B is no more than $t - r/2$, and the distance from v_2 to B is larger than $t + r/2$, or opposite. These two distances have a difference at least r. If r is large enough, it can make sure that the two vectors belong to different classes.

2.3. Thresholds decision

The thresholds t_i of the scheme should be carefully chosen to enhance the performance of the authentication. We can use the same t for all distinguish points, or compute the average distance from the distinguish point to the feature vectors and set it as the threshold to that point. However, these two ways don't directly link the thresholds to system performance. They are just determined by personal will, but not by theoretic reason. In our scheme, we consider a special way making more sense to decide the thresholds, which do make sense and will get a better performance than the previous two ways.

Assume there are totally m classes with their average feature vectors $O_1, O_2, O_3 \ldots O_m$ (treated as points in feature space). In our scheme we generate mp random points $B_1, B_2,$
$B_3 \ldots B_{mp}$ as the distinguish points. The corresponding mp thresholds are determined by

$$t_i = |O_q B_i|, (q = int((i-1)/m) + 1.)$$

Thus, for each average feature vector O_q, its distances to p points $|O_q B_{(q-1)p+1}|$, $|O_q B_{(q-1)p+2}|$, $|O_q B_{(q-1)p+3}| \ldots$ $|O_q B_{qp}|$ are set as the corresponding thresholds. The condition that $p = m = 4$ is shown in figure 3:

After this setting, consider a feature vector (point) P in class 1. P belongs to class 1 means that P is close to O_1, thus,

$$|PB_i - O_1 B_i| < \varepsilon, i = 1, 2, p.$$

in which ε is a small scalar. If $r/2 > \varepsilon$, then $|PB_i - t_i| = |PB_i - O_1 B_i| < r/2$, $i = 1, 2, \ldots p$, thus, the first p bits of the transformed vector should be ϕ. Then all the feature vectors in class 1 will be transformed to binary strings with first p bits ϕ, thus, the same. As the same way, the $p+1_{st}$ to $2p_{st}$ bits of the binary strings transformed from class 2 will be the same as ϕ, the $2p+1_{st}$ to $3p_{st}$ bits of the binary strings transformed from class 3 will be the sameand so on. In other words, this thresholds setting method can make sure p bits in the transformed binary vectors from the same class to be the same, thus, decreases the FRR.

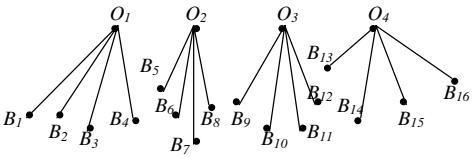


Figure 3. Thresholds specifying with average feature vectors.

3. EXPERIMENTAL RESULTS

In our experiments, we apply our threshold-specified protection scheme to fisherface [10] authentication systems to see its performance. The ORL database is used for testing, with 40 individuals and 10 images per person. The purpose of our experiments is to test how is the error rate of the authentication affected by the transformation process. In the experiment, length of the feature vector is 39, the parameter p is chosen 10.

The experiment results which test the transformation algorithm is shown in the following figures (figure 4 & 5). The pair of figures shows the error rates of the system without and with binary string transformation. The rates marked in the figures shows the cross-over error rates of separate test results. In figure 4, the cross-over error rate is 5.7%. In figure 5, the creoss-over error rates with different parameter r are almost the same as 6.5%. Thus the performance of the system is affected by the transformation algorithm about 0.8%.

4. Security analysis

4.1. Security enhanced system design

After transformation, the original face feature vectors are transformed into new binary strings. Between strings representing the same class there is variation of Hamming distance. Thus, we can apply error-correcting codes to the system. The BCH code is chosen and the encoding/decoding process from the fuzzy scheme [3] is shown as follows. In enrollment, assume the binary string that needs to be encoded is s. A BCH codeword c is randomly generated. Compute $s - c$ and hash the codeword c, then

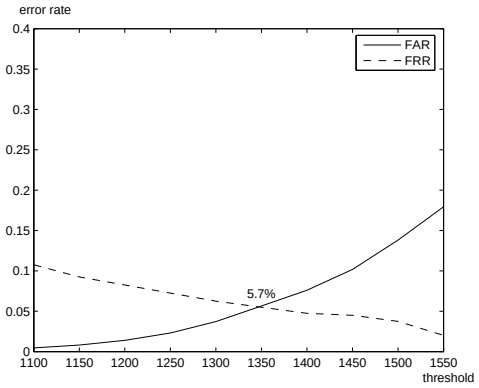


Figure 4. Experiment results with ORL database, LDA algorithm and without transformation.

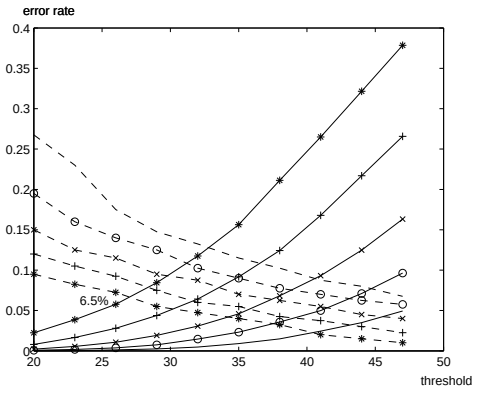


Figure 5. Results with ORL database, LDA algorithm and transformation. Lines with marks "-","o","x","+" and "*" separately represents parameter $r = 100, 120, 140, 160, 180$. The real lines represents the FAR, and the dashed lines represents the FRR.

store the hashing $Hash(c)$ and $s-c$ in database. In authentication, a new binary string s' is extracted from the presented biometric data. Take the data $(Hash(c), s-c)$ and compute $s'-(s-c)$. Do error correcting BCH decoding [13] to this $s'-(s-c)$ and we get s''. If s' and s has a Hamming distance no more than the threshold t, then $s'-(s-c) = c+(s'-s)$ has a Hamming distance no more than t with s. And the error-correcting decoding can correct $s'-(s-c)$ to s. That is s'' equals s. Compare $Hash(s'')$ and $Hash(s)$ we can know the decision of the whole authentication. The whole process is shown in figure 1. The stored $s-c$ is the message m shown in the figure which is generated in ECC encoding process and used for ECC decoding process.

4.2. Security level analysis

It is obvious that the security of our scheme depends on how many bits exist in the transformed bit strings (except the bits of value ϕ). If an attacker wants to access our system and he claims that he belongs to some class (suppose class 1), he should try to present a biometric data v belonging to the class and v will be transformed into a binary string b with some bits ϕ. If the binary string b has a Hamming distance no more than threshold t to the stored string s that represents class 1, b will be treated as class 1. Assume the length of the binary string is n, there are q bits with value "0" or "1" in the string b and pp bits of ϕ, the possibility that $Hm(b,s) \leq t$ is

$$Pr(Hm(b,s) \leq t) = (\sum_{x=0}^{t} C_q^x)/2^q.$$

The reciprocal is the security level of our system. From the equation, we know that the security level depends on q and t. Because different transformed binary strings may have different numbers of ϕ, thus different q because q equals to $n-pp$, we should analysis what's the distribution of pp's value after transformation. It depends on parameter r. With different r, the mean, maximum and minimum values of pp according to the binary strings are computed. Also, from figures 4 & 5 we can choose suitable threshold t to different r to get an error rate near the crossover error rate. Thus, with every r we get a t and pp, result in a security level, which is shown in figure 6.

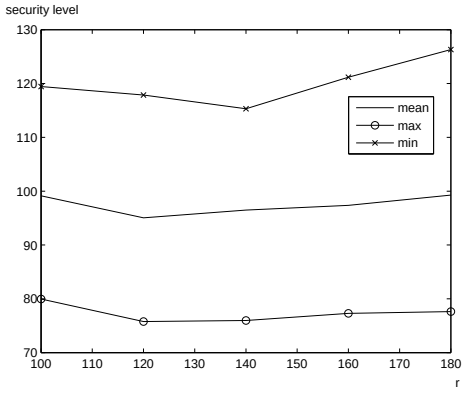


Figure 6. security level tested with ORL database and LDA algorithm. Line "max", "mean", "min" separately means the security levels computed from maximum, minimum and mean value of pp.

From the experiment results we know that the security level of our scheme highly depends on the string length, that is, how many distinguish points are used. This is depending on the feature vector length. Algorithm applied in ORL database with LDA algorithm can get a security level

of about 78∼126 bits.

5. Conclusion

This paper has proposed a cryptography based algorithm using BCH codes to protect face biometric data in authentication systems. The threshold-specified binary transformation scheme is proposed to solve the variation problem of face biometrics. Experiments have done and shows that this scheme is much feasible. It can highly enhance the security of the original system, and almost preserve the accuracy too with decreasing about 0.8%. Depending on how many distinguish points used, the system can get a security level of 78∼126 bits with ORL database. And the whole authentication can be done within about 0.1∼0.2s while implemented in matlab.

References

[1] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *End-to-End Security*, vol.40, 2001.

[2] Y. C. Feng and P. C. Yuen, "Protecting Face Biometric Data on Smartcard with Reed-Solomon Code," *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, p. 29, 2006.

[3] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," *Sixth ACM Conference on Computer and Communications Security*, pp. 28-36, ACM Press. 1999.

[4] G.I. Davida, Y. Frankel, and B.J. Matt, "On enabling secure applications through off-line biometric identification," *IEEE Symposium on Privacy and Security*, pp. 148-157, 1998.

[5] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *Proceedings of IEEE Internation Symposium on Information Theory*, p.408, 2002.

[6] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," *Proceedings of IEEE Internation Symposium on Information Theoryin Proc. ACM-SIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop*, pp.45-52, 2003.

[7] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *In Proc. Advances in Cryptology—Eurocrypt '04*, 2004.

[8] U. Uludag, S. Pankanti, S. Prabhakar and AK Jain, "Biometric Cryptosystems: Issues and. Challenges," *Proc. of the IEEE, Special Issue on Multimedia Security for Digital Rights*, vol. 92, no. 6, pp. 948-960, June 2004.

[9] D. Kirovski, N. Jojic, and G. Jancke, "Tamper-Resistant Biometric IDs," *in Information Security Solutions. Europe*, September 2004.

[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.," *IEEE Trans. on PAMI, 19(7)*, pp. 711-720, 1997.

[11] A. Goh and D. Ngo, "Computation of Cryptographic Keys from Face Biometrics," *Communications and Multimedia Security*, 2003.

[12] J.I. Hall, "Cyclic Codes," *Notes on Coding Theory*, 2003.

[13] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Trans. Inform. Theory,*, vol. 15, no. 1, pp. 122-127, January 1969.

Human Motion Segmentation and Tracking using Boosted Attention-based Features for Surveillance Applications

Chang Liu

Abstract

Human tracking and behavior recognition is the main task in a visual surveillance system. In order to get nice recognition results and give accurate alarm when abnormal activity is happening, an accurate tracking system is needed. However, when complex backgrounds with illumination variation or other disturbing factors are considered, the recent tracking systems works not very accurately. In this paper, we propose a multiple objects tracking system based on focus of attention model[1][2], adaptive background updating model and Adaboost algorithm[3], it can be proved from the experiment results that our system is accurate, real-time and robust to illumination variation and can achieve good tracking results in various backgrounds.

1. Introduction

Visual surveillance in dynamic scenes, especially for humans and vehicles, is currently one of the most active research topics in computer vision. A visual surveillance system attempts to detect and track certain objects from image sequences, and more generally to understand and describe object behaviors[4]. Visual surveillance of human activity usually requires that people should be tracked correctly, information about their behavior can be obtained from characteristics of their trajectories and the interactions between them. There are many sources of difficulties in performing an accurate tracking task. First, the persons to be tracked need to be detected, though this task can be solved by motion segmentation, it is often affected by varying illumination or occlusion. Moreover, we need to track the humans continuously across different frames with varying amounts of inter-object or scene occlusions. Four main categories into which most tracking algorithms fall are feature-based tracking, contour-based tracking, region-based tracking and model-based tracking[4], the main problem is how to increase the tracking accuracy in feature-based and region-based tracking, and how to reduce the computational complexity of contour-based and model-based tracking to make them work more efficiently.

We propose a human motion segmentation and tracking system using the feature-based method, in order to find more informative features to make this algorithm work more accurately, several attention based features are introduced: region contrast, object shape, object size, object location and motion information[2], and we combine these features in a adaboosted manner[3], which can select the best features with least errors in each round and combine them together into a strong classifier, this classifier can distinguish a positive human shape from negative background blobs. In addition, we propose a novel background updating method based on block changes, pixel difference and block variance difference are used to be the criteria that whether the block will be updated or not, the advantage of this approach is that it can balance the accuracy and computational complexity in pixel-based update method and object-based update method, with this method, illumination changes will cause background to update, so illumination variation will not be considered as motion information. We test the proposed tracking system with the well known video surveillance testing databases of CAVIAR[18] and PETS2001[19], satisfactory experimental results have been achieved.

The rest of the paper is organized as follows: Section 2 reviews the related work in human tracking; Section 3 gives a detailed description of our proposed system; Section 4 shows the experimental results; and Section 5 concludes this paper.

2. Related Work

Motion segmentation algorithms are normally based on background subtraction algorithms (BSAs)[6], some approaches combine this method with a temporal differencing approach[7]. These methods are based on extracting motion information by thresholding the differences between the current image and a reference image (background) or the previous image respectively. BSAs are widely used because they detect not only moving

objects but also stationary objects not belonging to the scene. BSAs are normally improved by means of updating their statistical description so as to deal with changing lighting conditions[8][9]. Haritaoglu[6] uses two different methods to update the background. First, a pixel-based update method changes the background model periodically to adapt to illumination changes in the background scene. Second, an object-based update method changes the background model to adapt to physical changes in the background scene. A deposited/removed object, or a parked car would be added into the background scene if it does not move for a long period of time.

Tracking algorithms establish a correspondence between the image structures of two consecutive frames, they usually have considerable intersection with motion segmentation during processing. Tracking methods are divided into four major categories[4]: feature-based tracking, contour-based tracking, region-based tracking and model-based tracking.

In the region-based category, modeling of the region's content by a histogram or by other non-parametric descriptions have become very popular in recent years. In particular, one of the most influential approaches is the mean-shift approach[11]. With the experience gained by using histograms and the mean shift approach, some difficulties have been studied in recent years. Some issues are the local basin of convergence that the mean shift algorithm has, the loss of spatial information and problems of occlusions. Another approach in Region-based tracking is to use part-based representation for human detection. Wu et al.[12] track the individual detected parts and then combine their responses in a combined tracker. The advantage of this approach comes from the observation that under partial occlusion conditions, some parts of the object remain visible and distinguishable and can provide reliable cues for tracking.

In the contour-based category, Jang et al.[10] propose an active template that characterizes regional and structural features of an object, which is built dynamically based on the information of shape, texture, color, and edge features of the region. By using motion estimation based on a Kalman filter, the tracking of a non rigid moving object is successfully performed by minimizing a feature energy function during the matching process.

Feature-based tracking algorithms perform recognition and tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images.[4] Polana et al.[13] provide a good example of global feature-based tracking. A person is bounded with a rectangular box whose centroid is selected as the feature for tracking. Even when occlusion happens between two persons during tracking, as long as the velocity of the centroids can be distinguished effectively, tracking is still successful. Viola et al[14] propose a pedestrian detection system that integrates image intensity information with motion information. They train a detector to take advantage of both motion and appearance information to detect a walking person by using adaboost. The implementation described runs at about 4 frames/second, detects pedestrians at very small scales and has a very low false positive rate.

Model-based tracking algorithms track objects by matching projected object models, produced with prior knowledge, to image data. The models are usually constructed off-line with manual measurement, CADtools or computer vision techniques.[4] Generally speaking, model-based human body tracking involves three main issues: construction of human body models; representation of prior knowledge of motion models and motion constraints; prediction and search strategies. As far as computational complexity is considered, search strategies are often carefully designed to reduce the solution space. As a recursive linear estimator, Kalman filtering can thoroughly deal with the tracking of shape and position over time in the relatively clutter-free case in which the density of the motion parameters can be modeled satisfactorily as Gaussian. To handle clutter that causes the probability density function for motion parameters to be multimodal and non-Gaussian, stochastic sampling strategies, such as Markov Chain Monte Carlo[15] and CONDENSATION (Particle Filter)[16] [17] are designed to represent simultaneous alternative hypotheses. Among the stochastic sampling strategies in visual tracking, particle filter is perhaps the most popular. This method is based on sampling the posterior distribution estimated in the previous frame and propagating these samples or particles to form the posterior for the current frame. However, it requires a relatively large number of samples to ensure a fair maximum likelihood estimate of the current state.

Generally speaking, feature-based and region-based tracking method have the problem of relatively lower accuracy, as illumination changes and occlusion is considered, while contour-based and model based tracking method have the problems of manual parameter initialization and computational complexity. As a feature-based tracking method, our proposed system use methods of focus of attention model, adaptive background updating model and Adaboost algorithm to build up a human tracking system which is real time and robust to illumination changes.

3. Human segmentation and tracking system

The overall system structure is as follows: Fig.1, it mainly consist of a block-based adaptive background updating model and an attention based adaboost model.

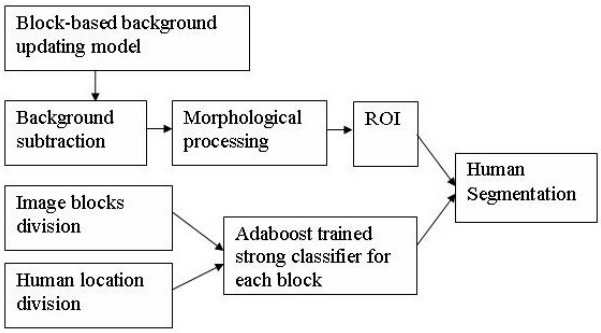


Figure 1. Flowchart of our human motion segmentation and tracking system.

3.1. Block-based adaptive background updating model

For a motion segmentation problem, background subtraction will be a very efficient method when the background is static and stable. In this method, moving regions are detected by taking the difference between the current image and the reference background image. However, this method is very sensitive to background changes when there are extraneous events in the video or are under unstable luminance conditions. Therefore, a good background model is needed to reduce the influence of these changes, in order that the background subtraction step will be more accurate and meaningful.

In this paper, we use a block-based background updating model for the motion segmentation step, which is illustrated in Table.1.

At the beginning, the first frame of the video is assumed to be the background and it is divided into several blocks, during the following frames, this background will be updated by exchanging each blocks of it. The assumption of this background updating method is that an image block will change if there is motion information variation in it. However, illumination changes will also make block change, so we also use block variance difference because illumination changes in blocks is a global transformation, and it will not change as fast as motion information. See Fig.2 for this.

```
Read Image(1) and divide it into m*n blocks
While not the end of the video
    Read image Image(i);
    For every block j
        If block pixel average difference>T1
        & block pixel variance difference<T2
            UpdateBlock(j);
        End
    End
    i=i+1;
End
```

Table 1. Block-based background updating model

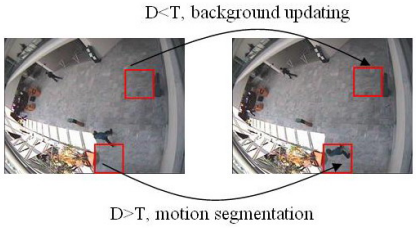


Figure 2. Illustration of the block-based background updating model, note that the block is updated when block difference D is lower than threshold T, while the block includes moving objects when D>T

3.2. Attention based human motion segmentation

Studies of visual attention and eye movement [5] have shown that humans generally only attend to a few areas in an image. Even when given unlimited viewing time, subjects will continue to focus on these few areas rather than scan the whole image. These areas are often highly correlated amongst different subjects, when viewed in the same context. Therefore, computers should pay attention to these factors which can provide more important information in a limited period of time.

A general observation is that objects which stand out from their surroundings are more likely to attract our attention, the factors which have been found to influence visual attention include: region contrast, object shape, object size, object location and motion information[2]. When object size information is considered, we notice that an object looks larger when it is near the camera, while the object looks relatively smaller when it is far away from the camera. Given sample videos and ground truth data, this object size information can be modeled with the assumption that object size is stable in each divided block in the image.

Another important attention based feature is object

location, if we concentrate on the object centroid and pay attention to the trajectories of the objects in several videos, we can find that the probabilities of the location where the object appears in the video are not the same, there always be some regions in the scene that object has more probability to appear or disappear, just like the entrance, exit, reception desk and so on.

Below is 3 ground truth images from CAVIAR database[18] and the probability map image trained from 14 video clips of CAVIAR (Totally 28 video clips), it can be seen that human movement trajectories are clustered at several most significant areas in the video.

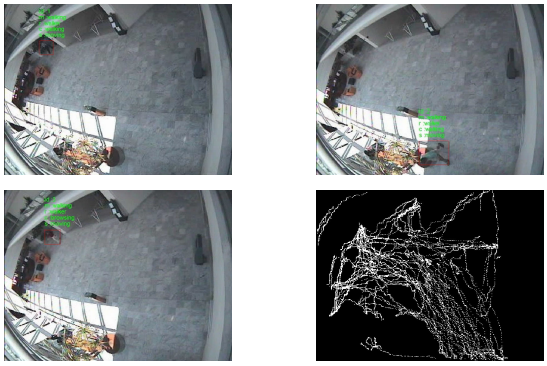


Figure 3. Three training images and the probability image

In the following step, we can combine the local features of region contrast and object shape, and use adaboost algorithm to select the most discriminative ones and segment human motion from videos.

3.3. Adaboost based on attentional local features

We integrate the background updating model and adaboost cascade into one framework. The flowchart of our approach is shown in Fig.4.

Adaboost[3] is a very efficient machine learning algorithm for feature selection, the main task of this algorithm is to construct several low computational weak classifiers based on their error rates, and combine these weak classifiers to a strong one(see four kinds of weak classifiers in Fig.5[3]),

The strong classifier will have the good property of both efficiency and accuracy, see Table.2 for more details of adaboost algorithm. For the real-time human tracking purpose, we use this algorithm to find the most discriminative features to represent the difference of a human shape and a non-human shape, and a region contrast difference between a human region(it means a small region which contain a whole human figure in it) and a non-human region.

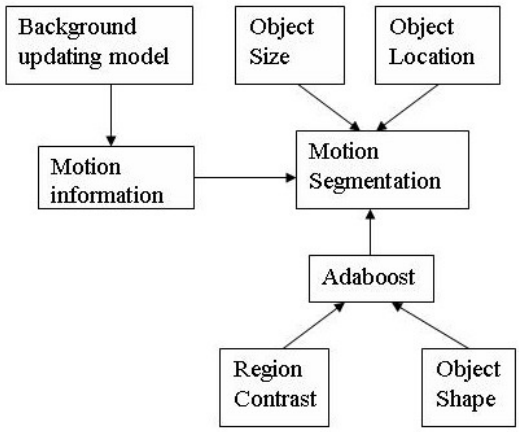


Figure 4. Flowchart of human motion segmentation using adaboost

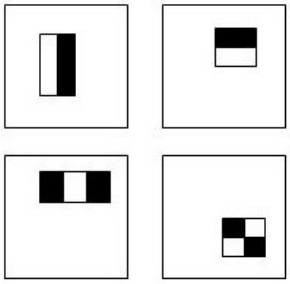


Figure 5. Four rectangle features that are used in this paper

1) Given $(x_1,y_1),(x_2,y_2)\ldots\ldots(x_m,y_m)$, $x_i \in \chi$, $y_i \in \{-1,+1\}$

Initialization: $D_1(i)=1/m$

2) For t=1,…,T

①Train weak classifier h_t according to distribution D_t

② Calculate the error rate of this weak classifier:

$\varepsilon_t = \sum D_t(i)[h_t(X_i) \neq Y_i]$

where $\alpha_t = 1/2 \ln((1-E_t)/E_t)$

③Update weights according to error rates

If $h_t(X_i) \neq Y_i$: $D_{t+1}(i)=D_t(i)*\exp(\alpha_t)$

If $h_t(X_i) = Y_i$: $D_{t+1}(i)=D_t(i)*\exp(-\alpha_t)$

Normalize D_t so that $\sum D_t(i)=1$

3) Output: $H(X)=\text{sign}(\sum \alpha_t h_t(X))$

Table 2. The adaboost algorithm[3]

In the initially offline training step, we use ground truth data which is represented by human object size and location information in the XML files, then humans are cropped from each frames to make up a positive dataset, and relative background images which have the same location and size are also cropped from the background updating model which has been stated before, these images build up the negative database, see the 1st column of fig.6 below. By using these two datasets, a strong classifier which can distinguish a human shape and a non human shape can be trained by adaboost.

Note that we also need to construct strong classifiers to distinguish human or non-human region contrast, these training set can be obtained by cropping the image which is double times of the width and height of the original image, because there are four diagonal directions around a shape sample, these region contrast database will be four times larger than de human shape database, see column 2 to 5 of fig.6 below, which represent the downright, downleft, upright and upleft region contrast images respectively.

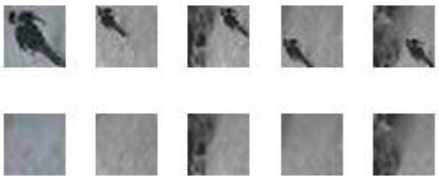


Figure 6. A few samples of training data, from left to right, they are respectively trained for shape feature, downright region contrast features, downleft region contrast features, upright region contrast features, upleft region contrast features

In the online detection step, strong classifiers trained by the above steps will be combined in a cascade manner, see fig.7, a positive sample should pass all the five cascades, it will be a negative sample if any strong classifier reject this sample.

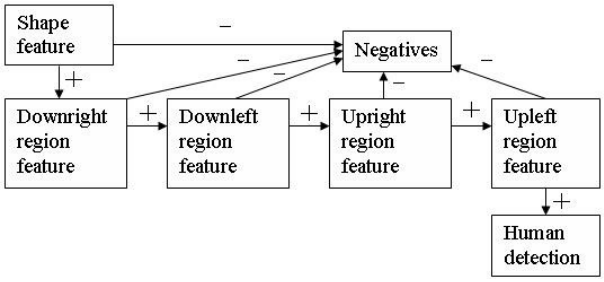


Figure 7. Cascade structure of the final classifier

Due to the original adaboost algorithm, each strong classifier has a fixed threshold $\frac{1}{2}\sum_{t=1}^{T}\alpha_t$, this will not be very effective because we always have to combine too many such strong classifiers to get a satisfactory results. In this paper, we use an adaptive thresholding method to make the strong classifier works more efficiently, $H^i(x)$ is given by

$$H^i(x) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T}\alpha_t^i h_t^i(x) \geq \theta^i \\ 0 \text{ otherwise} \end{cases}$$

Where $\chi^i = \arg\min_{\beta^i}(e_{FP}^i + e_{FN}^i)$ and $\beta^i = \sum_{t=1}^{T}\alpha_t^i h_t^i(x)$. θ^i is the threshold of the i^{th} strong classifier, which can be obtained by training data, e_{FP}^i and e_{FN}^i represent the false positive error and false negative error of the i^{th} strong classifier. Therefore, by using the adaptive thresholding method, thresholds which have the lowest overall error rate in the training dataset will be selected.

4. Experimental Results

We show results and evaluations on three video sets to demonstrate the effectiveness of our method. The first set is a selection from the CAVIAR video dataset[18], which is captured with a stationary camera, mounted a few meters above the ground and looking down towards a entrance lobby. There are totally 28 video clips which are classified by label:"Browsing", "Fighting", "Groups meeting", "Leaving bags", "Rest" and "Walking". The frame size is 384×288 and the sampling rate is 25 FPS. The second set of data is also a selection from the CAVIAR[18],it used a wide angle lens along and across the hallway in a shopping centre. For each sequence, there are two time synchronized videos, one with the view across and the other along the hallway. The frame size is 384×288 and the sampling rate is 25 FPS. These two data sets both have XML files that label the ground truth in the videos. The third set of data is a selection from PETS2001[19], which is well-known open database for video surveillance training and testing purpose. The frame size is 768×576 and the sampling rate is 25 FPS. The proposed real-time motion segmentation and tracking system was implemented on an 3G Hz compatible PC running Windows XP, the average speed to process a 384×288 image is about 0.15s and the average speed to process a 768×576 image is about 0.25s.

4.1. Training and Testing Results in CAVIAR Data Set

4.1.1. Training the Cascade. From the CAVIAR "entrance lobby" data set, we use 14 of the sequences (half of each label) and relative ground truth data to crop humans, and set up the training data set for each

block. In the experiment, we divide the whole image into 24×24 blocks, because the image size is 384×288, there will be 16×12=192 blocks totally. For each block, 5 strong classifiers should be trained: human shape classifier, downright region contrast classifier, downleft region contrast classifier, upright region contrast classifier and upleft region contrast classifier. See fig.8 for some positive training examples in block(5,5).

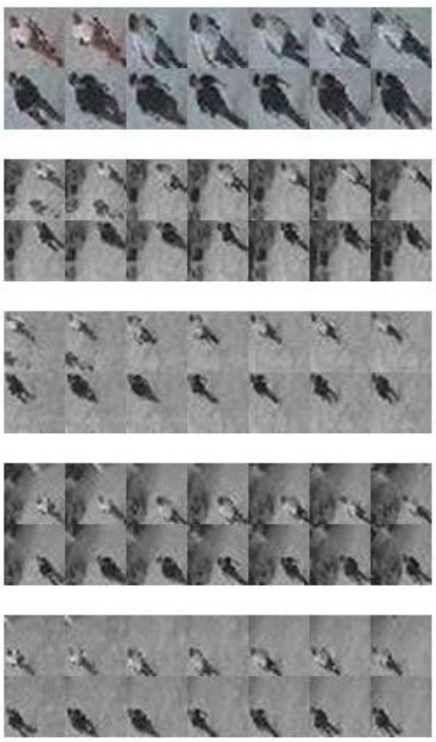


Figure 8. A small sample of positive training examples

Negative training samples are cropped from the background updating model, these samples have the same location and size as positive training samples, so their numbers are the same. The performance of five trained strong classifiers can be seen in Table.3. Then we combine these five strong classifiers as a cascade manner as is illustrated in Fig.7, and use the final strong classifier to segment human motion. The segmentation results will be shown in the next section.

4.1.2. Human Motion Segmentation Results. As is illustrated in Fig.1, from the background updating model, we use background subtraction and thresholding to get a binary image which represents the motion information, then morphological operations are used to find some larger motion segmentation regions. At last, by adding object location information, object size information and boosted object shape information and region contrast information, the final human motion segmentation areas can be decided. See the segmentation results in Fig.9, it is a test video in CAVIAR entrance lobby data set[18], notice that the downleft of each image is affected by unstable illumination, however, our system can track the person accurately, when the person is walking from an illuminated area to an unilluminated area.

	Shape	Downleft	Downright	Upleft	Upright
Total	14967	14914	14906	14884	14897
FP	228	221	226	216	218
FN	153	162	157	159	168

Table 3. Offline training errors comparative results, the first column is for shape feature, the other four columns are for respective region contrast features, the first row is total images, FP and FN stand for false positive and false negative respectively

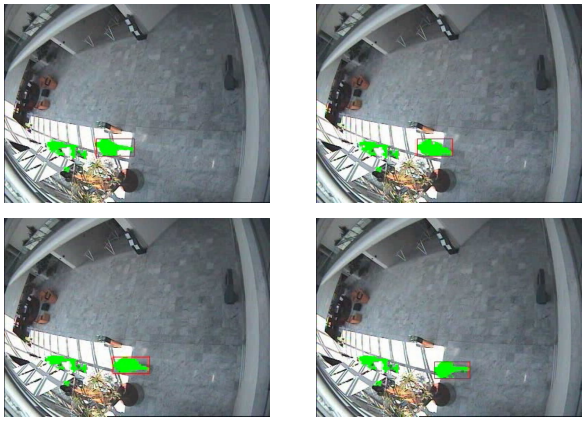


Figure 9. illumination, note that the left motion segmentation object(left green region) is rejected because it is not in region of interested(ROI)

Another experiment is on the CAVIAR shopping center data set[18], the same method is used and a part of the motion segmentation results are show in Fig.10. Note that there are more people appear in this data set and occlusion happens frequently, this will affect the segmentation result. This problem may be solved by kalman filter or particle filter by their continuous estimation property, which will be part of future work of this paper. Moreover, some people wear clothes that are very similar to the background, which will affect the background updating model and lead to miss segmentation. This problem may be solved by adding additional global features just like information theory, more research will be done on that topic later.

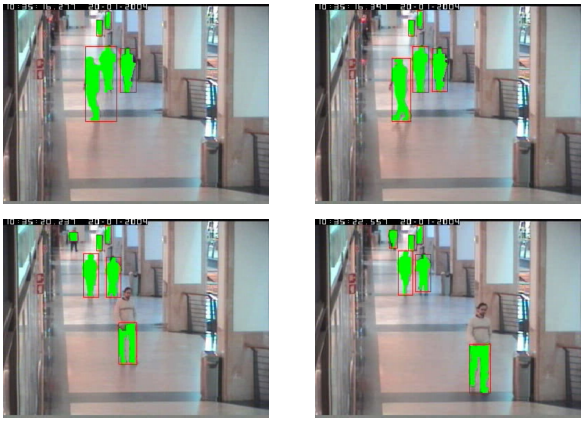


Figure 10. Testing results for CAVIAR shopping center, note that there are occlusion error and background subtraction error

4.2. Training and Testing Results in PETS2001 Data Set

Because there are a variety of moving objects in the PETS2001 database, object shape and size information will be no longer useful for the tracking system. Therefore, we only use this dataset to test the performance of background updating model, parts of the tracking results has been shown in Fig.11. Note that the fifth kinds of video clips don't show good segmentation result because these videos are recorded in a moving car, so the whole background is changing from time to time. In order to solve such kind of problems about moving camera object tracking, more robust algorithm should be proposed.

5. Conclusions

This paper proposes a human motion segmentation and tracking system, which uses methods of focus of attention model[1] [2], adaptive background updating model and Adaboost algorithm[3], experiment results have shown that this system has real-time property and can segment human motion from various kinds of backgrounds with satisfied results. Our future work will be concentrated on adding global features based on information theory to the system, and improve the robust property of the current tracking system. Moreover, we are considering using model based tracking method such as kalman and particle filter, it may give better results by only use feature based method in our tracking system.

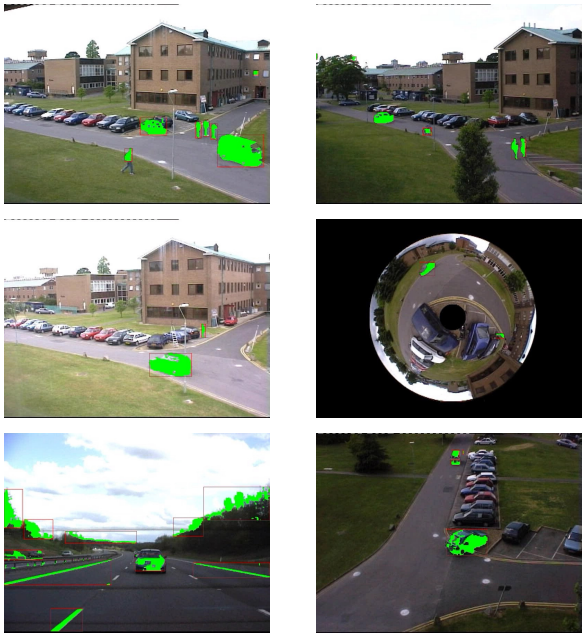


Figure 11. Parts of tracking results in PETS2001

References

[1] Itti Laurent, Koch Christof, Niebur Ernst. "A model of saliency-based visual attention for rapid scene analysis" [J]. IEEE Transactions on PAMI, 1998, 20(11): 1254-1259.

[2] W. Osberger and A. J. Maeder, "Automatic identification of perceptually important regions in an image. using a model of the human visual system" in Int. Conf. Patt. Recogn., pp. 701C704, Aug 1998.

[3] P. Viola. "Rapid object detection using a Boosted cascade of simple features". In: Proc IEEE Conference on Computer Vision and Pattern Recognition, pp:511-518, 2001.

[4] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank, "A survey on visual surveillance of object motion and behaviors", IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol. 34, No. 3, 2004, pp. 334-352.

[5] J.Senders. "Distribution of attention in static and dynamic scenes". In Proceedings SPIE 3016, pages186C194, SanJose, Feb1997.

[6] I. Haritaoglu, D. Harwood and L.S. Davis,"W4: Real-Time Surveillance of People and Their Activities", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), 2000, pp. 809-822.

[7] S. Huwer and H. Niemann,"Adaptive Change Detection for Real-time Surveillance Applications", Proceedings of The IEEE Workshop on Visual Surveillance, Dublin, 2000, pp. 37-43.

[8] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, "Tracking Groups of People", Computer Vi-

sion and Image Understanding 80, 2000, pp. 42-56.

[9] C. R. Wren, A. Azarbayejani, T. Darrel and P. Pentland, "Pfinder: Real-Time Tracking of the Human Body", Trans. Pattern Analysis and Machine Intelligence, 17(6), 1997, pp. 780-785.

[10] D.-S. Jang and H.-I. Choi, "Active models for tracking moving objects," Pattern Recognition, vol. 33, no. 7, pp. 1135C1146, 2000.

[11] Comaniciu,D.,Ramesh,V.,Meer,P.,"Real-time tracking of non-rigid objects using mean shift",In Proc.IEEE Conf.on Computer Vision and Pattern Recognition,2000,2:142-149.

[12] Bo Wu, Nevatia, R. "Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection" Computer Vision and Pattern Recognition, 2006 Volume: 1, On page(s): 951- 958

[13] R. Polana and R. Nelson, "Low level recognition of human motion," in Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, Austin, TX, 1994, pp. 77C82.

[14] Viola, P.; Jones, M.J.; Snow, D., "Detecting Pedestrians Using Patterns of Motion and Appearance", IEEE International Conference on Computer Vision (ICCV), Vol. 2, pp. 734-741, October 2003

[15] J. E. Bennett, A. Racine-Poon, and J. C. Wakefield, "MCMC for nonlinear hierarchical models," in Markov Chain Monte Carlo in Practice, W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds. London, U.K.: Chapman and Hall, 1996, pp. 339C357.

[16] M. Isard and A. Blake, "CONDENSATIONłConditional density propagation for visual tracking," Int. J. Comput. Vis., vol. 29, no. 1, pp. 5C28,1998.

[17] M. S. Arulampalam, S. Maskell, N. Gordon et al. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking" [J] IEEE Trans on Signal Processing. 2002, 50(2):174-188

[18] EC funded CAVIAR project under the IST Fifth Framework Programme (IST-2001-37540). Found at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[19] Found at http://ftp.pets.rdg.ac.uk/PETS2001/.

Face Recognition Based on Wavelet Decomposition and LDA

Limin Cui

Abstract

Face detection and recognition is one of the most notable branches of biometrics and it is also the one of the most active and challenging tasks for computer vision and pattern recognition. This paper presents a method for face recognition based on wavelet decomposition and LDA.

1. Introduction

Biometrics is automated methods using individual physiological or behavioral characteristics to verify identity. It provides a highly reliable approach to the identity recognition. Research in the face detection and face recognition made a great progress in recent 30 years. Automated face detection and recognition is one of the most important problems for computer vision and pattern recognition [1-3]. It is not only widely applied in a variety of personal identification systems such as national security, public security, justice, government, finance, business and security facilities, but also can be used in the fields of human-computer interface and visual communication.

The frame of face recognition system is shown in Fig. 1. First, face images are captured by camera. Because video is a rapid sequence of individual still images, it can also be used as a source of facial images. Using a pre-stored image database, the face recognition system should be able to identify or verify one or more person in the scene. Then preprocessing is to improve image qualities. Before face recognition is performed, the system should determine whether or not there is a face in a given image or a sequence of images. This process is called face detection. We can localize the face using face detection and set facial images to predefined size. Next, feature extraction can identify valid features to reduce dimension of pattern space. Classifier design can make classification of decision-making according to features. At last, we can identify the faces by features and classifier.

Different approaches of face recognition for still images can be classified into three groups as following [4]:

- **Feature-based Approach**

In feature-based approach, local features such as nose, eyes, mouth are extracted and segmented, and then these features are used as input data for classifier. Geometry, Hidden Markov Model methods belong to this group method.

Many earlier face recognition methods detect a set of geometrical features on the face such as the eyes, brows, nose and mouth [5, 6]. Properties and relations such as areas, distances, and angles between the features in the face are used as descriptors for face recognition. Hidden Markov Models are also important in earlier research work [7, 8].

- **Appearance-based Approach**

Appearance-based approaches use the whole face region as the raw input for a recognition system. An image is considered as a high-dimensional vector, namely, all points are in a high-dimensional vector space. Many appearance-based approaches use statistical methods to analyze the distribution of the object image vectors in the vector space, and derive effective representation (feature space) according to different applications. Given a test image, the similarity between the stored prototypes and the test view is carried out in the feature space.

Principle Components Analysis (PCA) has been widely applied to capture the face space in a low dimensional space composed of orthogonal eigenvectors: An eigenface is computed by estimating the eigenvectors of the covariance matrix of a training set. The eigenvectors corresponding to the largest eigenvalues are taken as the principle components of an eigenface and capture the main modes of variations in the training data set. PCA can be used as a low-dimensional linear representation for face detection and face recognition [9].

In order to have better identification, Linear Discriminant Analysis (LDA) can be applied to detect image variation due to external sources such as illumination and expression. Given a set of labeled face image, LDA can reduce dimension and maximize the separability of different faces [10, 11]. LDA involves the eigenanalysis of a product of two matrices, where one of these matrices has to be inverted. The obtained eigenvectors, used as LDA representations bases, are called 'fisherfaces'. In contrast with PCA, LDA is a supervised learning technique that relies on class labels, whereas PCA is an unsupervised technique. One characteristic of both PCA and LDA is that they produce

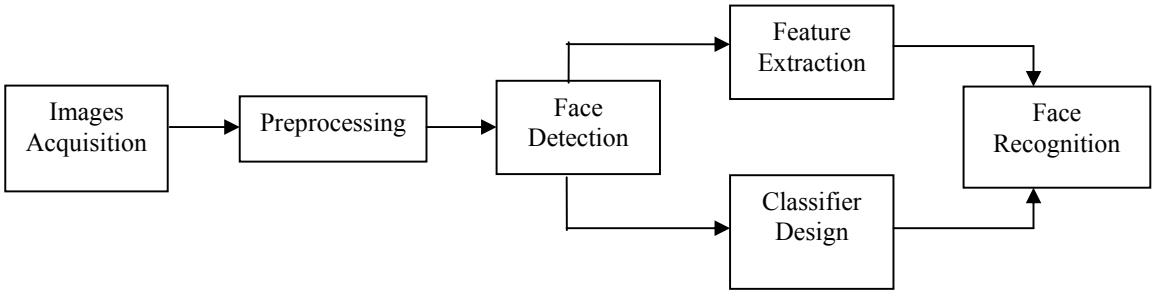


Fig. 1. The frame of face recognition system

spatially global eigenvector. In other words, the basis vectors produced by PCA and LDA are non-zero for almost all dimensions, implying that a change to a single input pixel will alter every dimension of its subspace projection. LDA is an effective face recognition method. By taking a transformation, we not only reduced dimension of feature space, but also make between-class scatter maximum and within-class scatter minimum. Character of classification can be improved.

- **Mixed Approach**

Mixed approach is combination of feature-based and appearance-based methods many obtain the best results. Cootes et al. present active shape models [12], and Chen et al present a method using a fast classifier to locate feature points candidates with a probabilistic output [13].

This paper mainly studies the approaches to frontal face detection and recognition in gray images. We proposed an approach of face recognition based on wavelet and LDA. The low frequency sub-images are obtained by 2-D wavelet transform for several times. The features are extracted by applying LDA to the sub-images. The nearest-neighbor classifier is designed to recognize the face. The results of experiment show that this method has good performance, and computation is reduced greatly.

2. Wavelet Decomposition

In recent years, wavelet transformation is being increasingly used in signal processing, image processing, computer vision and pattern recognition etc. The major characteristic of wavelet transform is, by means of continuous change of scale from wide to narrow, wavelet transform can gradually focus on any detail of analyzed object. As 'mathematical microscope', wavelet analysis becomes a new tool to simulate visual perception of multi-scale.

Nastar et al. studied the relations between face and its spectrum [14]. He found that facial expression and few occlusions only affect intensity manifold locally. If we represent by a frequency, it will only affect high frequency part. It is called high frequency phenomenon. We will have better effect if we represent human face by low frequency images and filter high frequency using wavelet.

Wavelet transformation is a method of representing signals across space and frequency. The signal is divided across several layers of division in space and frequency and then analyzed. The goal is to determine which space/frequency bands contain the most information about an image's unique features, both the parts that define an image as a particular type (fingerprint, face, etc.) and those parts which aid in classification between different images of the same type.

A 2-D wavelet transform is derived from two 1-D wavelet transform. 2-D wavelet decomposition is carried out by applying a 1-D transform to the row of the original image data and the columns of the row-transformed data respectively as shown in Fig. 2 (i). The image can be decomposed into four sub-images as shown in Fig. 2(ii). This decomposition can be repeated for n-levels as shown in Fig. 3. The image can later be reconstructed from these subspaces. Be useful if we are storing a large number of similar images. Fig. 4. is 2-D wavelet decomposition of a face image. First, we decompose the original face images (as shown in Fig. 4. (i)) into four sub-images via one-level wavelet decomposition as shown in Fig. 4. (ii). Then we decompose sub-image LL again, and we obtained 2-level decomposition image of original image as shown in Fig. 4. (iii). Following the process, we can make multiplayer wavelet decomposition. Sub-image LL denotes low frequency component of original face image and it is smoothing similar image of original image. In sub-image LH, sharp changes in the horizontal direction, i.e., vertical edges. However, in the sub-image HL, sharp changes in the vertical direction, i.e., horizontal edges. In the sub-image HH, sharp changes in non-horizontal, non-vertical directions, i.e., inclined edges. The sub-images LL by suitable wavelet transformation have the property of

invariant and good stability. After 2-D wavelet decomposition at appropriate level n, the size of low frequency image is only $1/2^{2n}$ of original image. The dimension has been effectively reduced.

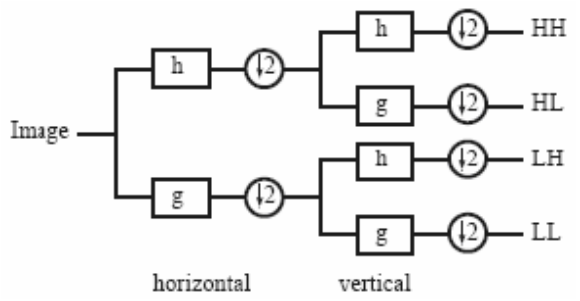


Fig. 2. (i) 2-D wavelet decomposition

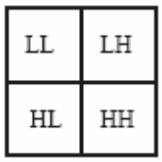


Fig. 2. (ii) Four sub-images after one-level wavelet decomposition

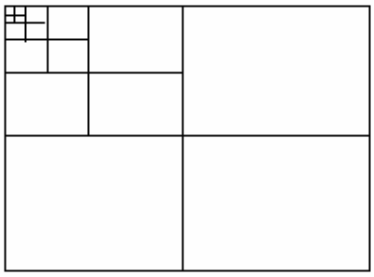
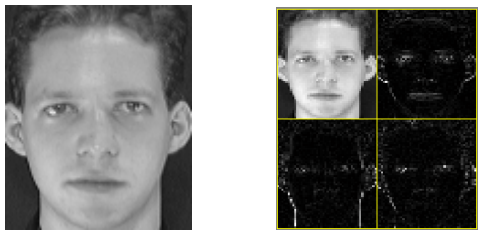


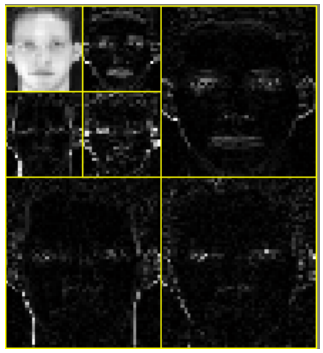
Fig. 3. n-level wavelet decomposition

3. LDA

There are many possible techniques for classification of data. Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two commonly used techniques for data classification and dimensionality reduction. Linear Discriminant Analysis (also called Fisher Linear Discriminant, FLD) easily handles the case where the within-class frequencies are unequal and their performances have been examined on randomly generated test data. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. LDA doesn't change the location but only tries to provide more class separability and draw a decision region between the given classes. This method also helps to better understand the distribution of the feature data.



(i) Face image (ii) Wavelet decomposition at level 1



(iii) Wavelet decomposition at level 2

Fig. 4. Wavelet decomposition of a face image

Linear Discriminant Analysis (LDA) [15] is a class specific method in the sense that it can represent data in form which is more useful for classification. Given a set of images $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, assume each image belongs to one of the c classes $\{X_1, X_2, \cdots, X_c\}$, and LDA selects a linear transformation matrix $\mathbf{W}$ in such a way that the ratio of the between-class scatter and the within-class scatter is maximized.

Mathematically, the between-class scatter matrix and the within-class scatter matrix are defined by

$$\mathbf{S}_B = \sum_{i=1}^{c} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

and

$$\mathbf{S}_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T$$

respectively, where $\boldsymbol{\mu}_i$ denotes the mean image of class X_i and N_i denotes the number of images in class X_i. If S_W is nonsingular, LDA will find an orthonormal matrix $\mathbf{W}_{opt}$ maximizing the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix. That is, the LDA projection matrix is represented by

$$\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \frac{\left|\mathbf{W}^T S_B \mathbf{W}\right|}{\left|\mathbf{W}^T S_W \mathbf{W}\right|} = [\mathbf{w}_1 \quad \mathbf{w}_2 \cdots \mathbf{w}_m]$$

The set of the solution $\{\mathbf{w} \mid i = 1,2,\cdots,m\}$ is that of the generalized eigenvectors of $\mathbf{S}_B$ and $\mathbf{S}_W$ corresponding to the m largest eigenvalues $\{\lambda \mid i = 1,2,\cdots,m\}$, i.e.,

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i, \quad i = 1,2,\cdots,m$$

4. The Frame of Face Recognition System

The proposed face recognition system consists of two processes, training and recognition processes. Fig,.5. is the fame of face recognition system based on wavelet decomposition and LDA.

- **Training Process**

For images training set, we adopt a kind of smoothing, compactly-supported orthonormal wavelets function for 2-D wavelet decomposition. After appropriate wavelet decomposition, we obtain the sub-image LL which is low dimension image. And we can obtain the wavelet eigenvector. Next, we compute $\mathbf{S}_W$ and $\mathbf{S}_B$, then we can get eigenvector corresponding maximum eigenvalue of $\mathbf{S}_W^{-1}\mathbf{S}_B$. So we can obtain feature of classification using LDA projection. And we adopt the nearest-neighbor classifier to create library of face recognition.

- **Recognition Process**

For an unknown image, we find wavelet eigenvector from its sub-image LL after appropriate wavlet decomposition. Then we can obtain feature of classification using LDA projection. According to the Nearest Neighbor Classifier, we can recognize the unknown face image.

5 Experiment Results

We adopt Daubechies wavelet because we not only need smooth, compactly-supported orthonormal wavelets, but also as few non-zero expansion coefficients as possible. Through numbers of experiments, we found db4 has the highest recognition rate.

We adopt db4 wavelet to decompose images at level 2 as following. The sub-image LL is a coarser approximation to the original image. HL and LH can show horizontal and vertical change of the image. HH is the high frequency component of the image. So we decompose the sub-image LL as the same method.

The experiment is performed on the Cambridge ORL face database, which contains 40 distinct persons as shown in Fig. 6. Each person has ten different images. There are variations in facial expressions such as open or closed eyes, smiling or nonsmiling, and glasses or no glasses. All the images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some side movements. There is also some variations in scale. We show four individuals as shown in Fig. 7.

In our face recognition experiments on the ORL database, we select 200 samples (5 for each individual) randomly as the training set. The remaining 200 samples are used as the test set. Using our method, we can obtain high recognition rate as shown in Fig. 8.

6 Conclusions

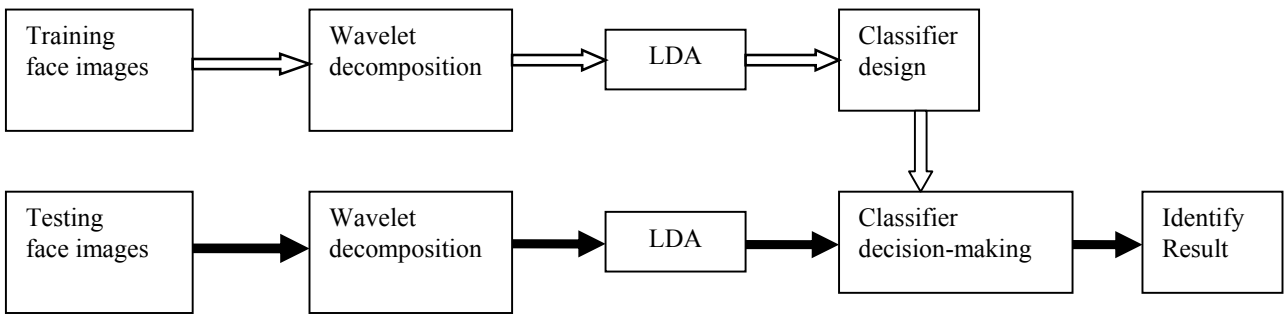


Fig. 5. The frame of face recognition system based on wavelet decomposition and LDA

This paper present a face recognition system based on wavelet decomposition and LDA. Different from traditional LDA face recognition, we took full advantage of the time-frequency localization properties of wavelet transform to reduce the computational load and gain good recognition rate.

Fig. 6. 40 distinct persons in ORL face database

Fig. 7. Four individuals in the ORL face database. There are 10 images for each person.

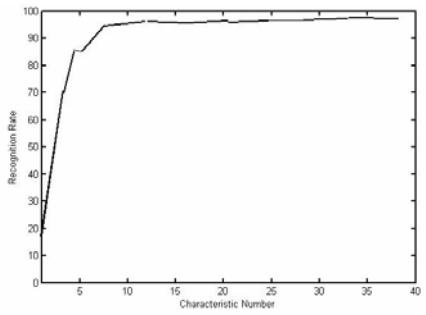


Fig. 8. Recognition Rate of face recognition based on wavelet decomposition and LDA

References

[1] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, 1995.

[2] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, Face Recognition: From Theory to Applications, *Springer-Verlag*, 1996.

[3] S. Gong, S.J. McKenna, and A. Psarrou, Dynamic Vision: from Images to Face Recognition, *Imperial College Press and World Scientific Publishing*, 2000.

[4] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, Face Recognition: A literature survey, *ACM Computing Surveys,* vol. 35, pp. 399-458, 2003.

[5] I. Cox, J. Ghosn, P.Yianilos, Feature-based face recognition using mixture-distance. *In: Proc. Int. Conf. Comput. Vis. Patt. Recogn*, pp. 209-216, 1996.

[6] B.S.Manjunath, R. Chellappa, and C.v.d. Malsburg, A feature based approach to face recognition, *Proc. IEEE Conf. Comput. Vis Patt. Recogn*, pp. 373-378, 1992.

[7] A. Nefian, M. Hayes, Hidden markov models for face recognition, *In: Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2771-2774, 1998.

[8] V. Kohir, E. Desai, Face recognition, *In: Proc. Int'l Conf. Image Processing*, pp. 309-312, 2000.

[9] M. Turk, A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience,* vol. 3, pp. 71-86, 1999.

[10] D. Swets, J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Transactions on Pattern Recognition Analysis and Machine Intelligence*, vol. 18, pp. 831-836, 1996.

[11] P. Bellhumeour, J. Hespanha,. D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Transactions on Pattern Recognition Analysis and Machine Intelligence*, vol. 19, pp. 711-720, 1997.

[12] T., Cootes, C. Tayor, D. Cooper, J. Graham, Active shape models-their training and application, *Computer Vision and Image Understanding*, vol. 61, 38-59, 1995.

[13] L., Chen, L. Zhang, H. M. Zhang, Abdel-Mottaleb, 3D shape constraint for facial feature localization using probabilistic-like output. *In: FGR04,* 2004.

[14] C. Naster, N. Ayache, Frequency-based non-rigid motion analysis. *IEEE Trans. PAMI*, 18(11):1067-1079, 1996.

[15] M.Turk and A.Pentland. Face recognition using eigenfaces. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

A Method of Handwritten Chinese Character Recognition

Jianjia Pan

Abstract

In recent years, the thorniest question that the off-line handwritten of Chinese character distinguished in pattern recognition region, has obtained the many research results. But, the handwriting Chinese character recognition was still considered as one of most difficult questions of writing recognition domain. This paper presents an application of SVM in small-set off-line handwritten Chinese characters recognition based on energy features of the wavelet sub-images. In this paper, we use wavelet energy as the feature. The Chinese characters are decomposed into several sub-images by wavelet transform, and energy features are extracted from these sub-images. Then software LibSVM is proposed for handwritten Chinese characters training. The experiments show that these wavelet-based features can effetely classify the training samples, and the SVM method can improve recognition rate.

Keywords: handwriting recognition, wavelet energy

1. Introduction

Pattern recognition of handwritten words is a difficult problem, not only because of the great amount of variations involved in the shape of characters, but also because of the overlapping and the interconnection of the neighboring characters. Furthermore, when observed in isolation, characters are often ambiguous and require context to minimize the classification errors. The existing development efforts have involved long evolutions of differing classification algorithms, usually resulting in a final design that is an engineering combination of many techniques.

Character Recognition plays an important role in many images processing tasks, ranging from remote sensing to medical imaging, robot vision and query by content in large image databases. Various methods for character feature extraction have been proposed during the last decades, but the character recognition problem remains difficult and subject to intensive research.

A major class of feature extractors relies on the assumption that character can be defined by the local statistical properties of pixel gray levels.

Several multi-channel character analysis systems have been developed [1] [2]. In particular, Gabor filters were employed to perform character segmentation [4] [5]. In the last decade, wavelet theory has emerged and became a mathematical framework, which provides a more formal, solid and unified framework for multi-scale image analysis [6] [7]. Typically, the wavelet transform maps an image on a low resolution image and a series of detail images. The low resolution image is obtained by iteratively blurring the image; the detail images contain the information lost during this operation. The energy or mean deviation of the detail images are the most commonly used features for character classification and segmentation problems [8] [9].

As a new machine learning method, Support Vector Machines is an effective method for pattern recognition. It has initially displayed performance better than the methods before, in solving small sample learning question, non-linear and high dimensional pattern recognition, SVM displays many unique superiorities[12]. Its basic thought may summarize as: first transform the input space to a high dimensional space through the nonlinear transformation. Then get the most superior linear classification surface in this new space, and the realization of this kind of nonlinear transformation is through the definition of suitable inner product function.

In this paper, we will present an application of SVM in small-set off-line handwritten Chinese characters recognition based on energy features of the wavelet sub-image.

2. Theory basic

2.1 Image wavelet-based features

The wavelet transform of an image $g(x,y)$ is a linear transformation of this image onto a set of

its inner products with basis functions $\Psi_{a,b}(x,y)$, called wavelets. These wavelets are generated from dilation and translation of a mother wavelet $\Psi(x,y)$ as follows:

$$\psi_{a,b}(x,y)=\frac{1}{\sqrt{a_x a_y}}\psi(\frac{x-b_x}{a_x},\frac{y-b_y}{a_y}) \qquad (1)$$

where $a=(a_x,a_y)^T$, a>0, and $b=(b_x,b_y)^T$ are the scaling, and the translation parameters respectively in x-y axes.

The wavelet transform of function f (x, y) is defined as

$$\Omega(f(x,y))=f(x,y)*\psi(x,y) \qquad (2)$$

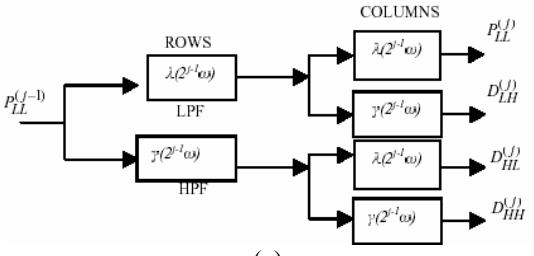
where, '*' stands for the two-dimensional convolution operator.

Mallat has shown that the WT of one-dimensional signal can be realized using a fast pyramid algorithm by performing signal dyadic sub sampling across successive scales, and using a bank of digital low pass and high pass filters [3]. Signal reconstruction is obtained by up sampling the signal along those successive scales, and using the same decomposition filters for orthogonal basis functions, or different filters for biorthogonal basis functions.

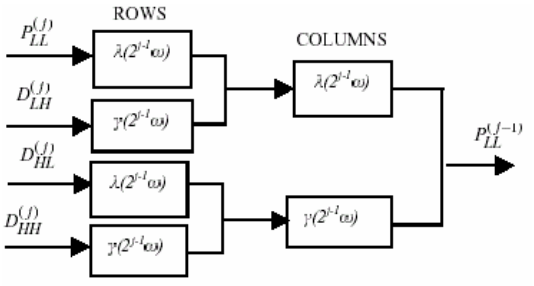
Actually, the redundant wavelet transform is widely used in image analysis because this redundancy can be exploited for target / background feature extraction along different scales.

The wavelet image decomposition is performed by applying the algorithm on the image's rows first and then on its columns [3].

Figure 1 shows the image decomposition and reconstruction using the undecimated one-dimensional, orthogonal pyramid algorithm at scale level j. It is evident from this figure that, at each scale level, one obtains four images of the original input image. One of these resultant images is the approximated image P_{LL}^{j}, while the others are the detailed images D_{LH}^{j}, D_{HL}^{j} and D_{HH}^{j}. The transfer functions of low-pass and high-pass filters are given by $\lambda(2^{j-1}\omega)$ and $\gamma(2^{j-1}\omega)$ respectively.



(a)



(b)

Figure1 Undecimated pyramid algorithm for (a) image decomposition and (b) image reconstruction.

If we repeat the process of wavelet decomposition in sub-image P_{LL}, then we can obtain the sub-image $P_{LL}^{(2)}$, $D_{LH}^{(2)}$, $D_{HL}^{(2)}$ and $D_{HH}^{(2)}$, as shown in Figure2.

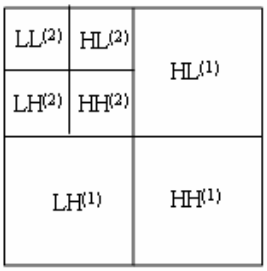


Figure2. Sub-image of wavelet decomposition at two scales

In this paper, an image is decomposed by wavelet transform into several sub-images, and therefore, the average energies of the sub-image are computed. We can use them as a set of the features to classify the Chinese handwritten characters. And two feature vectors are utilized, namely: (l) wavelet energy distribution feature (Fd) vector, and (2) wavelet energy distribution proportion feature (Fdp) vector.

The average energy of an image of size N*N is defined by

$$Energy \quad f=\sum_{m=1}^{N}\sum_{n=1}^{N}\frac{f^2(m,n)}{N^2} \qquad (3)$$

Different characters-images have different average energies.

After the multi-scale wavelet decomposition, the original document image has been decomposed into detailed sub-images LH, HH, and HL. The average energy distribution of the detailed sub-images at k-th level of wavelet decomposition can be defined by

$$ELH^{(k)} = \sum_{m=(N/2^k)+1}^{N/2^{k-1}} \sum_{n=1}^{N/2^k} \frac{(LH^{(k)}(m,n))^2}{(N/2^k)^2} \quad (4)$$

$$EHL^{(k)} = \sum_{m=1}^{N/2^k} \sum_{n=(N/2^k)+1}^{N/2^{k-1}} \frac{(HL^{(k)}(m,n))^2}{(N/2^k)^2} \quad (5)$$

$$EHH^{(k)} = \sum_{m=(N/2^k)+1}^{N/2^{k-1}} \sum_{n=(N/2^k)+1}^{N/2^{k-1}} \frac{(HH^{(k)}(m,n))^2}{(N/2^k)^2} \quad (6)$$

k=1,…, $\log_2 N$

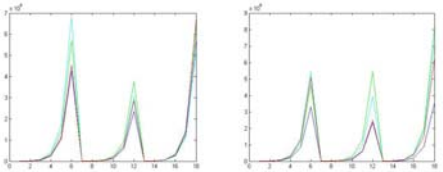
where,

$ELH^{(k)}$, $EHL^{(k)}$, $EHH^{(k)}$ (k=1,…, $\log_2 N$), respectively are multi-scale wavelet energy distribution features, which are called wavelet energy distribution features. They become the components of the wavelet energy distribution feature (Fd) vector:

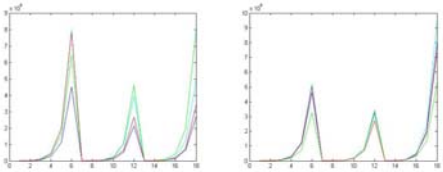
$$F_d = \left(ELH \;\; EHL \; EHH \right)$$

(k=1,…, $\log_2 N$)

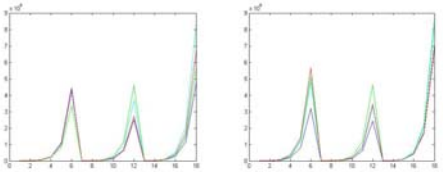
Some examples of the wavelet distributions of Fd in eleven characters, number in Chinese: form 0 to 10, are illustrated in Figure 3:



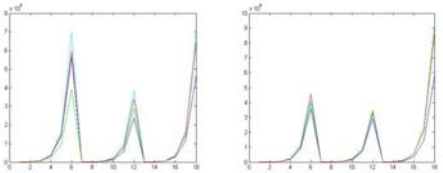
(a)One (b)Two



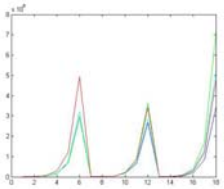
(c)Three (d)Four



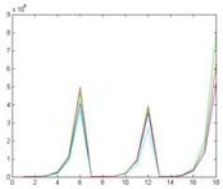
(e)Five (f)Six



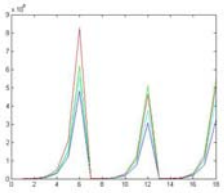
(g)Seven (h)Eight



(i)Nine



(j)Ten



(k)Zero

Figure3: Examples of the wavelet energy distributions features Fd of the sub-image

The wavelet energy distribution proportion features (Fdp) consist of three sets of features, namely:

$$F_{dp} = \left(EPLH \;\; EPHL \; EPHH \right)$$

(k=1,…, $\log_2 N$)

where, $EPLH^{(k)}$, $EPHL^{(k)}$ and $EPHH^{(k)}$ stand for the energy distribution proportions of sub-image LH, HL and HH at the *k*-th level of the wavelet decomposition respectively.

There are following facts: (l) Sub-image LH carries much more horizontal information than that of HL as well as HH. (2) Sub-image HL has more vertical features than those of LH and HH. (3) Sub-image HH contains more features in the direction of diagonal comparing with other two sub-images. In order to characterize the oriented property of the different character-images, we consider the ratio of the energy distribution of a specific detailed sub-image to that of all detailed sub-images at the same wavelet scale.

To illustrate the horizontal property of the different characters, the ratio of the energy distribution of LH sub-image, (ELH), to those of all detailed sub-images at the same wavelet scale, (ELH + EHL + EHH), is defined as energy distribution proportions of sub-image LH, which is symbolized by EPLH. It can be computed by

$$EPLH^{(k)} = \frac{ELH^{(k)}}{ELH^{(k)} + EHL^{(k)} + EHH^{(k)}} \quad (7)$$

(k=1,…, $\log_2 N$)

Using the similar way, we have $EPHL^{(k)}$ – the energy distribution proportion of sub-image HL at the *k*-th level of the wavelet decomposition, which is defined as

$$EPHL^{(k)} = \frac{EHL^{(k)}}{ELH^{(k)} + EHL^{(k)} + EHH^{(k)}} \quad (8)$$

(k=1,…, $\log_2 N$)

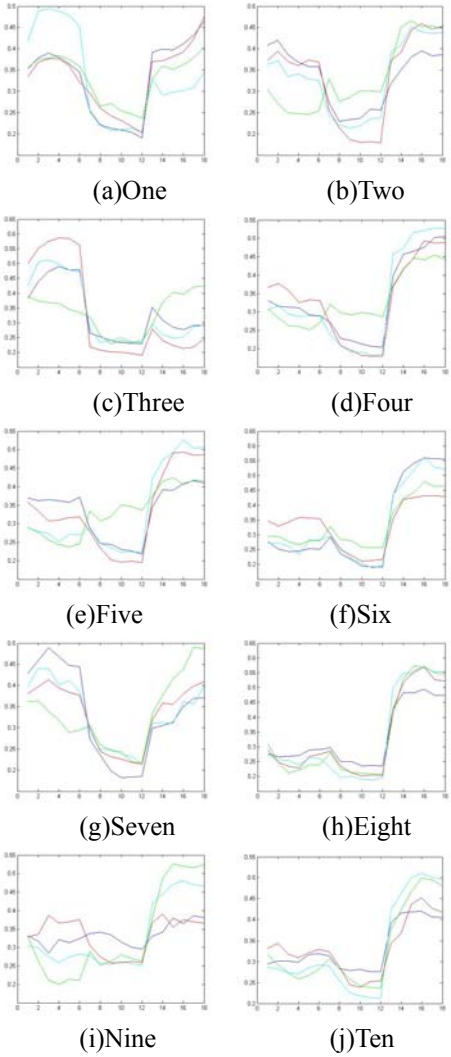
Similarly, the ratio of the energy distribution of HH sub-image, (EHH), to those of all detailed sub-images at the *k*-th wavelet scale is defined as $EPHH^{(k)}$– the energy distribution proportion of sub-image HH at the *k*-th level of the wavelet decomposition, and is represented by

$$EPHH^{(k)} = \frac{EHH^{(k)}}{ELH^{(k)} + EHL^{(k)} + EHH^{(k)}} \quad (9)$$

(k=1,…, $\log_2 N$)

which can characterize the diagonal property of the different characters.

Some examples of the wavelet distributions of Fdp in eleven characters, number in Chinese: form 0 to 10, are illustrated in Figure 4:



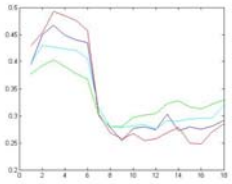
(a)One (b)Two

(c)Three (d)Four

(e)Five (f)Six

(g)Seven (h)Eight

(i)Nine (j)Ten



(k)Zero

Figure4: Examples of the wavelet energy distributions proportion features Fdp of the sub-image

As Figure 3 and Figure 4 shown above, we can find the wavelet energy distribution proportion features (Fdp) for a character image are more stable than the wavelet energy distribution features (Fd).

2.2 SVM theory

Support Vector Machines (SVM) has become a hot research topic in the international machine learning field because of its excellent statistical learning performance. It has been widely applied to pattern recognition. Simply, SVM can be comprehended as follows: it divides two specified training samples which belong to two different categories through constructing an optimal separating hyperplane either in the original space or in the projected higher dimensional space. The principle of constructing the optimal separating hyperplane is that the distance between each training sample and the optimal separating hyperplane is maximum.

There are two conditions, linearly separable situation and linearly inseparable situation, under which the principle of SVM is introduced as follows separately.

Under the linearly separable condition, a binary classification task is taken into account. Let $\{(x_i, y_i)\}$ ($1 \le i \le N$) be a linearly separable set. Where, $x_i \in R^d$, $y_i \in \{-1,1\}$, and y_i are labels of categories. The general expression of the linear discrimination function in *d*-dimension space is defined as $g(x) = w*x + b$, and the corresponding equation of the separating hyperplane is as follows: $w*x + b = 0$.

Normalize $g(x)$ and make all the x_i meet $g(x) \ge 1$, that is, the samples which are closed to the separating hyperplane meet $|g(x)| = 1$ .Hence, the separating interval is equal to $2/\|w\|$, and solving the optimal separating hyperplane is equivalent with minimizing $\|w\|$ .The object

function is as follows:

$$\text{Min}\ \Phi(w) = \frac{1}{2}\|w\|^2 \tag{10}$$

Subject to the constraints:

$$y_i(w \bullet x_i + b) \geq 1, i=1,\ldots, N \tag{11}$$

When adopting Lagrangian algorithm and introducing Lagrangian multipliers $\alpha = \{\alpha_1, ..., \alpha_N\}$, the problem mentioned above can be converted into a quadratic programming problem and the optimal separating hyperplane can also be solved.

Where, $w = \sum_i \alpha_i y_i x_i$,

x_i is the sample only appearing in the separating interval planes. These samples are named support vectors and the classification function is defined as follows:

$$f(x) = \text{sgn}\left(\sum_i \alpha_i y_i x_i \bullet x + b\right) \tag{12}$$

Under the linearly inseparable condition, on the one hand, SVM turns the object function into as follows through introducing slack variable ξ and penalty factor C

$$\text{Min} \quad \Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{N} \xi_i\right) \tag{13}$$

On the other hand, SVM converts the input space into a higher dimensional space through linear transform in which the optimal separating hyper-plane can be solved.

Additionally, the inner product calculation under the linearly separable condition is turned into $K(x_i, x_j) = \Phi(x_i) \bullet \Phi(x_j)$, where $K(x_i, x_j)$ is defined as inner product in Hilbert space and it is named kernel function here. Thus the final classification function is represented as follows.

$$f(x) = \text{sgn}\left(\sum_i \alpha_i y_i K(x, x_i) \bullet x + b\right) \tag{14}$$

The SVM was originally designed for binary classification. How to effectively extend it for multi-class classification is still an on-going research issue. Currently there are two types of approaches for multi-class SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation, stand for 'one-against-all' and 'one-against-one',

3. Experiments

Extensive experiments have been carried out to test the algorithm. In this experimental system, the recognition of the Chinese character to have 11 kinds, every kind of Chinese character has collected 100 different written samples. Altogether has the sample number is 1100. There are some pro-processing steps doing before the feature extraction and classification.

The characteristic extraction: Select tentative data for handwritten Chinese character. In this paper we select the handwritten Chinese number characters, which are widely used in banking and post department, which shows in figure 5. Preprocess these Chinese characters using the model match, binarization, thinning, denoising and so on. Use the appropriate characteristic extraction method to extract the Chinese character's vector characteristic as the support vector machine input sample.

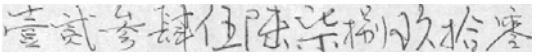


Figure 5: An example of the characters

Divide each Chinese character image those were binarizated and normalizatied, again do wavelet transformation to these divided images, obtain the wavelet coefficients of the Chinese character as the vector characteristic. As we have discussed in part Ⅱ, these eleven number Chinese characters have different wavelet energy distribution feature (Fd) vector, and wavelet energy distribution proportion feature (Fdp) vector, which have shown in Figure 2 and Figure3.

Data format normalization: Make normalized processing of the wavelet energy features, to adapt the request of SVM training software.

The overlapping confirmation basic idea is: Divide the sample collection to into two subsets, one group (training subset) trains the classifier, another group (examination subset) exam the training (which estimate the error of the trained classifier), then the basis the exam result to estimate the effect of classifier, then adjusts the related parameter of the classifier. Carries on the training and adjust again as before. When the error achieves the ideal value, then obtains classifier for goal classifier [11].

The software LibSVM is proposed for hand-written Chinese characters training. The multi-class method is 'one-against-one'.

The experiment procedure is as follows: First, input the samples, and do some normalization processing to remove noise and divide the samples to the wanted size. Second, do the

wavelet transform and extract the wavelet energy features Fd and Fdp. Third, normalize the features values for the SVM staining. Forth, use RBF function $K(x,y)=e^{-\gamma\|x-y\|^2}$ as kernel function, train the data to create a model with svmtrain by overlapping confirmation. Predict new input data with svmpredict and get the result.

Table 1 has listed the result of recognition rate of using the SVM classifier based on the wavelet energy features. As it shows, we can find the Fdp features are more stable than the Fd features. And it can prove the result that the features we extract in part Ⅱ, especially the Fdp of "zero" and "eight" are very different from others as in Figure 3 shows.

Accuracy of classification	Fd	Fdp
One	93.3%	96.5%
Two	90.6%	94.4%
Three	92.2%	95.2%
Four	87.3%	93.9%
Five	91.4%	93.1%
Six	89.5%	94.4%
Seven	92.7%	94.5%
Eight	95.2%	98.8%
Nine	92.5%	95.5%
Ten	94.6%	96.4%
Zero	96.1%	99.4%
Average	92.3%	95.7%

Table 1 The recognition rate of SVM based on the wavelet energy features

4. Conclusions and Future Works

In this paper, we introduce a character recognition method based on wavelet energy features and SVM classification. The experiments show that these wavelet-based features can effetely classify the training samples, and the SVM is a good recognition method.

According to different case and application, use different multi-class methods and feature extraction methods, the effect would be better.

The SVM kernel function is due to the different case. The SVM theory didn't give a way to say which kernel function is the best kernel function. So there are many research in kernel function. Reproducing kernel and the reproducing kernel Hilbert space (RKHS) play an important role in function approximation and regularization theory.

Riesz kernel, especially wavelet kernel, is widely applicable. It is operation significance to format the reproducing kernel suited to this approximation function characteristic.

In future works, we would research the kernel function of SVM, improve the features extraction method, and according to handwritten Chinese characters, improve adapted SVM multi-class methods.

References

1. M. Unser and M. Eden, Multiresolution feature extraction and selection for texture segmentation, IEEE Trans. Patt. Anal. Mach. Intell., vol. 11, pp. 717-728, 1989
2. A.K. Jain, Learning texture discrimination masks, IEEE Trans. Patt. Anal. Mach. Intell, vol. 18, no. 2, pp. 195-205,1996
3. S. Mallat, A Wavelet tour of signal processing, New York: Academic
4. B.S. Manjunath and W.Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Patt. Anal. Mach. Intell, vol. 18, no. 8, pp. 837-842, 1996
5. C.C. Chen and D.C. Chen, Multiresolutional Gabor filter in texture analysis," Patt. Rec. Lett., vol. 17, no. 10, pp. 1069-1076,1996
6. S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Patt. Anal. Mach. Intell., vol. 11, no. 7, pp. 674-693, 1989
7. I. Daubechies, Ten Lectures on Wavelets, Capital City Press, Montpellier, Vermont
8. A. Laine and J. Fan, Frame representations for texture segmentation, IEEE Trans. Im. Process., vol. 5, no. 5, pp. 771-780, 1996
9. G. Van de Wouwer, P. Scheunders, S. Livens, and D. Van Dyck, Wavelet correlation signatures for color texture characterization, Patt. Rec, 1998, to appear
10. Bian Zhaoqi, Zhang Xuegong, Pattern Recognition tsinghua University Press
11. Vanpnik V.N, Statistical Learning Theory [M]. New York: John Wiley and Sons

Learning Nominal Data Similarities for Kernel Methods

Victor Cheng and C.H.Li

Abstract

In many kernel algorithms, while similarities of continuous or ordinal attributes can be dealt easily, nominal attributes are difficult to handle. A popular approach is to represent a nominal attribute with a 0/1 overlap metric. This transformation, however, assume the attribute values are independent of each and ignore possible interactions. In this paper, we propose an approach for learning similarities for nominal attributes from labeled data. Similarity matrices are learnt by minimizing the margin of the separating hyperplane in the input space. This approach is further extended to feature space by dealing the minimizing margin problem in input space with iterative gradient method. The resultant matrices provide the metric interpretation of nominal attributes.

1 Introduction

Many machine learning algorithms rely on the similarity (or dissimilarity) measure between patterns. For example, Nearest-neighbor classifier evaluates the dissimilarities using Euclidean distance in finding neighbors and SVM [1, 2] uses inner product, in feature space, to evaluate pairwise similarities. Most approaches employed, however, only work with continuous attributes. Nominal attributes have to be processed to numerical values before the evaluation. Transforming a nominal attribute to numerical values is nontrivial because of lack of clear metric between any two nominal values. For example, an attribute describing the shape of object may has a value $a_i \in \{square, triangle, rectangle, circle\}$. It is not easy to evaluate the similarity between, say a square and a circle. A simple and popular approach is to use overlap metric. Under this metric, a nominal attribute is represented by a sequence of binary digits with each bit represents each possible value. The bit corresponds to the attribute nominal value is set to one while the rest bits are set to zero. With this scheme, any two possible values of a nominal attribute, a_i, a_j, has similarity one if they are equal and zero otherwise. This approach assumes all possible nominal values of an attribute are of equal distance to each other and located at different axes of the unit Euclidean space which have dimension equals to the number of possible values. Therefore, different degrees of similarities cannot be represented. As an example, it may be more appropriate for having the value of the $circle$ closer to $square$ than $triangle$. Under the linear discriminative frameworks with the separating hyperplane given by (1), the performance of this approach is not too bad because $\mathbf{w}$ obtained during the training process somehow explore the metric information of the nominal values.

$$f(x) = \mathbf{w}'\mathbf{x} + b \tag{1}$$

where $\mathbf{x}$ is the pattern vector.

Consider a nominal attribute $\mathbf{A}$ having m possible values and the similarity between values is given by the matrix

$$\mathbf{S_A} = \begin{pmatrix} s^{11} & s^{12} & \dots & s^{1m} \\ s^{21} & s^{22} & \dots & \\ \vdots & \vdots & & \vdots \\ s^{m1} & \dots & \dots & s^{mm} \end{pmatrix}. \tag{2}$$

In (2), superscript is used in identifying matrix elements because subscript is reserved for another usage in later sections. Assume the similarity is symmetric (i.e. $s^{i,j} = s^{j,i}$), there are total $m(m+1)/2$ variables in $\mathbf{S_A}$ accounting all possible similarities between any pairs of values. It is trivial that the bit representation of the overlap metric cannot take into account of all situations as $m(m+1)/2 > m$ whenever $m \geq 2$. Noting that it is often to assume the diagonal elements $s^{i,i}$ have same value, however this assumption is not employed in this paper.

Another very popular real-valued metric for nominal attributes is using the value difference metric (VDM) [3], or its variants [4, 5]. The dissimilarity between two attribute values a_i and a_j is defined as

$$d(a_i, a_j) = \omega(a_i) \sum_{c \in C} (P(c|a_i) - P(c|a_j))^2, \tag{3}$$

where C is the set of all class labels and $P(c|a_i)$ is the class conditional probability of the class $c \in C$ given a_i, and $\omega(a_i)$ is a weighting factor which attempts to give higher weight to an attribute value that is significant to class discrimination. Referring to (3), it is not hard to see that the

metric depends on the class discriminative ability of individual nominal value. Since the dissimilarity evaluation in (3) is not symmetric, and it is actually not a metric measure. More importantly, it considers the attributes individually and independently and sometimes causes the attributes useless. A simple example is all class conditional probability $P(c|a_i)$ have equal values and thus yields zero which is clearly undesirable.

In this paper, an approach for learning similarities between attribute values are proposed. Each nominal attribute is transformed to a real value attribute array. The inner product between two arrays equals the similarity between two values of that attribute. The rest of the paper is organized as follows. Section 2 gives the problem definition. The method of learning attribute similarity matrix is proposed in Section 3. Section 4 describes the extension of the method in feature spaces.

2 Problem Definition

Without loss of generality, the learning problems considered in this paper are concerned with patterns that have a number of continuous attributes and one and only one nominal attribute. Problems with more than one nominal attributes can be tackled with the proposed approach, described in latter sections, accordingly. In addition, only the nominal values similarity issue is considered in the sequel, the dissimilarity issue can be derived in similar manner.

Given a set of labeled training data $\chi = \{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbf{R}^d \cup \mathbf{A}$, $\mathbf{A} = \{a^1, \ldots, a^m\}$ is a set of nominal values. Let $\mathbf{x_i} = (\hat{\mathbf{x}}_\mathbf{i}, a_i^{\hat{}})$ where $\hat{\mathbf{x}}_\mathbf{i} \in \mathbf{R}^d$ and $a_i^{\hat{}} = a^{\hat{}}$. Similarity between two nominal values $a^{\hat{}}$ and $a^{\hat{j}}$ is denoted by $s^{\hat{i}\hat{j}}$. For any two patterns $\mathbf{x_i}$ and $\mathbf{x_j}$, the inner product in the input space is given by

$$\begin{aligned} < \mathbf{x_i}, \mathbf{x_j} > &= \sum_{l=1}^{d} x_{il}x_{jl} + s^{\hat{i}\hat{j}} \\ &= < \hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{x}}_\mathbf{j} > + s^{\hat{i}\hat{j}} \end{aligned} \quad (4)$$

The goal is to find a positive definite similarity matrix $\mathbf{S_A} \in \mathbf{R}^{mxm}$ such that the learner has low generalization error in classification. After having $\mathbf{S_A}$, it can be factorized into $\mathbf{LL}'$ where $\mathbf{L} \in \mathbf{R}^{mxm}$ is a symmetric matrix. If $\mathbf{b_i} \in \{0,1\}^m$ is a vector of bits represents the overlap metric of the attribute $a_i^{\hat{}}$, $\mathbf{b_i'L}$ will be the attribute vector representing $a_i^{\hat{}}$, where $\mathbf{b_i'}$ is the transpose of $\mathbf{b_i}$. Hence the input pattern $\mathbf{x}_i$ can be transformed to an all continuous value vector.

$$\mathbf{x}_i = (\hat{\mathbf{x}}_\mathbf{i}', \mathbf{b_i'L}) \quad (5)$$

since

$$\begin{aligned} < \mathbf{x_i}, \mathbf{x_j} > &= < \hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{x}}_\mathbf{j} > + (\mathbf{b_i'L})(\mathbf{b_j'L})' \\ &= < \hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{x}}_\mathbf{j} > + s^{\hat{i}\hat{j}}. \end{aligned} \quad (6)$$

3 Learning the Nominal Attribute Similarity Matrix

One approach to learn the attribute similarity matrix $\mathbf{S_A}$ is to use the SVM formulation. Maximizing the margin of the separating hyperplane involves in solving the optimization problem:

$$\begin{cases} \min_{\mathbf{w},\mathbf{S_A}} & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to} & y_i(\mathbf{w}'\mathbf{x_i} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \\ & s^{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m \end{cases} \quad (7)$$

As there are two set of parameters, $\mathbf{w}$ and $\mathbf{S_A}$, involved in the problem. An iterative scheme similar to the EM algorithm is employed to solve (7).

1. Initialize $\mathbf{S_A}$:
   - all diagonal elements equal to $1.0+$ a small random value.
   - initialize other elements to a small random value with $s_{ij} = s_{ji}$.
2. Solve (7) with assuming $\mathbf{S_A}$ is constant.
3. Update the matrix $\mathbf{S_A}$.
4. Go to the step 2 until changes of $\parallel \mathbf{S_A} \parallel^2 \leq \epsilon$.

In step 3, updating the matrix $\mathbf{S_A}$ can be done by solving the optimization problem below. It is same as (7) except only $\mathbf{S_A}$ is the parameter.

$$\begin{cases} \min_{\mathbf{S_A}} & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to} & y_i(\mathbf{w}'\mathbf{x_i} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n. \\ & s^{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m \end{cases} \quad (8)$$

The dual of (8) is

$$\begin{cases} \max_{\mathbf{S_A},\alpha} & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j < \mathbf{x_i}, \mathbf{x_j} > \\ \text{subject to} & \\ & \sum_{i=1}^n \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C \quad i = 1, \ldots, n, \\ & s^{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m \end{cases} \quad (9)$$

Let $L(\mathbf{X}, \alpha, \theta)$ be the Lagrangian of (9), where $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ and θ is another set of Lagrange multipliers catering other constraints. For any matrix element $s^{\hat{i}\hat{j}}$ in $\mathbf{S_A}$,

$$\frac{\partial L}{\partial s^{\hat{i}\hat{j}}} = \frac{\partial L(X)_{\alpha,\theta \text{ treated as constants}}}{\partial X}\frac{\partial X}{\partial s^{\hat{i}\hat{j}}} + \frac{\partial L(\alpha,\theta)_{X \text{ treated as constants}}}{\partial \alpha}\frac{\partial \alpha}{\partial s^{\hat{i}\hat{j}}}. \quad (10)$$

If α is the solution obtained from step 2, then

$$\frac{\partial L(\alpha,\theta)_{X \text{ treated as constants}}}{\partial \alpha} = 0 \tag{11}$$

Hence, (9) can be solve as if all α_i are constant. With this trick and substituting (4) into (9) and ignoring the constant terms, the optimization problem becomes

$$\begin{array}{ll} & \min_{\mathbf{S_A}} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j s^{\hat{i}\hat{j}} \\ \text{subject to} & \\ & s^{ij} \geq 0, \quad i = 1,\ldots,m, \quad j = 1,\ldots,m \end{array} \tag{12}$$

In (12), there are only one constraint $s^{ij} \geq 0$ left, as other of them in (9) are automatically satisfied due to α_i are evaluated from step 2.

Since s^{ij} in (12) is bounded below only, it is an unbounded problem, regularization should be introduced to avoid over-fitting. Moreover, too large variation in $\mathbf{S_A}$ may violate the assumption that support vectors are still support vectors during the optimization. One common method is to modify the objective function so that it is penalized if $\mathbf{S_A}$ is far different from the identity matrix. For example, (12) can be revised to

$$\begin{array}{ll} \min_{\mathbf{S_A}} & \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j s^{\hat{i}\hat{j}} + \gamma \parallel \mathbf{S_A} - \mathbf{I} \parallel_F . \\ \text{subject to} & \\ & s^{ij} \geq 0, \quad i = 1,\ldots,m, \quad j = 1,\ldots,m, \end{array} \tag{13}$$

where $\gamma \geq 0$ is a regularization constant, $\mathbf{I}$ is the identity matrix having same dimension as $\mathbf{S_A}$, and $\| \cdot \|_F$ denotes the Frobenius norm. Noting the objective function in (13) is quadratic and convex, it can be solved easily.

As stated previously we prefer the attribute similarity matrix to be positive definite. Perturbation on the matrix may be required because it obtained with the above framework is not guaranteed positive definite. Two common techniques can be applied. In the first technique, the spectral decomposition of the matrix is evaluated and then the matrix is reconstructed with only all positive eigenvalues and the associated eigenvectors. The eigenvectors associated with negative eigenvalues are ignored. Let the spectral decomposition of the matrix $\mathbf{S_A}$ be

$$\mathbf{S_A} = \sum_{i=1}^{m} \lambda_i \mathbf{v_i} \mathbf{v_i'}, \tag{14}$$

then the perturbed version of $\mathbf{S_A}$ is

$$\mathbf{S_A} = \sum_{i=1,\lambda_i>0}^{m} \lambda_i \mathbf{v_i} \mathbf{v_i'} \tag{15}$$

where $\mathbf{v_i}$ is the eigenvector associated with eigenvalue λ_i. As the new $\mathbf{S_A}$ now has all positive eigenvalues, it is positive definite. The another technique is a bit simpler. A symmetric matrix can be turned to positive definite by adding a value just greater than the absolute value of its most negative eigenvalues to all the diagonal elements. If λ_j is the most negative eigenvalue of $\mathbf{S_A}$, the matrix can be converted to positive definite by

$$\mathbf{S_A} = \mathbf{S_A} + (\mid \lambda_j \mid + \epsilon)\mathbf{I}, \tag{16}$$

where ϵ is a small positive number. After the perturbation, the matrix should be normalized so that its weighting among the attributes is not changed significantly.

4 Learning the Similarity for Feature Spaces

As the feature space usually has very high dimension or even infinite dimension, solving the optimization problems using the procedures described in the previous section to compute $\mathbf{S_A}$ is very difficult. Moreover, many feature spaces are formed from cross products (or other more complex function) among pattern attributes and hence the problems almost cannot be solved directly in feature space. Fortunately, they can be simplified by using the kernel trick $K(\mathbf{x_i}, \mathbf{x_j}) =< \Phi(\mathbf{x_i}), \Phi(\mathbf{x_j}) >$ where $\Phi(\mathbf{x})$ is the feature map mapping $\mathbf{x}$ to the feature space (a Reproducing Kernel Hilbert Space, RKHS) associated with the kernel function $K(\mathbf{x}, \mathbf{y})$.

By exploiting the kernel trick, the procedures described in Section 3 can still be used after some modification. Step 2 can be done with SVM packages such as SVM^{light} [6] . Nevertheless, the optimization (12) in step 3 is revised to

$$\begin{cases} \min_{\mathbf{S_A}} \mathbf{J} = \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) + \gamma \parallel \mathbf{K_A} - \mathbf{I} \parallel_F \\ \text{subject to } s^{ij} \geq 0, \quad i = 1,\ldots,m, \quad j = 1,\ldots,m. \end{cases} \tag{17}$$

where $\mathbf{K_A}$ is the kernel matrix. As $K(\mathbf{x}, \mathbf{y})$ is usually not linear, (17) is in fact a non-linear (and non-convex) optimization problem in $s^{\hat{i}\hat{j}}$ and it has no close-form solution. However, the objective function can be optimized locally with gradient methods. Updating s^{ij} is done by

$$s^{\hat{i}\hat{j}}_{(t+1)} = s^{\hat{i}\hat{j}}_{(t)} - \beta \frac{\partial \mathbf{J}}{\partial s^{\hat{i}\hat{j}}} \tag{18}$$

where β is the step size control factor.

References

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK 2000.

[2] B. Schölkopf, "Statistical Learning and Kernel Methods", Technical Report MSR-TR-2000-23, Microsoft Research, 2000.

[3] C. Stanfill, D. Waltz, ”Towards memory-based reasoning”, Commun. ACM 29(12),1986,1213-1228.

[4] S. Cost, S. Salzberg, ”A weighted nearest neighbor algorithm for learning with symbolic features”, Mach. Learning,10, 1993,57-78.

[5] D.R. Wilson, T.R. Martinez, ”Improved heterogeneous distance function, J. Artif. Intell. REs. 6, 1997,1-34.

[6] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

Orthogonal Nonnegative Matrix Factorization based POMDP Compression

Xin LI

Abstract

Partially observable Markov decision process (POMDP) is a commonly adopted framework to model planning problems for agents to act in a stochastic environment. Obtaining the optimal policy of POMDP for large-scale problems is known to be intractable, where the high dimension of its belief state is one of the major causes. The use of the compression approach has recently been shown to be promising in tackling the curse of dimensionality problem. In this paper, an orthogonal NMF based value-directed belief compression technique is proposed to compute the factored POMDP more accurately than our former work. We apply an orthogonal non-negative matrix factorization (NMF) based projection to the sampled belief sets for dimension reduction and compute the optimal policy in the low-dimensional projected belief state space. Meanwhile, we propose an oversampling method to enhance the conventional trajectory-based belief sampling regarding its belief space coverage. The proposed algorithm has been evaluated using the synthesized navigation problems and positive results were obtained.

1 Introduction

Building intelligent agents which can make optimal decisions in a stochastic environment is known to be popular and important in wide area. The problem is how to compute the optimal policy for an agent to decide its next action based on some feedback observed from the environment so as to maximize its long-term reward and at the same time complete a given mission. The agent policy can in general be defined as a mapping of its observed information to the recommended actions to be performed. The ideal scenario of the problem is that the stochastic environment can be accurately modelled by a set of abstracted fully observable true states. However, in many real situations, the agents cannot have full observation but only partial one for reflecting the current true state. This partially observable situation makes the optimal decision making more challenging as the agent needs ways to efficiently abstract its inflatable observation history for its decision on next action.

An Markov decision process (MDP) is a common model used in a fully observable environment which is characterized by a finite set of states $\mathcal{S}$, a finite set of actions $\mathcal{A}$, a set of corresponding state transition probabilities $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$ and a reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$. Solving an MDP problem means finding an optimal policy which maps each state to an action so as to achieve the best long-term reward. POMDP is the model that extends MDP to the partially observable situation. The probability distribution over the unobservable true states $\mathcal{S}$, commonly called the (*belief space*), is estimated via Bayesian updating for summarizing the observation history. The belief states form a continuous state space of $|S| - 1$ dimensions (later on called belief space). The policy of POMDP becomes a mapping between belief states and actions.

To compute the optimal policy, the value iteration technique [9] is typically adopted where the value function to optimize iteratively is the expected reward received by the agent. The best complexity bound for obtaining the optimal policy of POMDP with t step ahead considered is $O(\gamma_{t-1}^{|Z|})$[1] where γ_i is the space complexity of the value function at the i^{th} iteration. Even though there exist some computational shortcuts which take the advantage of the piecewise linear and convex (PWLC) property of the value function (e.g., the witness algorithm [1]), large-scale POMDP problems are still widely considered to be computationally intractable.

In this paper, we propose a way which is inspired by two recently proposed ideas, namely belief compression and value-directed compression, to make POMDP problems tractable. Also we argue that clustering the belief states prior to computing the optimal policy can further reduce the complexity as the task can be decomposed in a problem-specific manner. In particular, we apply an orthogonal nonnegative matrix factorization (NMF) to the sampled belief set and derive the corresponding approximated POMDP in low-dimensional space (including the low-dimensional reward and state transition functions), resulting in a factored POMDP. The use of orthogonal NMF can guarantee the elements of the low-dimensional representation of belief states to be non-negative, which is important as the representations should be by themselves probability distributions. Also the orthogonal NMF based method can guarantee an

accurate recovery of value iteration function in the high-dimensional space. To compute the optimal policy for the factored POMDP, the point-based value iteration technique is used. Meanwhile, we propose an oversampling method to enhance the conventional trajectory-based belief sampling regarding its belief space coverage, thus improve the accuracy of value functions' computation in low-dimensional space. The proposed algorithm has been evaluated using navigation problems and positive results were obtained.

This paper is organized as follows. Section 2 describes the problem formulation and the compression related methods for addressing the intractability issue. Section 3 describes the details of the proposed orthogonal based value-directed compression and Section 4 describes a supplementary sampling method in low-dimensional space which could reduce the low-dimensional belief spacing so to improve the low-dimensional policy through an. The experimental results can be found in Section 5. Concluding remarks and future research directions are included in Section 6.

2 Literature Review on Compression over POMDP

2.1 Formulation

A POMDP model is characterized by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, O, R >$, which contains a finite set of real states $\mathcal{S}$, a finite set of agents' actions $\mathcal{A}$, the state transition probabilities $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$, a reward function which depends on the action and the state $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, a finite set of observations $\mathcal{Z}$ and a set of corresponding observation probabilities $O : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{Z})$. Solving POMDP problems typically makes use of the belief state concept. A belief state is defined as a probability mass function over the current state, given as $b = (b(s_1), b(s_2), ...b(s_{|S|}))$, where $s_i \in \mathcal{S}, b(s_i) \geq 0$, and $\sum_{s_i \in \mathcal{S}} b(s_i) = 1$. $b_j = SE(b_i, a, z)$ is defined using Eqs.(1) and (2), given as

$$\begin{aligned} b_{t+1}(s_j) &= P(s_j|z, a, b_t) \\ &= \frac{O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b_t(s_i)}{P(z|a, b_t)} \end{aligned} \quad (1)$$

$$P(z|a, b_t) = \sum_{s_j \in \mathcal{S}} O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b_t(s_i). \quad (2)$$

The reward function of the j^{th} belief state b_j can then be computed as $\rho(b_j, a) = \sum_{s_i \in \mathcal{S}} b_j(s_i) R(s_i, a)$. Also, the transition function over the belief states becomes $\tau(b_i, a, b_j) = p(b_j|b_i, a)$ (see [1] for more details). To compute the optimal policy $\pi : \mathcal{R}^{|\mathcal{S}|} \rightarrow \mathcal{A}$ iteratively, a value function is typically involved, given as

$$V(b_i) = max_a[\rho(b_i, a) + \gamma \sum_{b_j} \tau(b_i, a, b_j) V(b_j)] \quad (3)$$

$$\pi^*(b_i) = argmax_a[\rho(b_i, a) + \gamma \sum_{b_j} \tau(b_i, a, b_j) V(b_j)]. \quad (4)$$

where γ is the discounting factor for the past history. In practice, it is common to have the optimal policy represented by a set of linear functions (so called α vectors) over the belief space, with the maximum "envelop" of the intersections forming the overall value function.

2.2 Belief Compression

Belief compression is a recently proposed approach [5] to address the curse of dimensionality problem of POMDP, by reducing the sparse high-dimensional belief space to a low-dimensional one via a projection. The idea behind is to explore the redundancy in computing the optimal policy for the entire belief space which is typically sparse. Using a sample of belief states as the training set, exponential principal component analysis (EPCA) was adopted to characterize the originally high-dimensional belief space using a compact set of belief state bases. This approach has been found to be effective in making some POMDP problems much more tractable. However, as the transformation used is non-linear, the value function of the projected belief states is no longer piecewise linear. Many efficient algorithms which take the advantage of PWLC property of value function become not applicable. In [5], the sampled belief states in the projected belief space were used as the states of a newly created MDP and one can compute the policy for that MDP instead. The limitation is that the quality of the resulting policy now depends not only on that of the belief compression, but also that of the grid-like approximation for policy computation. The latter one is problem dependent, making the applicability of this non-linear belief compression approach restricted. But of course, non-linear transformation is more effective in digging out the structure of the high-dimensional data than the linear ones, given the same compression ratio. Our previous work has shown that this belief analysis approach can further be extended by incorporating belief clustering so that an even more compact set of belief state bases can be identified [4].

2.3 Value-Directed Compression

Another interesting approach to address the dimensionality issue is called value-directed compression (VDC) [8] where a linear projections is used instead. VDC computes the minimal *Krylov* subspaces and thus the corresponding

reward and state transition functions so that the values governing the agent's decision making remain unchanged before and after the compression (thus called value-directed). To contrast with the belief compression approach, VDC does not perform any data analysis on belief space but computes a sub-space which is invariant to the compression projection matrix. As the projection is linear, the value function after the projection remains to be PWLC, and thus most of the existing algorithms for the policy solving can be adopted. Computing the *Krylov* sub-space is however time-consuming as a large number of linear programming problems are to be solved and yet a high compression ratio cannot be guaranteed. While a truncated *Krylov* iteration algorithm has been introduced in [8] for obtaining a forcibly compressed POMDP quickly stopping the *Krylov* iterations with loss due to the incomplete set of belief bases, there exists no mechanism for exploring the characteristics of the belief space for some particular domains as what the belief compression approach can provide. Also, the *Krylov* space analysis in VDC is applied to the complete problem. It is not straight forward to see how it can be combined with the belief clustering approach to be described in Section 3 and 4 for problem decomposition as what being suggested in this paper.

3 Value-Directed Compression With Belief Space Analysis

In this section, we propose a novel value-directed compression method that has belief state analysis incorporated as well. Recall that the goal of the value-directed compression is to keep the value (expected reward) of the belief states remain unchanged before and after converting the original problem to its compressed version. Given the new reward function to be $\tilde{R}$ and the new value function to be $\tilde{V}$, it has been proved in [8] that as long as we can find the proper reward function and transition functions with the base case $V_0^\pi(b) = R(b) = \tilde{R}(\tilde{b}) = \tilde{V}_0^\pi(\tilde{b})$, we could keep $V_{t+1}^\pi(b) = \tilde{V}_{t+1}^\pi(\tilde{b})$ hold throughout the whole horizon. The key question is how to find the proper reward function and transition functions in the low-dimensional belief space.

3.1 Belief Compression By Non-negative Matrix Factorization (NMF)

To explore the belief space's sparsity for compression and at the same time preserve the value function's PWLC property, we consider only linear projection techniques. In particular, we adopt the non-negative matrix factorization (NMF) [3] which guarantees all the elements of the reduced and the reconstructed belief state to be positive, and apply it to a belief state sample obtained via simulation to result in the reduced dimension belief space.

NMF is a technique to compute a *linear* and *non-negative* representation for approximating a given set of data. Given V to be an $M \times N$ matrix with each of its columns being an observation vector, one can approximate V using NMF so that $V \approx WH$, where W is an $M \times P$ matrix with its column forming a set of P (normally $< M$) non-negative basis components and the matrix H are the coefficients of the corresponding basis components. Intuitively, V is approximately represented by a weighted sum (H) of the basis components (W). Normally, W and H are derived using the standard updating rules in Eqs.(5) and (6), given as

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \quad (5)$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu W_{a\nu}}. \quad (6)$$

Eqs.(5) and (6) alternates until W and H converge. It has been shown that the updating rules are in effect minimizing a generalized Kullback-Leibler (KL) divergence between V and WH.

3.2 Incorporating NMF Into VDC

Let B denote a $n \times |S|$ matrix defined as $[b_1|b_2|...|b_n]^T$ where n is the number of belief states in the training sample, and $b_i(S_j) \geq 0$ is the j^{th} element of the belief state b_i. Also, denote F to be the $|S| \times l$ transformation matrix which factors B into the matrices F and $\widetilde{B}$ such that

$$B^T \approx F^T \widetilde{B}^T \quad (7)$$

where each row of B equals $b \approx b^r = \widetilde{b}F$ and the dimension of $\widetilde{B}$ is $n \times l$. As the main objective of deriving F is for dimension reduction, it is typical that $l << |S|$. To derive the reward function for the low-dimensional belief space after the compression, it was first observed that the expected reward of a belief state b in the original belief space is given as,

$$\begin{aligned} V(b) &= \sum_{s_i} b(s_i) \cdot R(s_i, a) \\ &= bR_{\cdot a} \\ &= \widetilde{b}FR_{\cdot a}. \end{aligned} \quad (8)$$

In the compressed belief space, the expected reward can be expressed as

$$V(\widetilde{b}) = \widetilde{b}\widetilde{R}_{\cdot a}. \quad (9)$$

Combining Eqs.(8) and (9) by allowing $V(b) = V(\tilde{b})$, it gives

$$\widetilde{R} = FR. \quad (10)$$

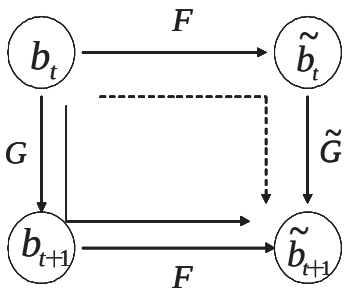


Figure 1. Two paths to reach the same next belief state in low-dimension.

To derive the low-dimensional state-transition function, let's consider two different paths for computing the next belief state in the high-dimensional space (shown as solid and dotted path in Figure 1). Given the current high-dimensional belief state, one path is to first apply the compression and then perform Bayesian updating in low-dimensional space. Another path is to first perform Bayesian updating in the high-dimensional belief space and then perform the belief compression. One can then obtain Eqs.(11) and (12), given as

$$\begin{aligned} \widetilde{b}^T_{t+1} &= \widetilde{SE}(\widetilde{b}_t, a, z) \\ &= \widetilde{G}^{<a,z>}\widetilde{b}^T_t \end{aligned} \quad (11)$$

$$\begin{aligned} b^T_{t+1} &= F^T\widetilde{b}^T_{t+1} \\ &= SE(b_t, a, z) \\ &= G^{<a,z>}b^T_t \\ &= G^{<a,z>}F^T\widetilde{b}^T_t. \end{aligned} \quad (12)$$

Equating Eqs.(11) and (12) using Eq.(7), we obtain

$$\begin{aligned} F^T\widetilde{G}^{<a,z>}\widetilde{b}^T_t &= G^{<a,z>}F^T\widetilde{b}^T_t \\ F^T\widetilde{G}^{<a,z>} &= G^{<a,z>}F^T. \end{aligned} \quad (13)$$

3.3 Analysis of Integrating Orthogonal NMF

So far, the equivalent low-dimensional POMDP problem has been established after the achievement of $\widetilde{R}$ and $\widetilde{G}$ (According to Eqs.(9) and (10)). Thus, the low-dimensional value function ($\widetilde{\alpha}$ vector) could be figured out by any traditional POMDP solving methods. Exploring the relationship between the original value function and the low-dimensional value function will help us directly project the low-dimensional α vectors back to the original space. Finally, the "true" policy can be achieved. The process of what we described so far would dramatically reduce the complexity of problem solving by computing policy in the low-dimensional space, if there is a guarantee that an accurate and compact low-dimensional "structure" could be probed. The following of this Section will describe how to efficiently find out the structure.

Recall that α vectors correspond to the value function over the belief space. Thus, the expected value for a belief b can be denoted as the inner product of an α vector and $b^\top$. The expected value for a belief b should be equal to that of the corresponding low-dimensional $\widetilde{b}$ based on the value directed idea. That is, equation

$$\alpha b^\top = \widetilde{\alpha}\widetilde{b}^\top \quad (14)$$

should hold (where α and b are row vectors). Since $b^\top = F^\top\widetilde{b}^\top$, we can substitute $F^\top\widetilde{b}^\top$ for $b^\top$ in Eq.(14) and derive the equations as below

$$\alpha b^\top = \alpha F^\top\widetilde{b} \quad (15)$$

$$\widetilde{\alpha} = \alpha F^\top. \quad (16)$$

Recall that F is a $k \times d$ projection matrix. Usually we assume $d > k$ to achieve dimension reduction. So when the rank of F is equal to that of $[F\widetilde{\alpha}]$, the above linear system is under determined and has infinitely many solutions. Naturally many linear programming techniques can be used to solve Eq.(16). In this paper, to get a unique solution of α vector and avoid solving large number of LP equations, we apply an additional constraint on F which is $F^\top F = I$. Thus, $\widetilde{\alpha}F = \alpha F^\top F$. and we then obtain $\alpha = \widetilde{\alpha}F$ which can be interpreted as the projection from the low dimension α vector to its counterpart in the original space.

In [2], Chris *et al.* introduced a specific orthogonal nonnegative matrix factorization which decomposed V into two nonnegative matrices with the constraint $W^\top W = I$. The objective function is :

$$\min_{W\geq 0, H\geq 0} \|V - WH\|^2, s.t. W^\top W = I. \quad (17)$$

To solve Eq.(17), Chris et al. [2] used the lagrangian multiplier method and derived lagrangian function to be minimized becomes

$$L = \|V - WH\|^2 + Tr[\lambda(W^\top W - I)]. \quad (18)$$

Where λ is the lagrangian multiplier. Given

$$\|V - WH\|^2 = Tr(V^\top V - 2W^\top VH^\top + HH^\top W^\top W), \quad (19)$$

the gradient of L becomes

$$\frac{\partial L}{\partial W} = -2VH^\top + 2WHH^\top + 2W\lambda. \quad (20)$$

The corresponding KKT complementary condition gives

$$(-2VH^\top + 2WHH^\top + 2W\lambda)_{ik}W_{ik} = 0 \quad (21)$$

(either $\frac{\partial L}{\partial W} = 0$ or $\lambda W = 0, \lambda > 0$). Since the above nonlinear system is difficult to solve, Christ et al,[2] proposed the following updating rules

$$W_{ik} \leftarrow W_{ik}\sqrt{\frac{(VM)_{ik}}{WW^\top VM}_{ik}} \quad (22)$$

$$M_{jk} \leftarrow M_{jk}\frac{(V^\top W)_{jk}}{(MW^\top W)_{jk}} \quad (23)$$

$$H = M^\top, \quad (24)$$

and prove that W and H would converge to a local minima of the problem with these updating rules, given any initialization.

While the orthogonal condition to be integrated in POMDP compression is

$$\min_{W\geq 0, H\geq 0} \|V - WH\|^2, s.t. WW^\top = I. \quad (25)$$

W is a $d \times k$ matrix ($d > k$). So for constraint $WW^\top = I$, the linear system is overdetermined. In general the overdetermined systems have no solution and need some techniques to find an approximate solution e.g, linear least squares. In the following description, we will show the updating rules Eqs.(22) and (23) are still reasonable approximations for solving Eq.(25). For Eq.(25), we can first derive the lagrangian function to minimize, which is given as

$$L' = \|V - WH\|^2 + Tr[\lambda(WW^\top - I)]. \quad (26)$$

The only difference between Eq.(18)and Eq.(26) is the second term, where $\frac{\partial Tr[\lambda(WW^\top - I)]}{\partial W} = \frac{\partial Tr[\lambda(W^\top W - I)]}{\partial W} = 2W\lambda$. Thus, the gradient of L' is the same as that of L and the respectively corresponding KKT complementarity conditions are the same as well (both are $(-2VH^\top + 2WHH^\top + 2W\lambda)_{ik}W_{ik} = 0$). Therefore, the updating rules Eq.(22,23) are also appropriate for solving Eq.(25).

Note that $\alpha b^\top = \widetilde{\alpha}\widetilde{b}^\top$ is expected to hold at any time for any belief state. Thus, $\alpha b_t^\top = \widetilde{\alpha}\widetilde{b}_t^\top$ holds at the time step t for belief b_t and state

$$\alpha b_{t+1}^\top = \widetilde{\alpha}\widetilde{b}_{t+1}^\top \quad (27)$$

should also hold at the time step $t+1$ where b_{t+1} is evolved from b_t. The evolvement of beliefs follows

$$b_{t+1} = T^{<a,o>} b_t^\top \quad (28)$$

and

$$\widetilde{b}_{t+1} = \widetilde{T}^{<a,o>}\widetilde{b}_t^\top. \quad (29)$$

Making a substitution of b_{t+1} and $\widetilde{b}_{t+1}$, Eq.(27) becomes

$$\alpha T^{<a,o>} b_t^\top = \widetilde{\alpha}\widetilde{T}^{<a,o>}\widetilde{b}_t^\top. \quad (30)$$

Substituting $\widetilde{\alpha}F$ and $F^\top\widetilde{b}_t$ for the left side α and $b^\top$ of above equation respectively, we get

$$\widetilde{\alpha}FT^{<a,o>}F^\top\widetilde{b}_t^\top = \widetilde{\alpha}\widetilde{T}^{<a,o>}\widetilde{b}_t^\top. \quad (31)$$

The updated projection of transition functions between the original space and the reduced space becomes

$$FT^{<a,o>}F^\top = \widetilde{T}^{<a,o>}. \quad (32)$$

Comparing with VDC proposed in [8], one advantage of our formulation is that there is no need to obtain $\tilde{R}$ and recover high-dimensional α vectors using the LP technique as adopted in [8]. What we propose is more efficient in computing the projection and the corresponding low-dimensional reward function and state transition functions as there is no need to solve a large number of linear programs.

In [2], another version of orthogonal NMF given as

$$\min_{W\geq 0, H\geq 0} \|V - WH\|^2 s.t. HH^\top = I. \quad (33)$$

was shown to be equivalent to K-means clustering (with column vectors of V as data) where $H^\top$ is the cluster indicator matrix for K-means clustering of columns of V. Taking the transpose of elements in Eq.(25), it is easy to see the version of orthogonal NMF is equivalent to K-means clustering with W is the cluster indicator matrix, but each column of V becomes the probabilities of falling on a particular true state at different time instances. Therefore, a possible future direction is to adopt this version of orthogonal NMF and then the NMF will aggregate d "true states" into k "abstract states" according to the sampled beliefs. This aggregation is unlike the one adopted in PolCA [7] where an action-based decomposition is used for the partitioning via analyzing and simplifying the state transition graph. Intuitively, this new version of orthogonal NMF based aggregating is more straightforward and automatic. It is interesting to see whether this could provide an alternative way to identify subgoals of POMDP in our future work.

4 Policy Computation and Application

4.1 Point Based Value Iteration

As mentioned before, orthogonal NMF involves linear transformations only and thus can preserve the PWLC properties, making the POMDP of reduced dimension suitable for any existing POMDP algorithms. In our experiments, Perseus is applied [10], which is an efficient randomized point-based approximate value iteration algorithm for computing the optimal policy. The combination is straightforward, except that Perseus requires a *backup* belief set used for reducing the number of α vectors to be stored. The belief set is created using the trajectory-based approach which is a common approach for belief state generation.

4.2 Sample Spacing Analysis in Low-Dimensional PBVI

In the literature, Pineau [6] did point based value iteration (PBVI) over a trajectory-based sample set simulating the belief simplex. And the sample size is continuously increasing as the iterative steps for computing value function are incremented. In [10], a trajectory-based sampling with a fixed sample size was used instead to cover the belief space. For these PBVI related algorithms, it has been proved that the error bound, denoted as $||V_t - V_t^B||_\infty$, is proportional to the sample spacing [6].

Let ϵ denote the error between the approximate value function derived based on the belief sample B and the original belief simplex $\triangle$. Also let δ_B denote $\max_{b' \in \triangle} \min_{b \in B} ||b - b'||_1$. Pineau *et al.* [6] showed the errorbound is

$$\epsilon \leq \frac{R_{max} - R_{min}}{1-\gamma} \delta_B. \tag{34}$$

Obviously, a good belief sample with small δ_B will reduce the value of ϵ and improve the accuracy of value function as $\frac{R_{max}-R_{min}}{1-\gamma}$ is constant. Therefore, the trade-off for the existing point based algorithms to play around with is how to improve the coverage of belief space via the belief sample and yet control the sample to be reasonably sized. For POMDP defined in the low-dimensional space, we have a similar error bound

$$\epsilon_{\widetilde{B}} \leq \frac{\widetilde{R}_{max} - \widetilde{R}_{min}}{1-\gamma} \delta_{\widetilde{B}} \tag{35}$$

where $\delta_{\widetilde{B}} = \max_{\widetilde{b}' \in \widetilde{\triangle}} \min_{\widetilde{b} \in \widetilde{B}} ||\widetilde{b} - \widetilde{b}'||_1$, and $\epsilon_{\widetilde{B}}$ corresponds to the error of the approximate value function. Since our proposed method is about solving the factored POMDP in the low-dimensional space and project these low-dimensional value function back to the high-dimensional space, the relationship between high-dimensional ϵ and low-dimensional spacing measure $\delta_{\widetilde{B}}$ can be shown as[1]

$$\begin{aligned}
\epsilon &\leq \alpha' \cdot b' - \alpha \cdot b' && (36)\\
&= (\alpha' - \alpha) \cdot (b' - b) && (37)\\
&= (\alpha' - \alpha) \cdot (F^\top \tilde{b}' - F^\top \tilde{b}) && (38)\\
&\leq \|\alpha' - \alpha\|_\infty \|F^\top\|_1 \|\tilde{b}' - \tilde{b}\|_1 && (39)\\
&\leq \frac{R_{max} - R_{min}}{1-\gamma} \|F^\top\|_1 \delta_{\widetilde{B}}, && (40)
\end{aligned}$$

[1]where α' is the best value function for b' and α is the best one for b. According to Eq.(40), choosing the proper F with minor $\|F^\top\|_1$ is another consideration to improve the policy performance in addition to the orthogonality of F and recovery accuracy of compressed data.

4.3 Towards Reducing Low Dimensional Sample Spacing

In this section, we propose an oversampling method to reduce $\delta_{\widetilde{B}}$ thus reduce the value of ϵ and $\epsilon_{\widetilde{B}}$ according to Eq.(35) and (40). Note that the belief space of POMDP with $|S|$ states is a $|S|-1$ dimensional belief simplex, and a convex hull in the $|S|-1$ dimensional Euclidean space as well. Naturally, uniformly picking the points to explore the reachable beliefs in the low dimension space will likely decrease $\epsilon_{\widetilde{B}}$. Given N beliefs $b_1, b_2 ... b_N$, the convex hull $C \equiv \{\sum_{j=1}^{N} \lambda_j b_j : \lambda_j \geq 0, \sum_{j=1}^{N} \lambda_j = 1\}$, Since orthogonal NMF is a linear transformation, the convex hull for the low-dimensional beliefs is the linear projection from C to $C \circ f$, and $C \circ f \equiv \{\sum_{j=1}^{N} \lambda_j (b_j \circ f) : \lambda_j \geq 0, \sum_{j=1}^{N} \lambda_j = 1\}$. Obviously, the points lying in the lines bridged by any two low-dimensional beliefs are in the projected convex hull which also belongs to the low-dimensional belief space. Randomly collecting those points to supplement the low-dimensional belief sample set would likely reduce $\delta_{\widetilde{B}}$. In this paper, an oversampling algorithm (See Algorithm 1) is adopted to generate additional low-dimensional belief sample points to make a better coverage of the low-dimensional space.

Algorithm 1 *OVERSAMPLE*(S,N,k)

```
Input: The transformed low-dimensional belief set S, the number of new points N, the number of picking points in a line k
Output: An over sampled low-dimensional belief set
Compute and sort distance between points in S
Record ⌈N/k⌉ pairs of index points with larger distance
while N>0 do
    Pop a recorded pair of index(c,s)
    for l← 1 to k do
        Compute: gap=rand
        for j← 1 to dim do
            Compute: dif=S[c][j]-S[s][j]
            NewSample[j]=S[s][j]+gap*dif
        end for
        N=N-1
        Absorb NewSample into S
    end for
end while
return
```

5 Performance Evaluation

To evaluate the effectiveness of the proposed value-directed compression method, the hallway and hallway2

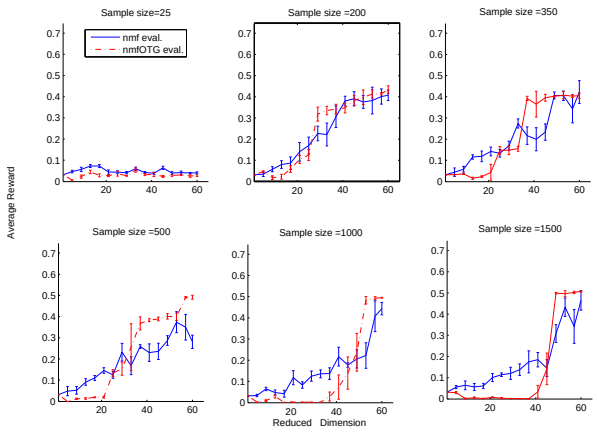


Figure 2. The comparison between NMF compression and orthogonal NMF compression on hallway problem with different sample size

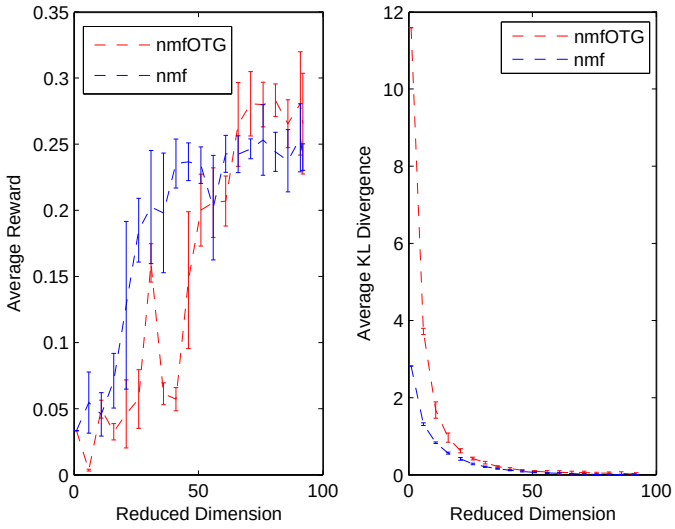


Figure 3. The comparison between NMF compression and orthogonal NMF compression on hallway2 problem)

problem [1] were tested with limited solving time. Figure 2 shows the comparison between NMF compression and orthogonal NMF compression over different sample sizes for hallway problem. Obviously the latter one gave policies with reasonably better quality, resulting in higher average reward values in simulations over all sample sizes. The upper left subfigure shows the example with 25 belief sample points and the result was found to be the worst as expected. Whereas a large sample set does not guarantee a good result either if a fixed solving time is given, there might be exists too redundant beliefs which only sacrifice the computation time instead of contributing to the computation of value function. In our experiments, an empirical sample size is 500 (bottom left of Figure 2). Regarding the decomposition accuracy shown in Figure 4, when the dimensionality change from 1 to 30 dimension, the performance of computed policy based on orthogonal NMF is worse than that of NMF as the accuracy of orthogonal NMF was lower than that of NMF. From 40 onward, both orthogonal NMF and NMF achieve the accurate factorization, orthogonal NMF got the results with higher average reward than NMF as orthogonal NMF achieved more accurate recovery of value function in high-dimensional space (See Section 3.3). Figure 3 shows the policy performance of hallway2 problem computed using NMF and orthogonal NMF. And we observed the similar scenario as Figure 2. Orthogonal NMF based compression made better policy than NMF version once the accuracy reconstructions of NMF and orthogonal NMF are close enough beyond 60 dimension. As shown in Figure 3, orthogonal NMF (nmfOTG) dominates the typical NMF approach only after the dimension size is over 60. However, as shown in the lower left subfigure of Figure 2, it achieves a clear dominance over the typical NMF only after the dimesion size reaches 30.

That is because $W^\top W$ is a more likely approximation to $WW^\top$ as the state size 60 is smaller than 90. Figure Figure 5 shows the policy performance computed over random sample set and sample set generated using our proposed over sampling Algorithm 1. The latter sample set is generated as the following description: Firstly, we randomly sampled 1000 beliefs in the original belief space. Then we reduced the precision of these beliefs with $\hat{S} = \lceil S * 1000 + 0.5 \rceil / 1000$ (S is the belief sample matrix and with row as belief). After filtering those unique rows in $\hat{S}$, we get a compact set $\grave{S}$ with 643 beliefs. For each specified dimension reduction using orthogonal nmf over $\grave{S}$, we call Algorithm 1 on to over sample the low-dimensional belief set $\grave{S}$ and makeup the size $\grave{S}$ to 1000, with parameters $N = 357, k = 3$. Figure 5 shows the over sampling technique is effective for improving low-dimensional policy performance, especially better for those much lower dimensions than computing policy over the samples generated by random trajectory based sampling method. However this technique does not make an obvious effectiveness over hallway problem as the random trajectory based sampling has achieved a good simulation for the belief simplex of hallway problem. So we tend to use this oversampling technique to the larger POMDP problem to generate an effective sample set.

6 Conclusion

This paper illustrates the motivation and the details of a novel value-directed compression method based on orthog-

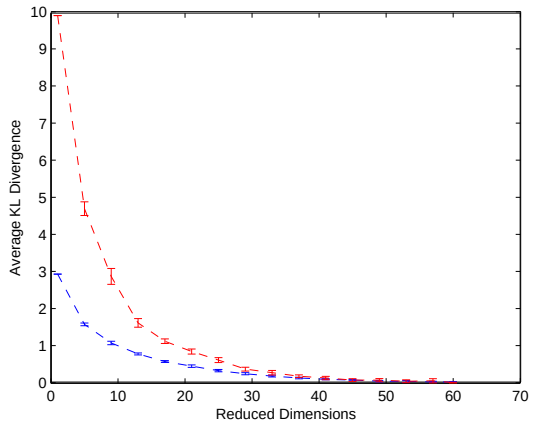


Figure 4. Comparison of data recovery between NMF compression and orthogonal NMF compression on hallway problem with sample size 500.

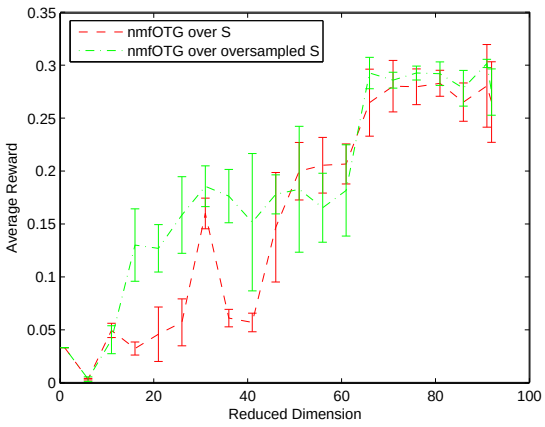


Figure 5. Comparison of orthogonal NMF compression over random sampled 1000 beliefs and an oversampled sample set for hallway2 problem

onal NMF which explores also the belief space sparsity thus computational complexity reduction. The proposed method has been evaluated using navigation related problems with positive results.

While the experimental results obtained so far are positive, there still exist a number of possible extensions to further improve the proposed method. An approximate updating rule to solve the orthogonal NMF used in this paper has shown to be effective, a more accurate solution for orthogonal NMF will be promising to enhance the policy performance. We believe that this is an immediate and important extension of this work to be pursued in the future. The state abstraction induced by orthogonal NMF is also an interesting and promising issue to be addressed in our future work to discovery the subgoals of POMDP problem and further reduce the complexity of problem solving for large-scale problems. Lastly, it is also interesting to see how the proposed method can be extended to support online learning (e.g., Q-learning [11]) of POMDP under the partial observation scenario.

References

[1] A.Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. U.Brown, 1998.

[2] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Philadelphia, PA, USA*, pages 126–135, 2006.

[3] D. D.Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.

[4] X. Li, W. K. Cheung, and J. Liu. Decomposing large scale pomdp via belief state analysis. In *Proceedings of 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, Compiegne, France, 2005.

[5] N. Roy, G. Gordon and S. Thrun. Finding approximate POMDP solutions through belief compressions. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

[6] G. G. Pineau, J. and S. Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.

[7] J. Pineau and S. Thrun. An integrated approach to hierarchy and abstraction for pomdps. Technical Report CMU-RI-TR-02-21, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.

[8] P. Poupart and C. Boutilier. Value-directed compression of POMDPs. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, 2003.

[9] E. J. Sondik. *The Optimal Control of Partially Observable Markov Decision Processes*. PhD thesis, Stanford University, Stanford, California, 1971.

[10] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[11] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge Univ., Cambridge England, 1989.

Maximum Entropy in Regularized Classifier Ensemble

Zhi-li Wu

Abstract

In linearly combining classifiers, majority vote can be understood to follow the maximum entropy principle, which treats all base classifiers equally, despite their performance difference. However, there are some true target labels of training examples available for training, the label prediction loss constitutes a criterion for classifier combination too. We in this paper propose a regularized classifier combination strategy, which aims to maximize the entropy of different probabilities assigned to base classifiers, and also in a same objective function tries to ensure a large enough margin to separate different classes of data points.

1 Introduction

Classifier combination or ensemble [6, 8, 3] usually refers to the technique of combining outputs of multiple classifiers (they are called base classifiers), so as to obtain a more accurate and robust classifier.

Diversified and accurate base classifiers play a key role in constructing a successful classifier ensemble [4], while a less touched issue in ensemble research is the combination strategy. Among some existing combination strategies, majority vote as mostly seen in Bagging [1] is the the simplest but most pervasive one; Weighted combination is adopted in Adaboosting [5] and some neural network ensemble methods [11, 9]; While more advanced stacking generalization methods [10] are to learn a meta classifier upon the outputs of base classifiers; Hierarchical combination such as hierarchical mixture of experts can be found [7] too.

Some studies shows that majority vote though simple has generally good performance. This property make it particularly favorable to ensemble practitioners. We in this paper try to understand majority vote from a maximum entropy view. This view reveals that under the situation of no any bias to any particular base classifier, majority vote acts as the safest way in combining base classifiers, as measured by the maximum entropy of the probabilities assigned to base classifiers.

On the other hand, the typical majority vote based ensemble — Bagging, as compared with other advanced ensemble approaches, — is believed to be less competitive. We in this paper further introduce the margin concept, which shows that in majority vote a large enough margin cannot be guaranteed, hereby may cause performance decrease. We can utilize the training data and true labels to enforce a margin to separate different classes of points. This can be regarded as incorporating label prediction loss into maximum entropy criterion, so as to achieve a balance between the maximum entropy of probabilities assigned to base classifiers, and the generalization performance of the classifier ensemble.

This paper in essence conducts research on improving majority vote in a regularization framework. Its content is organized as follows. In section 2 a maximum entropy interpretation is given to majority vote, accompanied by the introduction of margin concept in constructing classifier ensemble. Section 3 presents a novel objective function that considers both the maximum entropy of probabilities to base classifiers and the margin to separate two classes, and then mathematical deduction is presented for transforming this objective function. Section 4 further refines this formulation, to make it applicable to the case of allowing some data points separated by a smaller margin, but with a penalty term added. Section 5 presents a numerical method to solve the objective function, and some discussions on parameter selection are given too. Section 6 demonstrates the performance of our method, as compared with bagging, and genetic algorithm based ensemble methods. In the ending section, some discussion and further research issues are identified.

2 Majority Vote From a Maximum Entropy and Margin View

Suppose a set of d base classifiers $\{f_i\}_{i=1}^d$ have been trained upon data samples bootstrapped from a

Table 1. Three Base Classifiers

	Base Classifiers' Outputs				Voting	
$(\mathbf{x}_i, y_i)$	f_1	f_2	f_3	f_4	f_1, f_2, f_3 Vote (F Value)	All Vote (F Value)
$(\mathbf{x}_1, +1)$	+1	-1	+1	-1	+1 (1/3)	±1 tie (0)
$(\mathbf{x}_2, -1)$	-1	-1	+1	-1	-1 (-1/3)	-1 (-1/2)
$(\mathbf{x}_3, +1)$	+1	-1	+1	+1	+1 (1/3)	+1 (1/2)
$(\mathbf{x}_4, -1)$	+1	+1	-1	-1	+1 (-1/3)	±1 tie (0)

dataset, while for simplicity we assume the outputs of these base classifiers are ±1 only [1].

As shown in Table 1, four base classifiers (f_1, f_2, f_3, f_4) are trained, and their outputs for a set of four points $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ are given in corresponding columns, while the true labels y_i for the four points are given in the left column. For this task, a simple majority vote by the first three base classifiers can predict three of the four labels perfectly, while the majority vote of all four classifiers can only correctly predict two, and leaves two undetermined due to the ties of votes.

To understand why the number of base classifiers affects ensemble performance, and how they affect the performance, it is useful to regard the majority vote as a weighted combination of base classifiers, followed by a *sign* operation.

$$F(\mathbf{x}) = \sum_{i=1}^{d} w_i f_i(\mathbf{x}).$$

For instance, the majority vote of the first three classifiers (f_1, f_2, f_3) can be described by the following form, with $w_1 = w_2 = w_3 = \frac{1}{3}$:

$$F(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + w_3 f_3(\mathbf{x}),$$

Since the weights $\mathbf{w}$ forms a probability distribution, that is, $\sum_{i=1}^{d} w_i = 1, w_i \geq 0$, these uniform weights in majority vote in fact satisfy the maximum entropy criterion [2]:

$$\max: \quad H(\mathbf{w}) = -\sum_{i=1}^{d} w_i \ln w_i.$$

More generally all $w_i = \frac{1}{d}$ will always maximize the entropy $H(\mathbf{w})$ (See Fig. 1). It should be pointed out that uniform weights in fact maximize a family of criteria, such as the negation of the l_2 norm of $\mathbf{w}$ as given by $-||\mathbf{w}||^2 = \sum_{i=1}^{d} w_i^2$, but the entropy criterion has a natural physical meaning. It indicates that when there is no bias to any particular base classifier and treat all of them with equal probabilities, the combination result has the largest entropy and smallest risk.

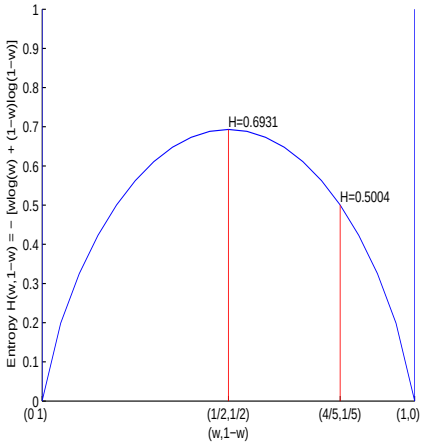


Figure 1. Entropy Criterion: uniform weights $(\frac{1}{2}, \frac{1}{2})$ achieve maximum entropy

Average combination adopted by majority vote maximizes the entropy of probability weights to base classifiers, but does not necessarily minimize the prediction error, as shown in the toy example in Table 1. In addition, Table 1 lists the weighted combination value F in the last two columns, as enclosed by brackets. It can be verified that for the combination of the first three base classifiers, as given by $F(\mathbf{x}) = \sum_{i=1}^{3} w_i f_i(\mathbf{x})$, we have the following except for $\mathbf{x}_4$:

$$F(\mathbf{x}_j) \geq \frac{1}{3}, \; y_j = +1,$$

and

$$F(\mathbf{x}_j) \leq -\frac{1}{3}, \; y_j = -1,$$

So this implies that the weighted values for positive and negative points are separated by a margin of $\frac{2}{3}$, if excluding $\mathbf{x}_4$. To correct the prediction to $\mathbf{x}_4$, at least one positive label output from base classifiers should

[1] Continuous or probability outputs, and multiclass case shall fit for the same analysis too.

[2] ln() rather than $\log_2()$ is used here for simplicity, which has no effect on conclusions presented in this paper.

be flipped, such that the condition $F(\mathbf{x}_4) \leq -\frac{1}{3}$ holds true. Generally speaking, to correctly predict all labels based on the majority vote of an odd number of base classifiers, a positive point should be associated with a $F \geq \frac{1}{d}$, and a negative point should be with a $F \leq -\frac{1}{d}$, and then all positive and negative points are separated by a margin no less than $\frac{2}{d}$.

For the combination of all four base learners in Table 1, we only have $F(\mathbf{x}_i) \leq 0$ for negative points and $F(\mathbf{x}_i) \geq 0$ for positive points. Hereby some positive and negative points are not separated at all. It can be analytically figured out that in order to perfectly separate two classes of points by the majority vote of an even number of base classifiers, the condition of $F(\mathbf{x}_i) \geq \frac{2}{d}$ for positive points and $F(\mathbf{x}_i) \leq -\frac{2}{d}$ for negative points should be satisfied. Hereby a margin of $\frac{4}{d}$ should be guaranteed. But in many real cases this margin criterion is even harder to achieve than combining a close but odd number of base classifiers, just like the toy example shown in Table 1.

Hereby, we can know the simple uniform weights to maximize the entropy criteria cannot guarantee a large enough margin to separate two classes of training points. The remedy way we adopt is to allow nonuniform weights, which although will cause the entropy value reduced a little, however could guarantee a lower bound for the margin in following way:

$$\sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \geq \gamma, y_j == +1$$

$$\sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \leq -\gamma, y_j == -1$$

$$\gamma \geq 0, \forall j = 1, \ldots, n,$$

where n is number of training points. Hereby these constraints ensure a margin equal to 2γ. In addition, this set of criteria are equivalent to

$$y_j \sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \geq \gamma, \forall j = 1, \ldots, n. \quad (1)$$

3 Objective Function and Deduction

Now we can summarize our objective function as follows: we are seeking for the probability weights to maximize the entropy $H(\mathbf{w})$:

$$\max: \quad H(\mathbf{w}) = -\sum_{i=1}^{d} w_i \ln w_i,$$

with respect to the constraints (Eq. 1) and

$$\sum_{i=1}^{d} w_i = 1, w_i \geq 0, \forall i = 1, \ldots, d.$$

To solve this objective function, it is useful to convert it into a dual form by introducing Lagrange multipliers $\alpha_j, \alpha_j \geq 0, j = 1, \ldots, n$ and unconstrained β. We then get

$$\begin{aligned} L &= -\sum_{i=1}^{d} w_i \ln w_i + \sum_{j=1}^{n} \alpha_j y_j \sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \\ &\quad -\gamma \sum_{j=1}^{n} \alpha_j - \beta(\sum_{i=1}^{d} w_i - 1) \end{aligned}$$

Maximizing L with respect to w_i is equivalent to minimizing L when all w_i vanishes, so let

$$\frac{\partial L}{\partial w_i} = 0, \forall i = 1, \ldots, d.$$

These result in

$$-\ln w_i = 1 + \beta - \sum_{j=1}^{n} \alpha_j y_j f_i(\mathbf{x}_j), \forall i = 1, \ldots, d.$$

Utilizing the unit summation of all w_i, we can get the closed- form solution to w_i as:

$$w_i = \frac{e^{\sum_{j=1}^{n} \alpha_j y_j f_i(\mathbf{x}_j)}}{\sum_{i=1}^{d} e^{\sum_{j=1}^{n} \alpha_j y_j f_i(\mathbf{x}_j)}}$$

The dual objective function to be minimized is given by

$$L_d = 1 + \beta - \gamma \sum_{j=1}^{n} \alpha_j$$

which can be further written into

$$L_d = \ln(\sum_{i=1}^{d} e^{\sum_{j=1}^{n} \alpha_j y_j f_i(\mathbf{x}_j)}) - \gamma \sum_{j=1}^{n} \alpha_j$$

Before introducing a numerical method for minimizing this dual objective function, we first in next section relax the constraints in Eq. 1 a little, to allow some data points separated by a smaller margin.

4 Margin Relaxation

If for some $\mathbf{x}_j$ the constraint in Eq. 1 cannot be fulfilled, (e.g. in an extreme case, the y_j is +1 but all

$f_i(\mathbf{x}_j)$ is -1), we in this case should tolerate the decrease of margin for such hard points, so we introduce the following criteria instead:

$$\sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \geq \gamma - \xi_j, y_j == +1$$

$$\sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \leq -\gamma + \xi_j, y_j == -1$$

$$\gamma, \xi_j \geq 0, , \forall j = 1, \ldots, n$$

Similarly, these relaxed constraints can be rewritten into

$$y_j \sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) \geq \gamma - \xi_j, \forall j = 1, \ldots, n$$

In this case, we can modify the objective function to maximize entropy, and also to minimize the total summation of penalty terms ξ_j, balanced by a user specified parameter C, $(C \geq 0)$:

$$\max : \quad -\sum_{i=1}^{d} w_i \ln w_i - C \sum_{j=1}^{n} \xi_j$$

Through deduction, we can get the exact same dual form, but with a upper bound C to all Lagrange multipliers.

$$L_d = \ln(\sum_{i=1}^{d} e^{\sum_{j=1}^{n} \alpha_j y_j f_i(\mathbf{x}_j)}) - \gamma \sum_{j=1}^{n} \alpha_j$$

$$0 \leq \alpha_j \leq C, \forall j = 1, \ldots, n$$

5 Deepest Gradient Descent Method

We in this section present a deepest gradient descent method to minimize the dual objective function obtained before.

First let $y_j f_i(\mathbf{x}_j) = z_{ji}$, we are aiming to solve

$$\min : \quad L_d = \ln(\sum_{i=1}^{d} e^{\sum_{j=1}^{n} \alpha_j z_{ji}}) - \gamma \sum_{j=1}^{n} \alpha_j$$

w.r.t.

$$0 \leq \alpha_j \leq C$$

Taking the partial derivative of α_j :

$$\nabla_j = \frac{\partial L_d}{\partial \alpha_j} = \frac{\sum_{i=1}^{d} z_{ji} e^{\sum_{j=1}^{n} z_{ji}}}{\sum_{i=1}^{d} e^{\sum_{j=1}^{n} z_{ji}}} - \gamma.$$

Hereby we can summarize the deepest gradient descent method as follows:

- Start from all $\alpha_j = 0$, and choose a step length S and a termination value ϵ.
- Calculate the derivative vector $\boldsymbol{\nabla} = \{\nabla_1, \ldots, \nabla_j, \ldots, \nabla_n\}$.
- Update $\alpha_j^{new} = \alpha_j^{old} - S\nabla_j$, which might be subject to truncation due to the bound condition $0 \leq \alpha_j \leq C$. By taking the truncation into account, the update is actually given by $\alpha_j^{new} = \alpha_j^{old} - S\delta_j$, where $\delta_j = \max(\frac{\alpha_j^{old} - C}{S}, \min(\nabla_j, \frac{\alpha_j^{old}}{S}))$.
- If the norm of the vector formed by all δ_j is larger than the termination value ϵ, go to step 2. Otherwise, calculate the solution to w_i according to all α_j, and then terminate the algorithm.

In our formulation, γ and C are two parameters to be specified beforehand. In the following we provide some analysis on their properties, so as to derive some heuristic rules for parameter selection.

Property 1. $\gamma \leq 1$: Since in this classifier combination setting we have $|\sum_{i=1}^{d} w_i f_i(\mathbf{x}_j)| \leq 1$, it implies that the maximum γ can be 1. Set a γ that is larger than 1 will always introduce non-zero penalty terms ξ into the objective function. Hereby γ is suggested to be a value no larger than 1.

Property 2. $\gamma \geq v = \min_{j=1}^{n} \frac{\sum_{i=1}^{d} z_{ji}}{d}$: The majority vote is regarded as an average combination before, with each $w_i = 1/d$, which achieves the maximum entropy among all possible value sets of w_i. In this average combination case, the combined output for each $\mathbf{x}_j$ is given by $u_j = \frac{1}{d} \sum_{i=1}^{d} f_i(\mathbf{x}_j)$. And let the minimum value be $v = \min_{j=1}^{n} y_j u_j$. If γ is set to be a value smaller than or equal to this value v, it can be easily known average combination is always the solution. Hereby γ is suggested to be a value no smaller than v.

To determine a proper (range of) C, it is necessary to study the summation of penalty terms $\sum_{j=1}^{n} \xi_j$, since C is multiplied on it.

In fact, at the optimal condition, each $\xi_j = y_j \sum_{i=1}^{d} w_i f_i(\mathbf{x}_j) + \gamma$, hereby $\xi_j \leq 1 + \gamma$.

Provided that the entropy term will be in the range $[0, \ln(d)]$, to make the penalty term roughly in the same scale of the entropy term, we might suggest the $C = \frac{\ln(d)}{n(1+\gamma)}$. However, if the margin is expected to be larger, a larger C can be used.

6 Experiment

6.1 Synthetical Example

We test an artificial task of combining the outputs of five base classifiers, so as to demonstrate how our algo-

Table 2. Synthetical Task by Five Base Classifiers

	Base Classifiers' Outputs				
y_i	f_1	f_2	f_3	f_4	f_5
-1	-1	1	1	1	-1
1	1	1	-1	1	-1
1	1	-1	1	1	-1
-1	-1	-1	1	-1	1
-1	-1	1	-1	-1	-1
-1	-1	1	1	1	-1
1	-1	1	1	-1	1
1	1	-1	-1	1	1
-1	1	-1	1	-1	-1
-1	-1	-1	-1	-1	-1
1	-1	1	1	-1	1
1	1	1	-1	1	1
1	1	-1	-1	-1	1
-1	-1	-1	1	-1	1
-1	-1	-1	-1	1	1
1	1	1	1	-1	1
1	1	1	1	1	1
-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1
1	1	1	1	1	1

rithm perform. Assume the real target can be obtained from an optimal weighted combination, as given by $y_i = sign(0.2465f_1 + 0.2192f_2 + 0.0895f_3 + 0.1877f_4 + 0.2570f_5)$. And 20 sets of outputs are listed in Table 2, while the first ten are used for training our algorithm, and the remaining ten for testing.

For this task, the majority vote will predict the label incorrectly for two training points and one for testing point. Figure 2 shows some results of our algorithm. It can be noticed from the graph the objective function value reduces quickly, and after 10 iterations, both the training and testing error drop to zero. And many α coefficients are zero since the beginning of training, which shows that to determine weights, only the label outputs for a few training points are essential. And the final weights obtained are $(0.2724, 0.1517, 0.1517, 0.1517, 0.2724)$.

6.2 Real-Task

We compared this entropy based combination strategy with bagging, and a weighted combination method optimized by genetic algorithm [11]. And also, among all the base classifiers, the single base classifier achieving the best testing accuracy is recorded, as well as the mean accuracy of all base classifiers.

We run 5 trials of 2-fold cross validation, while in each fold of training, a validation set is bootstrapped from the training set for parameter selection. The base classifiers we use are neural networks, or the decision trees which follow the random forest setting [2] to split a subset features only at each node.

Table 3,4 show the results by combining an odd number of base learners, and Table 5,6 list results from combining an even number of base learners. All tables shows the trend that our Maximum entropy method (MaxEnt) has advantages over other methods like majority vote, or the genetic algorithm based weighted combination (GA-Weight). Nearly in most cases, the MaxEnt approach perform better than the majority vote, which verifies that majority vote can be improved by taking the classification accuracy into account, through a regularization way like our MaxEnt doess. The column of results entitled "Mean" denote the average of the accuracy values of all separate base classifier upon the testing set, while the "Single Best" column denotes the base classifier that has the highest accuracy among all base classifiers trained on the testing set. It can be noticed that the ensemble approaches like majority vote, GA-weight and MaxEnt usually outperform the single best base classifier, and also show a large performance gain compared with the mean accuracy of all single base classifiers.

7 Conclusion

This paper presents a method to combine base classifiers, which are featured with entropy maximization and bounded margin. Entropy maximization ensures that probability weights to base learners are spread as uniformly as possible, and the margin bound can ensure the ensemble outputs for two classes of points are separated by a pre-specified value. We propose a novel objective function to achieve these two targets simultaneously, the numerical method we adopt can efficiently solve this objective function, thus makes a new way of base classifier combination possible. Experiments shows that our new method outperforms majority vote and other weight approximation methods.

As for future study, the convergence of the deepest gradient descent method and the sensitivity analysis for parameters should be investigated. Other than the application to classifier combination, the adaptation of this objective functions to other learning scenarios, like ordinary classification cases in which the data features play the same role as the outputs of base classifiers.

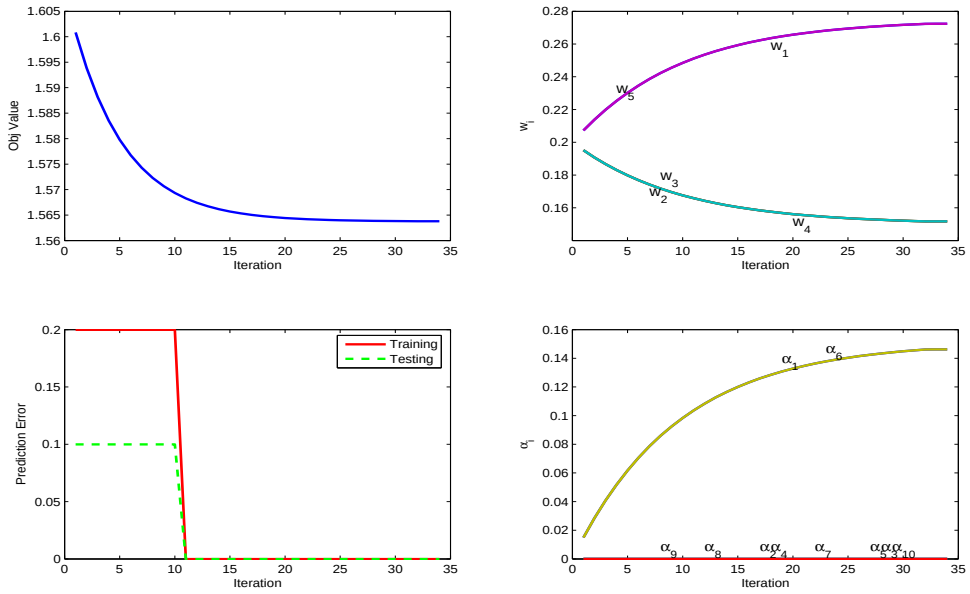


Figure 2. Synthetical Example: $\gamma = 0.1, C = 0.1463$

Table 3. CMC Data, 1473 points, 9 features, 3 classes

# of trees	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
3	0.7215	0.7361	**0.8029**	0.7464	0.8018
5	0.7260	0.7461	0.8416	0.8241	**0.8442**
7	0.7221	0.7543	0.8713	0.8592	**0.8756**
9	0.7221	0.7484	0.8793	0.8620	**0.8816**
11	0.7188	0.7494	0.8801	0.8770	**0.8865**
# of neural networks	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
3	0.5410	0.5589	0.5561	**0.5597**	0.5588
5	0.5531	0.5758	0.5741	0.5800	**0.5815**
7	0.5511	0.5801	0.5804	0.5854	**0.5871**
9	0.5529	0.5842	0.5883	0.5905	**0.5989**
11	0.5503	0.5845	0.5822	0.5804	**0.5905**

Table 4. Vehicle Data, 846 points, 18 features, 4 classes

# of trees	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
5	0.8608	0.8803	0.9630	0.9498	**0.9659**
11	0.8667	0.8893	0.9936	0.9910	**0.9948**
17	0.8641	0.8924	0.9974	0.9960	**0.9979**
23	0.8642	0.8924	0.9995	0.9995	**0.9998**
29	0.8640	0.9043	**1.0000**	0.9986	**1.0000**
# of neural networks	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
5	0.8156	0.8481	0.8652	0.8652	**0.8709**
11	0.8090	0.8564	0.8735	**0.8780**	0.8770
17	0.8098	0.8590	0.8770	0.8789	**0.8820**
23	0.8116	0.8659	0.8758	0.8827	**0.8829**
29	0.8115	0.8640	0.8815	**0.8915**	0.8851

Table 5. German Data, 1000 points (+:- = 7:3), 24 features, 2 classes

# of trees	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
4	0.8426	0.8630	0.9168	**0.9252**	0.9250
6	0.8435	0.8634	0.9470	0.9434	**0.9528**
8	0.8455	0.8712	0.9690	0.9668	**0.9770**
10	0.8441	0.8774	0.9742	0.9742	**0.9794**
# of neural networks	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
4	0.7710	0.7974	0.7898	0.8092	**0.8108**
6	0.7696	0.8028	0.7950	0.8154	**0.8158**
8	0.7672	0.8022	0.7910	0.8044	**0.8100**
10	0.7661	0.8060	0.8026	0.8148	**0.8190**

Table 6. DNA Data, 2000 points, 180 features, 3 classes

# of trees	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
10	0.8144	0.8474	0.9930	0.9890	**0.9939**
20	0.8162	0.8648	**0.9996**	0.9994	**0.9996**
30	0.8131	0.8644	0.9998	0.9997	**0.9999**
40	0.8164	0.8686	**1.0000**	**1.0000**	**1.0000**
50	0.8147	0.8659	**1.0000**	0.9992	**1.0000**
# of neural networks	Mean	Singe Best	Majority Vote	GA-Weight	MaxEnt
10	0.8339	0.9348	0.9729	0.9758	**0.9784**
20	0.8526	0.9445	0.9847	0.9848	**0.9873**
30	0.8381	0.9336	0.9866	0.9882	**0.9887**
40	0.8524	0.9456	0.9912	0.9922	**0.9928**
50	0.8447	0.9488	0.9901	0.9896	**0.9918**

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] T. G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[4] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[6] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001, 1990.

[7] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. Technical Report AIM-1440, 1993.

[8] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[9] N. Ueda. Optimal linear combination of neural networks for improving classification performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(2):207–215, 2000.

[10] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[11] Z.-H. Zhou, J.-X. Wu, Y. Jiang, and S.-F. Chen. Genetic algorithm based selective neural network ensemble. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, volume 2, 2001.

Lightweight Piggybacking for Packet Loss Recovery in Internet Telephony

Wing Yan Chow

***Abstract*—We consider an Internet telephony system in which the service provider operates a telephone gateway in each servicing city to serve the general public. We propose a packet loss recovery system, called *lightweight piggybacking*, for this system. This scheme applies two stages of erasure coding and fragmentation, such that only a small redundancy is piggybacked to each voice packet while this redundancy can be shared by multiple voice streams to a large extent for effective packet loss recovery. Compared with the conventional piggybacking scheme, the lightweight piggybacking scheme can effectively: (i) increase the probability of recovering the lost packets using the same or smaller amount of redundancy, and (ii) recover the loss of multiple and consecutive packets.**

I. Introduction

Internet telephony has shown a substantial growth in recent years because of low service charge for long-distance calls and value-added functions [1]-[2]. When a service provider wants to provide Internet telephony to the general public, it can operate a telephone gateway in each servicing city to bridge the local telephone network and the Internet [1]. In this manner, the general public can use telephones to access the telephone gateway for long-distance calls through the Internet. Fig. 1 shows this configuration.

Each gateway collects voice streams from its servicing city, packetizes them and transmits the resulting packets to the destination gateway through the Internet. In transmission, a voice packet may be lost or erroneous (if a voice packet is received after its playout time because of delay, it is equivalent to being lost). The lost packet cannot be retransmitted because of the stringent delay requirement for real-time and long-distance telephone conversation. As a result, the voice quality is affected.

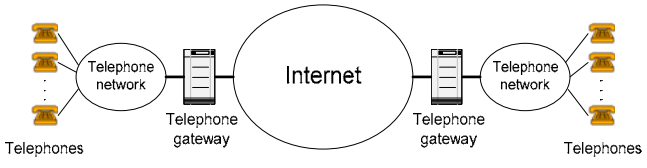


Figure 1. An Internet telephony system based on telephones to telephones.

To tackle the packet loss problem, one major approach is to perform *packet loss recovery* [3]. In this approach, the source adds redundant information so that the destination may be able to use this redundant information to recover the lost packet. By controlling the amount of redundancy added, the probability of recovering the lost packets can be significantly increased and hence the voice quality can be significantly improved [3]-[9]. In the literature, there are several packet loss recovery methods:

(1) *Parity coding* [3]-[4]. This method performs bitwise XOR operation on n voice packets to produce one parity packet. When any one of these n+1 packets is lost, the destination can recover the lost packet by performing a similar bitwise XOR operation on the n received packet.

(2) *Erasure coding* [10]. Erasure coding is more powerful than parity coding, and several methods apply erasure coding for packet loss recovery [11]-[13]. In general, erasure coding is applied on k voice packets to produce n–k redundant packets. When the destination can receive at least any k of these n packets, it can recover the original k voice packets.

(3) *Piggybacking* [6]. Piggybacking is well-known and extensively used for real-world applications (e.g., it is adopted in Free Phone [14] and Robust Audio Tools [15]). For voice packet i, the source produces a further-compressed packet by discarding the less-important bits and attaches this small and redundant packet to voice packet i+1. When voice packet i is lost, the destination extracts the redundant one from voice packet i+1 so that it can recover the further compressed version of voice packet i. Piggybacking only requires a small redundancy per voice packet, but it cannot recover the loss of consecutive packets.

In this paper, we design a new packet loss recovery scheme, called *lightweight piggybacking*, for the Internet telephony system shown in Fig. 1. The proposed scheme applies two stages of erasure coding and fragmentation, such that only a small redundancy is piggybacked to each voice packet while this redundancy can be shared by multiple voice streams to a large extent for effective packet loss recovery. As a result, the proposed scheme can effectively: (i) increase the probability of recovering the lost packets using the same or smaller amount of redundancy, and (ii) recover the loss of multiple and consecutive packets. We conduct computer simulation to demonstrate that lightweight piggybacking has significantly better performance than piggybacking.

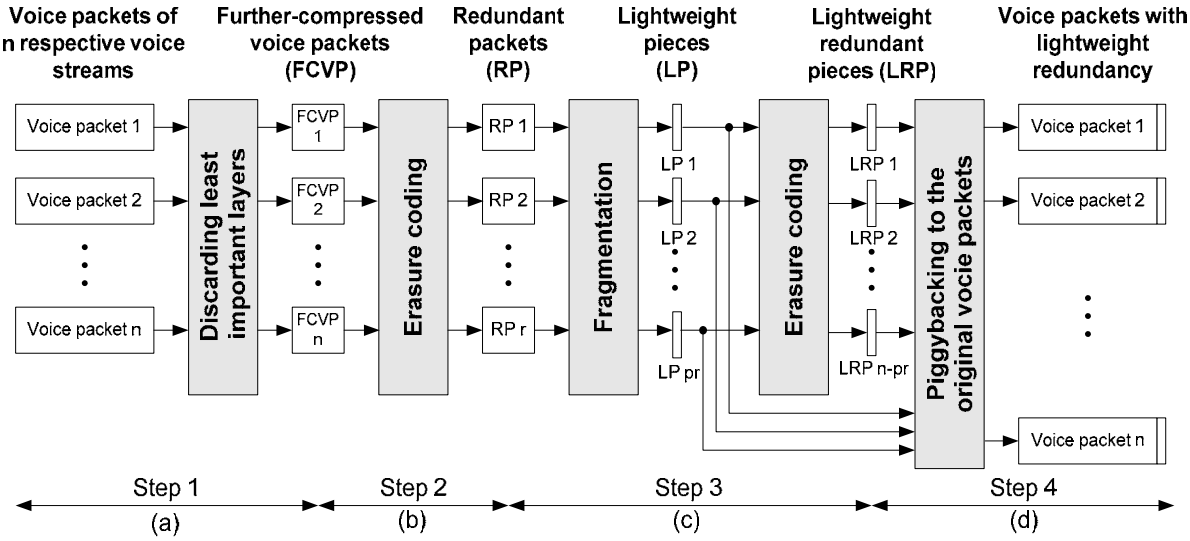


Figure 2. Steps of lightweight piggybacking.

The rest of the paper is organized as follows. In section II, we describe the lightweight piggybacking scheme. In section III, we explain the packet loss recovery procedure. In section IV, we generalize the lightweight piggybacking scheme to multipath communication environment such that the packet loss recovery capability is further improved. In section V, we present simulation results to demonstrate the effectiveness of lightweight piggybacking. Finally, we conclude our study in section VI.

II. LIGHTWEIGHT PIGGYBACKING

Suppose a source gateway is sending *n* voice streams to a destination gateway. To compute and add redundancy, the source gateway executes four main steps which are described in the following.

A. Step 1 – Discarding Least Important Layers

We suppose that each of the *n* voice streams has been suitably compressed. For each stream, we use its further compressed version to compute the redundancy in the subsequent steps. This is done by simply discarding the least important bits.

For example, many voice coding schemes belong to the class of multilayer coding [16]-[17], by which a voice stream is composed of a base layer (containing the most important information) and multiple enhancement layers (containing additional information). In this case, we can discard some least-important enhancement layers of a voice stream to obtain its further compressed version.

In each packetization period, there are *n* voice packets from the *n* respective voice streams. After discarding their least important bits, we produce *n* further compressed voice packets (FCVP). Fig. 2(a) shows the schematic.

B. Step 2 – First Stage of Erasure Coding

We apply erasure coding on the *n* further compressed voice packets (obtained in Step 1) to produce *r* redundant packets (RP), where *r* is a design parameter. Fig. 2(b) shows the schematic.

The RPs are included to recover the loss of FCVPs. In particular, when the destination can eventually obtain at least *n* out of the *n* FCVPs and *r* RPs, it applies erasure decoding to recover all the *n* FCVPs.

C. Step 3 – Fragmentation and Second Stage of Erasure Coding

Step 3 is a core step of lightweight piggybacking. It performs fragmentation to further reduce the redundancy per voice packet and executes another stage of erasure coding to improve the loss recovery capability.

Fig. 2(c) shows the schematic. First, we perform fragmentation to divide each redundant packet (RP) produced in Step 2 into *p* pieces called *lightweight pieces* (LPs), where *p* is a design parameter. Totally, there are *pr* lightweight pieces. Second, we perform erasure coding on these *pr* lightweight pieces to produce *n – pr lightweight redundant pieces* (LRP). In this manner, when the destination can obtain any *pr* pieces (including LPs and LRPs), it can recover all the LPs so that it can reconstruct all the *r* RPs.

D. Step 4 – Piggybacking

After Step 3, we obtain *n* pieces (*pr* lightweight pieces and *n-pr* lightweight redundant pieces). We piggyback each piece to one distinct voice packet. Fig. 2(d) shows the schematic. After piggybacking, the resulting voice packets with lightweight redundancy are transmitted to the destination gateway.

E. Discussion

Redundancy requirement: In conventional piggybacking, a further compressed voice packet is piggybacked to each voice packet. In lightweight piggybacking, a lightweight (redundant) piece is piggybacked to each voice packet where a piece is *p*

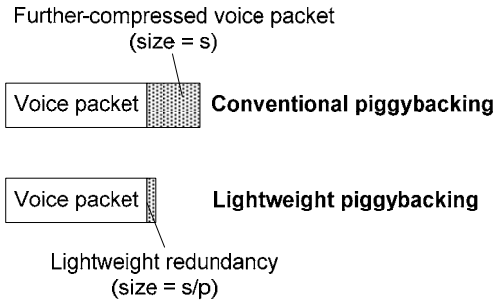


Figure 3. Lightweight piggybacking has smaller redundancy than piggybacking.

times smaller than a further compressed voice packet because of fragmentation in Step 3 (see Fig. 2). Fig. 3 illustrates the redundancy requirement.

Loss recovery capability: In conventional piggybacking, each individual voice stream uses its own redundancy for loss recovery. In lightweight piggybacking, multiple voice streams share the lightweight redundancy to a great extent via two stages of erasure coding, so that it has better loss recovery capability. We will present simulation results to demonstrate this point in section V.

III. Packet Loss Recovery

If the destination cannot receive all the n voice packets in a packetization period, it makes use of the lightweight redundancy for loss recovery. The steps are described in the following:

A. Step 1 – Reproducing Redundant Packets by Erasure Decoding and Reassembly

When the destination receives any k out of n packets, it extracts the lightweight (redundant) pieces from the received packets, performs erasure decoding on these pieces to obtain the lightweight pieces, and then reassembles these lightweight pieces to form the redundant packets.

B. Step 2 – Producing Further-compressed Voice Packets

The destination extracts k voice packets from the k received packets and discards their least important layers to produce the further compressed voice packets.

C. Step 3 – Recovering Lost Packets by Second Stage of Erasure Decoding

The destination executes another stage of erasure decoding on the redundant packets obtained in Step 1 and the further-compressed voice packets obtained in Step 2, so that it recovers the lost packets (further-compressed version).

D. Illustrative Example

Fig. 4 shows an example to illustrate the steps for loss recovery. In this example, the source produces 6 voice packets but the destination can only receive packets 1, 2, 5 and 6.

- In Step 1, the destination extracts 2 lightweight pieces and 2 lightweight redundant pieces, performs erasure decoding on these 4 pieces to recover the 4 lightweight pieces, and reassembles them to form 2 redundant packets.
- In Step 2, the destination extracts 4 voice packets and discards their least important bits to produce 4 further compressed voice packets.
- In Step 3, the destination performs the second stage of erasure decoding on the 4 further compressed voice packets and the 2 redundant packets. In this manner, it can recover the 2 lost packets (the further compressed version).

E. Conditions of Loss Recovery

The destination can recover the loss if any one of the following conditions is satisfied:

1) The destination receives at least $n - r$ voice packets and pr pieces (lightweight pieces or lightweight redundant pieces). In this case, the destination recovers the pr lightweight pieces via one stage of erasure decoding, constructs r redundant packets by reassembling these pr lightweight pieces, and applies the second stage of erasure decoding on the $n - r$ further compressed voice packets and the r redundant packets to recover all the lost packets (further compressed version).

2) The destination receives at least $n - r'$ voice packets and pr' lightweight pieces such that it can reassemble them to produce r' redundant packets for any $r' < r$. The destination can then apply erasure decoding on the $n - r'$ further compressed voice packets and r' redundant packets to recover all the lost packets (further compressed version).

IV. Lightweight Piggybacking over Multipath

Multipath streaming is a recent hot research topic [18]. Using this approach, a media stream is coded into multiple bitstreams of equal importance via multiple description coding, and these bitstreams are transmitted over multiple diverse paths in the Internet. This can reduce the adverse effect caused by the variability of individual paths, thereby achieving more stable transmission of the media. In multiple streaming, one of the research issues is how to enforce to transmit the bitstreams over multiple paths [18].

For the Internet telephony system shown in Fig. 1, we can easily enforce to send the voice streams over multiple paths. Consider the example shown in Fig. 5. A source gateway sends a part of the voice streams to an intermediate gateway and then the destination gateway, sends another part to another intermediate gateway and then the destination gateway, etc. This enforces to send the voice streams over multiple diverse paths, thereby achieving more stable transmission of the voice stream.

In the following, we further enhance the lightweight piggybacking scheme for multipath environment such that it is more powerful for packet loss recovery.

A. Source Telephone Gateway Operation

Suppose there are N_s voice streams at the source telephone gateway. In every packetization period, each stream produces one voice packet, forming a total of N_s voice packets. Assume

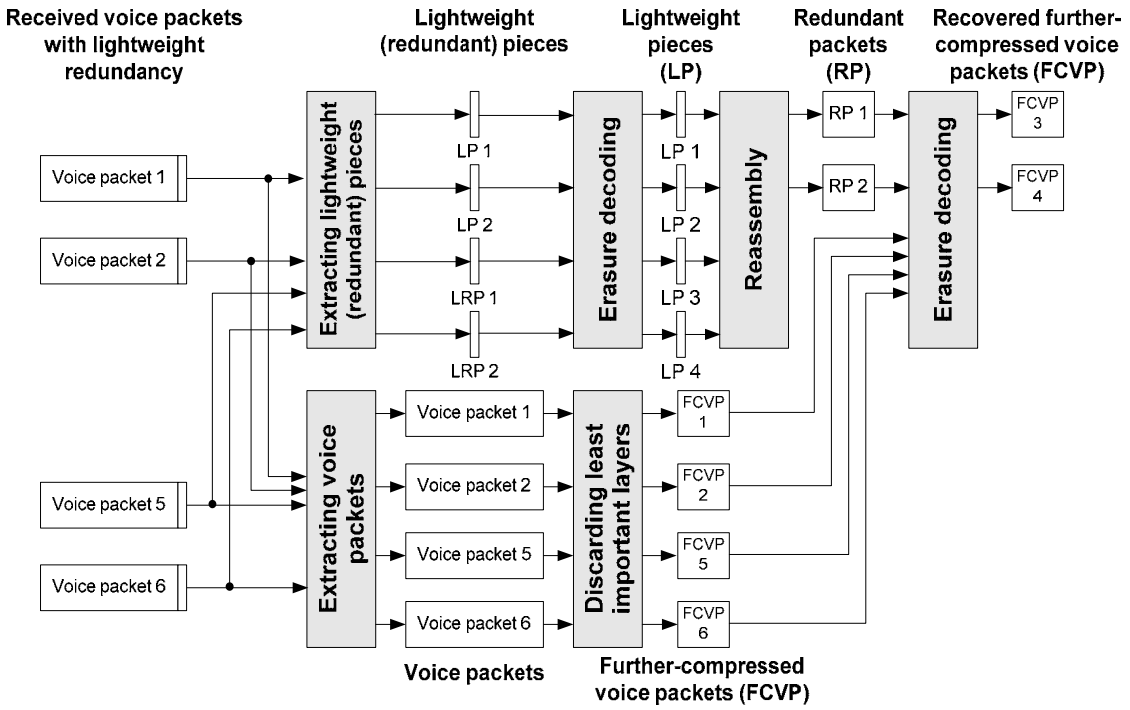


Figure 4. Packet loss recovery of lightweight piggybacking.

there are N_p disjoint paths from the source telephone gateway to destination telephone gateway. The N_s voice packets are then divided into N_p groups and distributed evenly on N_p paths. When N_s is not divisible by N_p, i.e. $N_s \bmod N_p \neq 0$, some groups or paths will have one more packet than others. The number of packets (m_x) transmitted on the x th path [19] is calculated using (1).

$$m_x = \left\lfloor \frac{N_s}{N_p} \right\rfloor + c_x, \quad 0 \le x < N_p, \tag{1}$$

where $$c_x = \begin{cases} 1, & if\ x < N_s \bmod N_p \\ 0, & otherwise \end{cases}$$

The performance of erasure coding degrades when the number of voice packets gets larger. If the destination gateway decodes a large number of packets at the same time, it will cause significant delay to users. Therefore, we divide the incoming voice streams into groups so that each group can carry out erasure coding and decoding independently with a smaller number of packets. As a result, the whole decoding process will be more efficient.

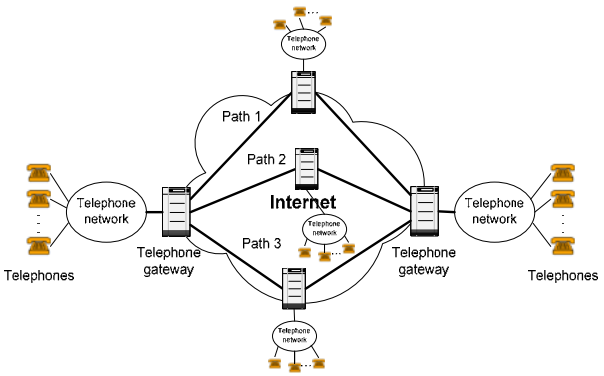


Figure 5. Realization of multipath transmission.

Let n_i be the number of packets in the i th group where $0 \le i < N_p$. Generally, the n_i packets of each group are distributed from the first to last path in a round robin manner. As illustrated in Fig. 6, the first packet of each group is distributed on each path, followed by the second and later packets. Combining FEC with interleaving helps combat loss variability [18]. To reduce consecutive loss of packets in the same group, groups of packets are interleaved with each other. This makes neighboring packets on the same path belonging to different groups and having different sequence numbers. After packet distribution, the number of voice packets for each group on each path will differ by at most 1. Details of the packet transmission algorithm are shown in the Assignment Algorithm.

Each group produces its redundancies independently. For the i th group, r redundant packets are produced by erasure coding where $r \le n_i$. Following the steps of lightweight piggybacking mentioned in Section II, each redundant packet is fragmented into p pieces where $pr \le n_i$. With second erasure coding, a total of l_i lightweight pieces (LP) can be produced from pr pieces. Lightweight pieces are then piggybacked to packets of previous group. For instance, the j th LP of the i th group is piggybacked to the j th packet of the i–1 th group where $0 \le j < n_i$–1. Therefore, the source should produce n_i–1 LP so that all LP of the i th group can be piggybacked to original voice packets of

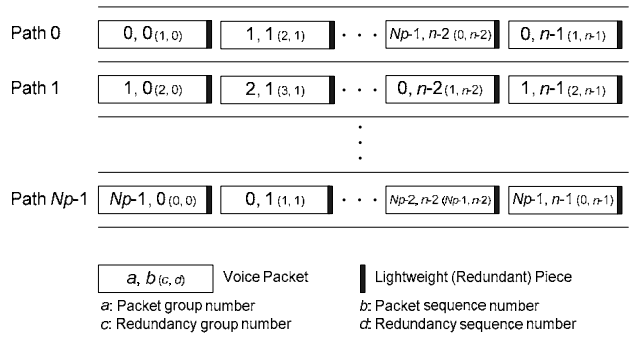


Figure 6. Lightweight piggybacking over multiple disjoint paths.

the i–1 th group.

Using this approach, voice packets and LP of each group can be distributed evenly on each path. This also reduces losses of consecutive packets and LP of the same group. The number of lightweight (redundant) pieces to be produced for each group (l_i) is determined by (2).

$$l_i = n_k, \quad 0 \le i < N_p, \tag{2}$$

where $$k = \begin{cases} i-1, & \text{if } i-1 \ge 0 \\ N_p - 1, & \text{otherwise} \end{cases}$$

After piggybacking, the packets are transmitted into different paths simultaneously.

Assignment Algorithm

/* Assign lightweight (redundant) pieces to voice packets and assign voice packets to paths*/

for j = 0 to packets_per_group **do**
  $k = j \bmod N_p$
  for x = 0 to $N_p - 1$ **do**
    if $j * N_p + x + 1 > N_s$ **then**
      BREAK
    end if
      $i \leftarrow k$
      $k \leftarrow k + 1$
      if $k = N_p$ or $N_s < N_p$ and $k = N_s$ **then**
        $k \leftarrow 0$
      end if
      Piggyback Lightweight Piece (k, j) to Packet (i, j)
      Transport Packet (i, j) into the x th path
      $i \leftarrow k$
    end for
end for

B. Destination Telephone Gateway Operation

At the destination telephone gateway, packets are received from all disjoint paths. Voice packets and LP are then extracted. Each group carries out packet loss recovery independently. The recovery conditions can be divided into two cases. For each group, the destination should receive:

1) at least $n_i - r$ packets AND pr LP from all disjoint paths
2) at least $n_i - r$' packets AND reassembling pr' LP to form r' RP where $r' < r < n$

Each group recovers lost packets following the recovery steps mentioned in Section III.

V. Simulation Results and Discussion

We conduct simulation experiments to evaluate the performance of lightweight piggybacking. We adopt two performance measures: (1) residual packet loss rate P_R which is the packet loss rate after performing packet loss recovery [3]-[4], and (2) the amount of redundancy required.

A. Simulation Model

Many experimental studies [19]-[20] reveal that packet losses in the Internet are dependent, so we adopt the two-state Markov model to model this type of dependent loss [21]-[22]. In this model, each transmission link has two states, i.e. 0 and 1. State 0 is a zero loss state while state 1 is a lossy state. As shown in Fig. 7, P_{01} is the transition probability from state 0 to state 1, while P_{10} is the transition probability from state 1 to state 0. We assume P_L to be the average packet loss rate and L_B to be the average burst length (average number of consecutively lost packets). P_{01} and P_{10} are calculated using (3) and (4) respectively.

$$P_{10} = \frac{1}{L_B} \tag{3}$$

$$P_{01} = \frac{P_{10} \times P_L}{1 - P_L} \tag{4}$$

In multipath transmission environment, we assume that the paths are independent and have the same statistical loss behavior.

According to some measurement experiments [23], the average packet loss rate of the Internet is below 0.1, so we select the range of average packet loss rate in our simulation experiment to be between 0.01 and 0.1.

B. Single Path Transmission of Lightweight Piggybacking

In this simulation experiment, we compare the performance of lightweight piggybacking with the conventional piggybacking (Fig. 8 & 9).

We vary the average packet loss rate but keep the average burst loss length unchanged. It can be seen in Fig. 8(a) that lightweight piggybacking performs better than piggybacking. When P_L is low (e.g. 0.01), P_R of piggybacking is 6.7×10^{-3} while that of lightweight piggybacking (R=0.3, p=2) is 5.3×10^{-5}.

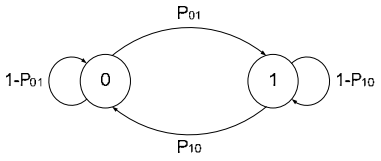


Figure 7. Two state Markov model.

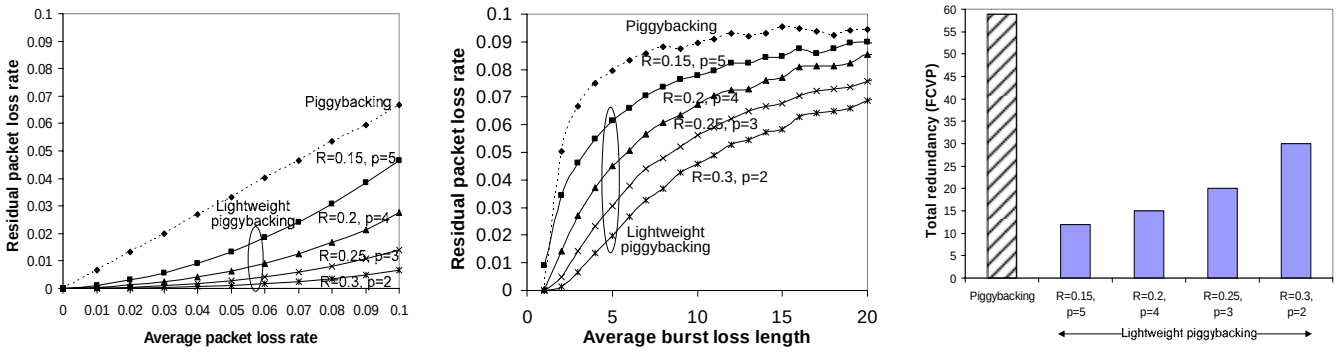


Figure 8(a). Effect of increasing average packet loss rate on piggybacking and lightweight piggybacking in single path transmission (n=60, L_B=3).
(b). Effect of increasing burst loss length on piggybacking and lightweight piggybacking in single path transmission (n=60, P_L=0.1).
(c). Total redundancy required for piggybacking and lightweight piggybacking (in terms of the number of further compressed voice packets (FCVP)).

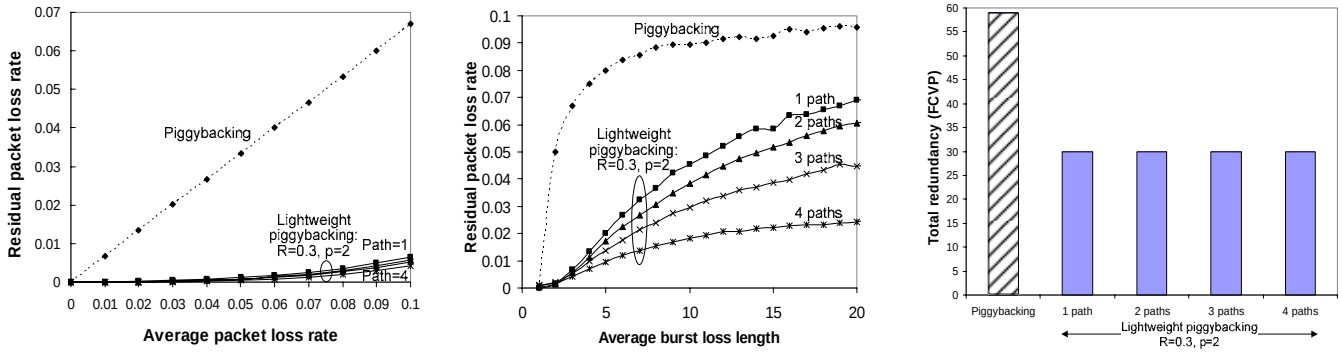


Figure 9(a). Effect of multipath transmission on lightweight piggybacking with increasing average packet loss rate (n=60, L_B=3).
(b). Effect of multipath transmission on lightweight piggybacking with increasing average burst loss length (n=60, P_L=0.1).
(c). Comparison of total redundancy between piggybacking and lightweight piggybacking over multipath.

This shows a 99.2% decline of P_R compared with piggybacking. When P_L is high (e.g. 0.1), P_R for piggybacking seems to rise quickly (to approximately 6.7×10^{-2}), but that for lightweight piggybacking (R=0.3, p=2) is 6.7×10^{-3}, which reduces P_R by 90% compared with piggybacking. Also consider Fig. 8(c), the total redundancy required for lightweight piggybacking (R=0.3, p=2) is only half of that in piggybacking. These suggest that lightweight piggybacking can significantly reduce P_R with considerably lower total redundancy compared with piggybacking.

We show how burst losses affect the performance of piggybacking and lightweight piggybacking by varying the average burst loss length with a fixed average packet loss rate. In Fig. 8(b), when the average burst loss length increases, P_R for both schemes increases sharply. But increasing R for lightweight piggybacking can notably reduce P_R when $L_B < 10$. Compared with piggybacking, lightweight piggybacking (R=0.3, p=2) reduces P_R by around 50% when L_B=10. As L_B further increases, lightweight piggybacking with a larger R tends to perform better. Thus a larger R may be needed to recover lost packets when burst packet losses are serious.

Fig. 8(c) shows the total redundancy for piggybacking and lightweight piggybacking. It indicates that lightweight piggybacking requires much smaller total redundancy than piggybacking. Piggybacking has a redundant packet for each original voice packet, so its total redundancy is very large. In contrast, the total redundancy for lightweight piggybacking depends on the number of pieces (p). When p increases, each redundant piece is smaller and thus the total redundancy is reduced.

C. Effect of Multipath Transmission on Lightweight Piggybacking

In this simulation experiment, we investigate the effectiveness of lightweight piggybacking in multipath communication environment.

Firstly, we vary the average packet loss rate but use the same average burst length. Fig. 9(a) shows that using more disjoint paths for packet transmission can further improve the performance of lightweight piggybacking (R=0.3, p=2). When P_L is 0.01, P_R of piggybacking is 6.7×10^{-3} while that of lightweight piggybacking (R=0.3, p=2) is only 5.3×10^{-6}, which is 99.9% lower than piggybacking. Note that no matter how many paths are used for transmission, R and p for all the four cases of lightweight piggybacking are the same. Thus the total redundancy (see Fig. 9(c)) for them is the same.

Fig. 9(b) illustrates the effect of burst packet losses on piggybacking and lightweight piggybacking. Now the average packet loss rate is kept constant. Burst losses seriously affect the performance of piggybacking even though it keeps a very high total redundancy (Fig. 9(c)). When the average burst loss length is low (i.e. L_B=2), P_R for piggybacking begins to rise

abruptly. Only a few lost packets can be recovered when the $L_B > 10$. This is because the lost packet from a particular voice stream cannot be recovered when the redundant packet is also lost in the next voice stream.

Secondly, we use the same average packet loss rate but different burst loss lengths. The effect of multipath transport for lightweight piggybacking in bursty loss environment is shown in Fig. 9(b). It demonstrates that if more disjoint paths are available, the effect of burst losses can be considerably reduced. When the transmission link condition is extremely bad (e.g. L_B=20), piggybacking cannot recover the lost packets at all. On the other hand, single path transmission for lightweight piggybacking also gives a high P_R (i.e. 6.9×10^{-2}). Significant reduction of P_R begins when more than 2 disjoint paths are used. When 4-path transmission is used, P_R can be further reduced to around 2.4×10^{-2}, which shows 60% and 46.7% reduction of P_R compared with 2-path and 3-path transmission respectively. These may be explained by the effect of multipath transmission. It ensures the original voice packets and their corresponding redundancies to be transported on different paths, such that they are not lost concurrently when one path is congested. Moreover, as packets are divided into groups and distributed evenly on each independent path, the burst loss probability of packets in the same group is significantly reduced. Thus multipath transmission can significantly reduce P_R of lightweight piggybacking with the same total redundancy (Fig. 9(c)).

VI. Conclusion

We proposed the lightweight piggybacking scheme for packet loss recovery in the Internet telephony system shown in Fig. 1. This scheme is an improved version of the well-known piggybacking scheme. It applies two stages of erasure coding and fragmentation, such that only a small redundancy is added to each voice packet while this redundancy can be shared by multiple voice streams to a large extent for effective packet loss recovery. We observed the following results from our simulation experiments:

1) Compared with piggybacking, lightweight piggybacking can significantly reduce the residual packet loss rate (i.e., increase the chance of recovering the lost packets) using the same or smaller redundancy.
2) Lightweight piggybacking is particularly more effective than piggybacking when multiple and consecutive packets are lost.
3) When the telephone gateways of the system form a multiple transmission environment, lightweight piggybacking is even more powerful in packet loss recovery.

References

[1] "Special Issue on Internet Telephony," *IEEE Internet Computing*, vol. 6, no. 3, May/Jun. 2002.

[2] "Special Issue on Internet Telephony," *IEEE Communications Magazine*, vol. 38, no. 4, Apr. 2000.

[3] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, Sep/Oct. 1998, pp. 40-48.

[4] W. Jiang and H. Schulzrinne, "Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss", *in Proc. of International Workshop on NOSSDAV*, Miami Beach, Florida, May 2002.

[5] S. W. Yuk, M. G. Kang, B. C. Shin, and D. H. Cho, "An adaptive redundancy control method for erasure-code based real-time data transmission over the Internet," *IEEE Trans. Multimedia*, vol. 3, no.3, Sep. 2001, pp. 366-374.

[6] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Chapter 6, 2nd ed., Addison Wesley, 2003.

[7] K. Mochizuki, Y. Yoshimura, Y. Uematsu, and R. Suzuki, "Forward error correction for visual communication systems using VBR codec," *IEICE Trans. Commun.*, vol. 89, no. 2, Feb. 2006, pp. 334-341.

[8] J. C. C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for Internet telephony," *Proc. IEEE INFOCOM*, 1999.

[9] P. Frossard, "FEC performance in multimedia streaming," IEEE Commun. Lett., vol. 5, no. 3, Mar. 2001, pp. 122-124.

[10] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols," *ACM Computer Communication Review*, vol. 27, Apr. 1997, pp. 24-36.

[11] A. McAuley, "Reliable broadband communication using a burst erasure correcting code," *in Proc. of the ACM symposium on Communications architectures & protocols*, Philadelphia, United States, 1990.

[12] J. P. Macker, "Reliable multicast transport and integrated erasure-based forward error correction," *in Proc. of MILCOM*, Monterey, USA, Nov. 1997, pp. 973-977.

[13] T. Nguyen and A. Zakhor, "Distributed video streaming with forward error correction," *in Proc. of Packet Video Workshop*, 2002.

[14] Free Phone [Online]. Available: http://www-sop.inria.fr/rodeo/fphone/

[15] Robust Audio Tools [Online]. Available: http://www-mice.cs.ucl.ac.uk/multiedia/software/rat/

[16] S. McCanne, M. Vetterli, and V. Jacobson, "Low-Complexity Video Coding for Receiver-Driven Layered Multicast", *IEEE JSAC*, vol. 16, no. 6, Aug. 1997, pp. 983-1001.

[17] B. Girod, K. W. Stuhlmüller, M. Link, and U. Horn, "Packet Loss Resilient Internet Video Streaming," in *Proc. of SPIE Visual Communications and Image Processing '99*, vol. 3653, Jan. 1999, pp. 833-844.

[18] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," *IEEE Communications Magazine*, Aug. 2004, pp. 80-87.

[19] X. Yu, J. W. Modestino, and I. V. Bajic, "Modeling and analysis of multipath video transport over lossy networks using packet-level FEC," *in Proc. of The Eleventh International Conference on Distributed Multimedia Systems*, 2005.

[20] J. Wenyu and H. Schulzrinne, "Modeling of packet loss and delay and their effects on real-time multimedia service quality," *ACM NOSSDAV*, 2000.

[21] X. Yang, C. Zhu, Z. G. Li, X. Lin and N. Ling, "An unequal packet loss resilience scheme for video over the Internet," *IEEE Transactions on Multimedia*, vol. 7, no. 4, Aug. 2005.

[22] A. Began, Y. Altunbasak, O. Ergun and M. Ammar, "Multi-path selection for multiple description video streaming over overlay networks," *Signal Processing: Image Communication*, vol 20, 2005, pp. 39-60.

[23] AnalogX. (2006, September). Internet traffic report [Online]. Available: http://www.internettrafficreport.com/

Performance Evaluation of IEEE 802.11 DCF with Internal UDP Traffic

Yan Yong

Abstract

With increasing demand for wireless Internet connectivity, there has been a growing interest in improving the performance of current wireless networks, which are based on the IEEE 802.11 protocol. Two types of network operation modes are supported by the 802.11 protocol, one is IBSS (independent Basic Service Set); the other is ESS (Extended Service Set), where all wireless nodes exchange messages through Access Points. This paper is to evaluate the performance of both networks, in different measures, to show the advance of ad hoc network under a certain circumstance, and to demonstrate that the competition for channel occupancy between Access Points and clients is the key factor that causes performance reduction in the infrastructure mode. Possible means of performance improvement are also presented, and could be further discussed in future research work.

1. Introduction

Nowadays, more people like to have wireless connections as the last hop for them to access Internet, in libraries, airports, conference centers, and other public places, where wireless connections could bring more convenience than wired ones. The rapid growth in the number of users and bandwidth requirements has brought challenges to the underlying infrastructure of wireless networks. IEEE 802.11 protocols are the dominant protocols worldwide in wireless network. Among various protocols in the 802.11 family, some of them have been finalized and being popular in industrial products, such as 802.11b and 802.11g; on the other hand, some protocols are still active in research period, or as a 'pre-draft' of IEEE standard, such as 802.11n (pre-draft 2007), 802.11T (working-2008), and even some protocols are still at their proposal stages, such as 802.11u (early proposal stages). This paper will focus on the performance of standard 802.11g networks, and also consider 802.11b in some instances.

Basically, the IEEE 802.11 protocols support two kinds of operation modes: one is the network without Access Point (AP), also known as the *Ad hoc wireless LAN*; and the other is the network that relies on an Access Points as a central node, which is called the *Infrastructure mode*. Instead of transmitting messages from peer to peer as in Ad hoc mode, in the infrastructure mode, any node has to send its message to AP first, while AP is responsible for forwarding all messages to destinations, which can be either inside or outside of this wireless LAN. The infrastructure mode is widely adopted by most of the current wireless Internet networks, where AP performs the role of a portal (router) to Internet, as well as a central node of the local wireless AP network.

To illustrate the difference between wireless Ad hoc mode and infrastructure mode, we provide the basic concepts in both networks.

1.1 The independent BSS as an Ad hoc network [1]

The independent BSS (IBSS) is the most basic type of IEEE 802.11 LAN. A minimum IEEE 802.11 LAN may consist of only two stations.

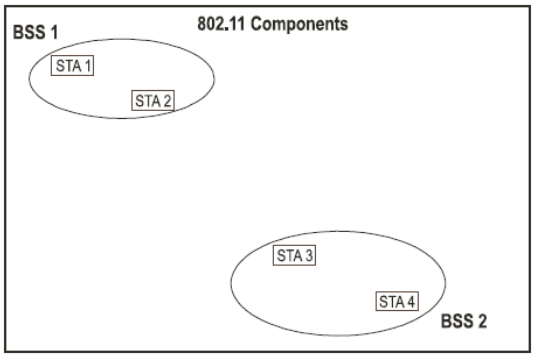


Figure 1 Figure 1—Basic service sets [1]

Figure 1 shows two IBSSs. This mode of operation is possible when IEEE 802.11 stations are able to communicate directly. Because this type of IEEE 802.11 LAN is often formed without pre-planning, for only as long as the LAN is needed, this type of operation is often referred to as an *ad hoc network.*

1.2 Interconnected BSSs via Distribution system and Access Points

Instead of existing independently, a BSS may also form a component of an extended form of network that is built with multiple BSSs. The architectural component used to interconnect BSSs is the *distribution system* (DS).

IEEE 802.11 logically separates the wireless medium (WM) from the distribution system medium (DSM). Each logical medium is used for different purposes, by a different component of the architecture. The IEEE 802.11 definitions neither preclude, nor demand, that the multiple media be either the same or different.

An access point (AP) is a station (STA) that provides access to the DS by providing DS services in addition to acting as a STA.

Figure 2 adds the DS and AP components to the IEEE 802.11 architecture picture. Data move between a BSS and the DS via an AP.

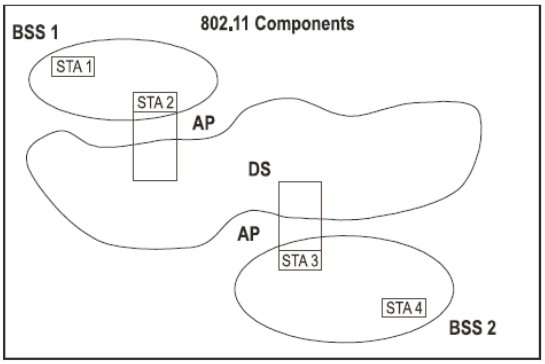


Figure 2 Distribution systems and access points

1.3 Integration with wired LANs

To integrate the IEEE 802.11 architecture with a traditional wired LAN, a final *Logical* architectural component is introduced—a *portal.*

A portal is the logical point at which MSDUs (MAC service data units) [1] from an integrated non-IEEE 802.11 LAN enter the IEEE 802.11 DS. For example, a portal is shown in Figure 3 connecting to a wired IEEE 802 LAN.

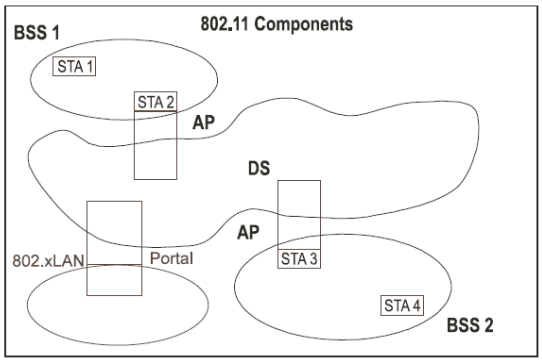


Figure 3 Connecting to other IEEE 802 LANs

In this paper, we first give a brief introduction to the 802.11 MAC protocol and the 802.11g standard, then we analyze the difference between DCF and AP networks, demonstrating the problems with the AP mode. Afterwards, we present our experimental setup and results, followed by some analysis and suggestions of improvement. Finally, we give the conclusion and our future research plan at last.

2. The IEEE 802.11 MAC protocol

The IEEE 802.11 standard is working on both the physical (PHY) and medium access control (MAC) layers of the network. Other than considering about the physical details, we will concentrate on the MAC layer protocol itself.

The basic access method in the 802.11 MAC protocol is DCF (Distributed Coordination Function) known as carrier sense multiple access with collision avoidance (CSMA/CA) [1]. Different from Ethernet, which uses CSMA/CD (Carrier Sense Multiple Access with Collision Detection) [2] as its medium sharing mechanism, collisions in wireless LAN can not always be detected by the involved senders, CSMA/CD is not suitable for wireless LAN, hence the 802.11 standard defines the CSMA/CA mechanism. DCF employs a distributed CSMA/CA algorithm and an optional virtual carrier sense using RTS and CTS control frames. When using the DCF, before initiating a transmission, a station senses the channel to determine whether another station is transmitting [4]. If the medium is found to be idle for an interval that exceeds the *Distributed InterFrame Space* (DIFS), the station proceeds with its transmission. However if the medium is busy, the transmission is deferred until the ongoing transmission terminates. A random interval, henceforth referred to as the *backoff interval,* is then selected; and used to initialize the *backoff timer*. The backoff timer is decreased as long as the channel is sensed idle, stopped when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for more than a DIFS. The station transmits when the backoff timer reaches zero. CSMA/CA is a strategy that intends to avoid collisions, but it can not eliminate collisions. When more than one node are counting down their backoff timers simultaneously, there's a probability *p* that some of them have their timers reach zero at the same time slot, and start transmitting at the beginning of next time slot exactly the same time, which brings a collision. The probability *p* increases with the number of nodes that have packets to send.

Figure 4 illustrates the mechanism of DCF. [4]

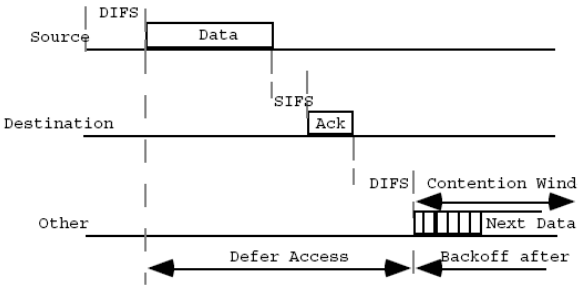


Figure 4 DCF: the basic access mechanism of 802.11 networks.

2.1 The backoff timer

As mentioned before, the DCF requires each node to wait for a random backoff period after the channel is idle for DIFS. The DCF adopts a slotted binary exponential backoff technique. The backoff time is calculated as below,

Backoff Time = Random() × aSlotTime [1],

where Random() indicates a uniformly distributed random integer between [1, CW-1], and CW is the contention window, starting from CWmin, doubled each time a retransmission occurs, until CWmax. Figure 5 is the basic mechanism of Contention Window, and Figure 6 shows an example of the increase of CW value.

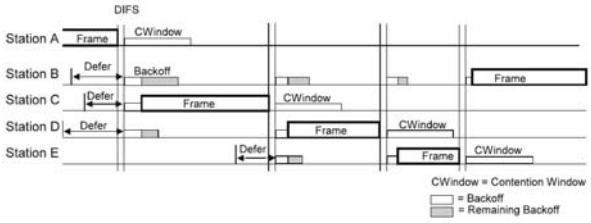


Figure 5 The basic mechanism of Contention Window [1]

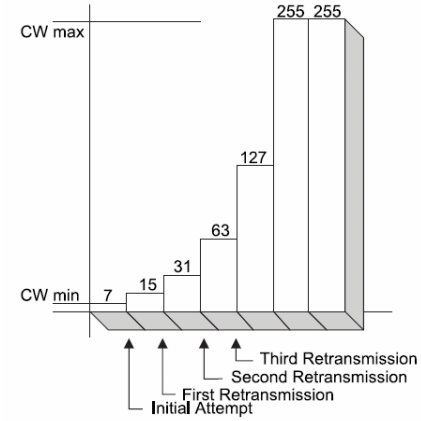


Figure 6 An example of exponential increase of CW [1]

2.2 The 802.11g standard

The IEEE 802.11g standard brings changes and additions to IEEE Std 802.11, 1999 Edition, as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001. In particular, the 802.11g specifies new parameters as summarized in figure 7.

Parameter	Value
SlotTime	9 μ s (short), 20 μ s
SIFSTime	10 μ s
DIFS	28 μs (DIFS = 2 x Slot time + SIFS =)
aCWmin(0)	31
aCWmax	1023
Supported Rates	1, 2, 5, 5, 6, 9, 11, 12, 18, 24, 36, 48, and 54Mb/s
Mandatory Rates	1, 2, 5.5, 11, 6, 12, and 24Mb/s

Table 1 IEEE 802.11g PHY Characteristics [3], [5]

From Table 1 we can see that the theoretical upper bound of maximum transmission rate in 802.11g network is 54Mb/s [3]. However, we will show in the following parts of this paper that in practice, the figure 54M is far beyond what the current network can achieve.

2.3 Comparison of Ad hoc and Infrastructure mode

The main difference between these two modes is that, Ad hoc mode supports peer-to-peer transmission, while the infrastructure mode requires all nodes sending and receiving messages through the Access Point, which means, even though in case that node A and node B can communicate directly to each other, they need to send all their messages to the AP first, and waiting for the AP to forward their messages. Hence, at this point, the infrastructure mode will have an inherent problem, that the traffic is doubled at the point of AP. Figure 8 shows the doubled traffic of infrastructure mode than the Ad hoc mode.

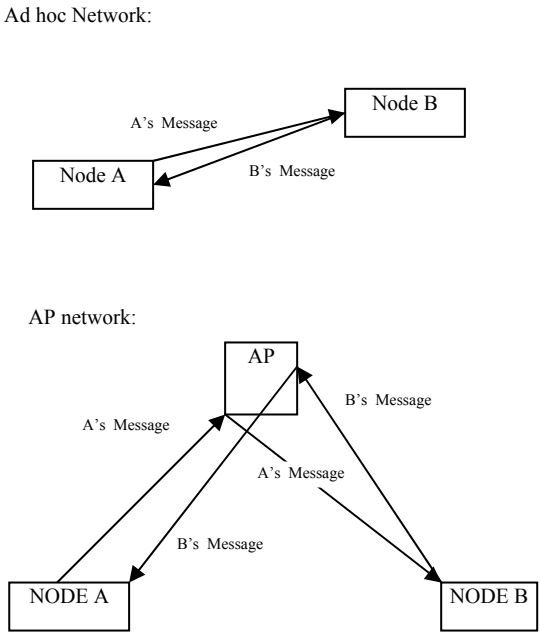


Figure 8 Doubled traffic in AP network.

Since this problem is inevitable in AP network, the performance of Ad hoc network should be better than AP network at least by twice.

Actually, in AP network, the Access Point has to compete with all other nodes, equally sharing the same channel that this wireless network is working on. Theoretically, the chance that AP occupies the channel is *1/n* in a network consisting of *n* nodes including AP. Taking into account that only AP is able to forward a packet to its final destination, the infrastructure mode will perform worse as the number of nodes increases, since AP will have smaller chance to forward messages as *n* goes higher. Under saturation conditions, most of the packets sent from the source nodes can not reach their destinations, as they will be dropped by the AP when AP's buffer is full, which introduce a very big packet loss rate or retransmission rate. In theory, for a network with internal UDP traffic, if the goodput for Ad hoc mode is s_n where *n* is the number of sending nodes, the goodput of the corresponding infrastructure mode is:

$$s_{(n+1)} = (\frac{1}{n+1})s_n, \qquad (1)$$

where *n* is the number of sending nodes in an infrastructure network, excluding the AP, and $s_{(n+1)}$ is the goodput of this infrastructure network with *n* sending nodes and *one* AP. Equation (1) holds under the condition of the network with internal traffic.

To demonstrate all of these, we present a series of experimental results in the following sections.

3. Experimental setup

To evaluate the performance of wireless networks in both Ad hoc and infrastructure modes, we make use of a set of wireless devices and a set of software to capture the real wireless signals in the air. The experimental

environment is shown in Figure 9. The hardware used in our experiments are listed in Table 2.

Figure 9 An experimental Wireless LAN

No.	Name	Role	IP	MAC
0	Netgear	Capturer		
1	XZ6	node	112.142	00:13:02:BA:FA:3B
2	CZ6	node	75.123	00:13:02:B9:95:87
3	CT43	node	21.58	0C:E1:95
4	DELL	node	163.93	E8:EB:01
5	T60	node	46.176	AD:45:77
6	XT43	node	70.253	14:35:0F
7	PD	node	169.254.89.231	00-15-E9-2D-16-37
8	P4	node	168.254.178.132	00-15-E9-2D-0F-CE
9	AP	Access Point	NA	00:13:46:E4:EE:DA
10	LAN CT43	Ethernet	169.254.9.116	00-15-58-09-61-23
11	LAN PD	Ethernet	169.254.81.180	8D-C5-D7

Table 2 List of Experimental wireless devices.

4. Experimental results

To evaluate the performance of wireless networks, our main measure is the UDP goodput of the whole network; we also consider the delaying time and packet loss rate in some instances. Different from **throughput**, which means the total amount of bits of both data and control transferred in one second, **goodput** indicates the number of bits that are successfully received by all the destinations in one second.

4.1 Goodput performance comparison

First of all, in Figure 12, we present the goodput results of Ad hoc and AP networks, with different number of nodes, under the same packet load (1472 bytes UDP packet).

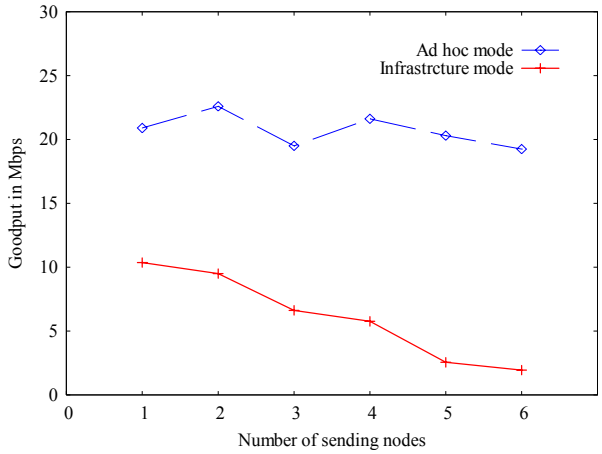


Figure 12 Goodput comparison of Ad hoc and Infrastructure modes. (Upper is Ad hoc, below is Infrastructure)

Figure 12 shows the difference of the two networks. We can see that Ad hoc has a highest goodput of about 23Mbps, while infrastructure mode only gets around 10Mbps at its best. At 1 or 2 nodes, Ad hoc performs better than AP network nearly by twice. While the performance of AP network falls dramatically with the increase of nodes, Ad hoc perform far better than AP in the condition of 3 or more nodes. This result basically coincides with our analysis in equation (1).

It is noted that in Ad hoc mode, the goodput of 3 nodes is lower than the goodput of 4 nodes, the reason of this will be explained later.

Figure 13 presents the goodput comparison under different packet load in Ad Hoc network, with 1, 2, and 3 nodes respectively.

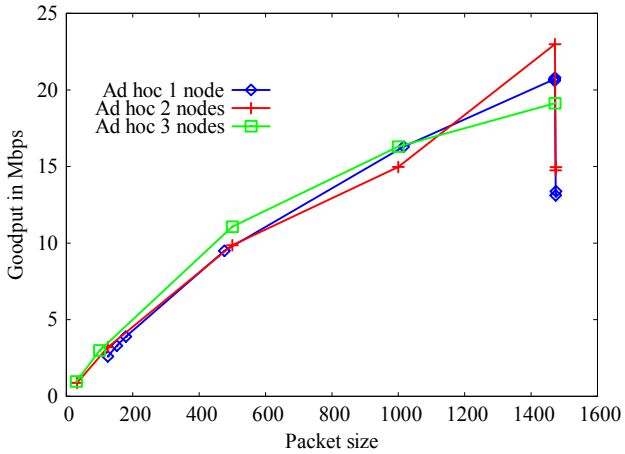


Figure 13 goodput of different packet size in Ad hoc network

While the goodput is rising as long as the packet size increases, a big fall of goodput occurs at the packet size of 1472-1474, because 1472 is the largest IP layer data unit that the wireless link layer can carry in one packet. Meanwhile, we can see from Figure 13 that in Ad hoc network, the increase of nodes has almost no influence on the network's system goodput, when the number of nodes is relatively small.

4.2 Random backoff period test

Of more interest is the test in an infrastructure network with only one wireless node and one AP which is

connected to a PC via Ethernet. This special network helps us to understand the behavior of the wireless node in an environment without competition of channel occupancy from other nodes, which is important to the analysis of random waiting time before sending, in case that the sender is a wireless node, or AP.

4.2.1 Test of a 1-node-to-AP network

Figure 14 presents the average time cost that a node spends in waiting for a random period and sending a 1472 byte packet, in a network of 1-node-to-AP, where AP is connected to a PC via Ethernet. According to the CW strategy in DCF, CW is doubled at each retransmitting. This figure shows that for this device, its random waiting time slot number mainly falls in the interval [0-15] and [31-32], the corresponding waiting time is [280-415] and [550-568]. The rule is, 280+i*9, where 9 is the period of one time slot in 802.11g standard.

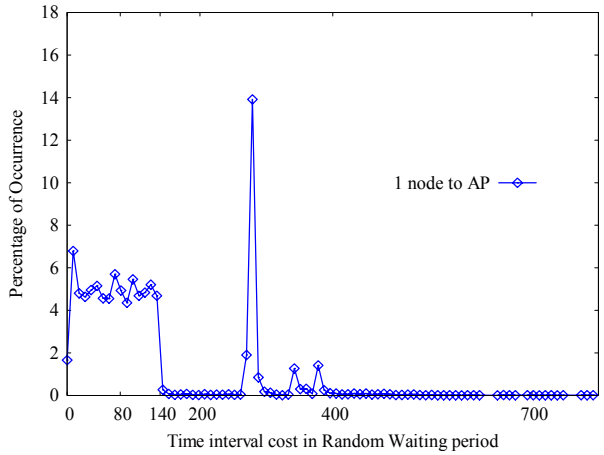


Figure 14 Random waiting time in 1-node-to-Ethernet infrastructure network

4.2.2 Test of an AP-to-1-node network

Similar behavior is observed from AP-to-1-node network, in Figure 15. The only difference is that the Figure 15 does not have a sharp pulse in the interval 550-568 as Figure 14 does.

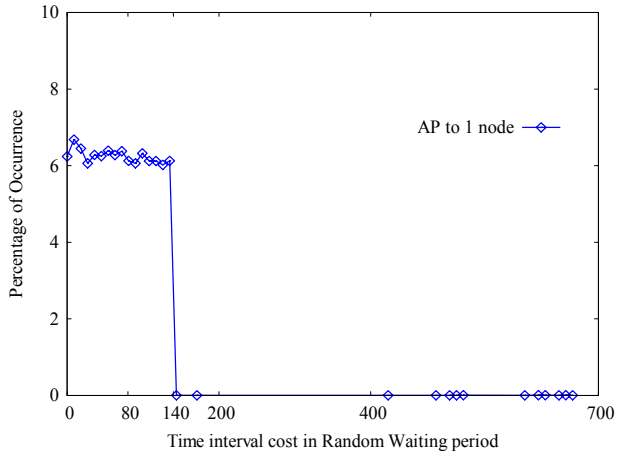


Figure 15 Random waiting time in Ethernet-to-1-node infrastructure network

4.2.3 Test of multi senders in both modes

Figure 16 to Figure 20 are the random waiting period of 1 to 5 senders in Ad hoc or AP networks. These figures show that the average random waiting period becomes longer as the number of nodes increases, no matter in Ad hoc or infrastructure mode. When nodes number increases, the whole system has to spend more time in random waiting period.

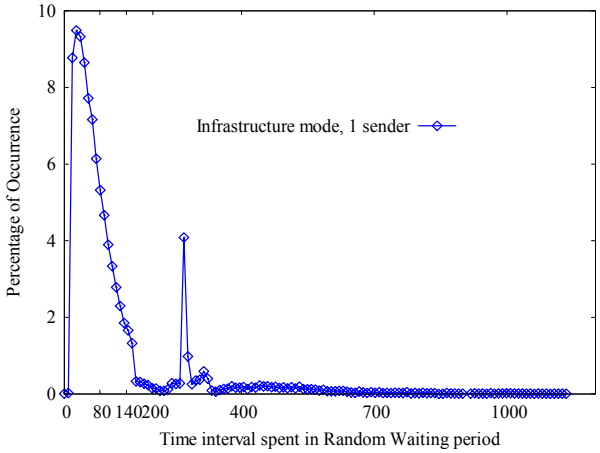


Figure 16 Random waiting time in 1 sender infrastructure network

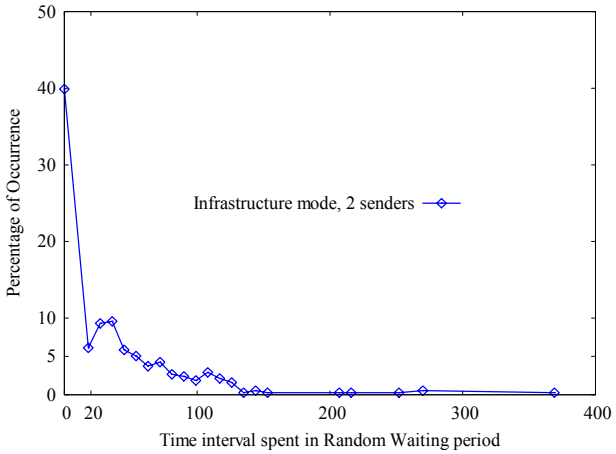


Figure 17 Random waiting time in 2 sender-infrastructure network

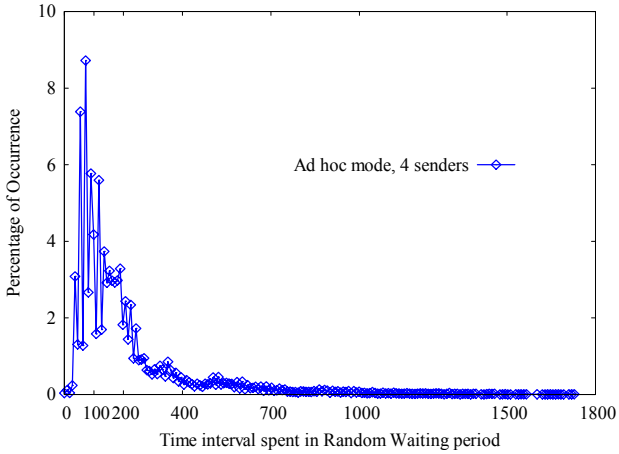


Figure 18 Random waiting time in 2 sender-infrastructure network

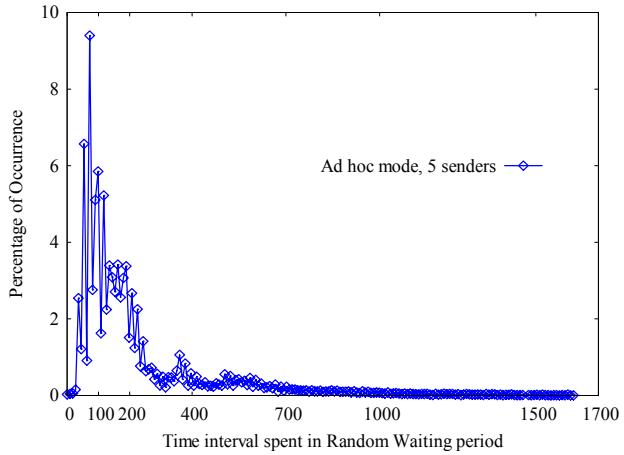


Figure 19 Random waiting time in 5 sender-infrastructure network

Note a special case reported in Figure 17, an unfair competition between different wireless devices is shown. Most of waiting time in figure 17 is in the interval of 260-268, which is the shortest period for a node to wait and send a packet of 1472 bytes in 802.11g network. This situation happens when a node has a higher priority to send as soon as possible, while all other nodes are waiting much longer. That means, a device with shorter waiting period in a wireless network will introduce unfair competition in that network.

In Figure 20, we can see that the 3-node curve is mainly at the right side of the 4-node curve, which indicates the average waiting period spent in this 3 nodes network is longer than that of 4 nodes network. Hence, Figure 21 explains the reason why the goodput of 3 nodes is lower than the goodput of 4 nodes in Ad hoc networks, as illustrated in Figure 12. However, this result does not conflict with the argument that the average waiting time increases as long as the number of nodes increases, because we use different wireless devices in these two tests. In practice, the difference in wireless devices makes slight difference in the overall goodput, while Figure 21 points out what the difference is: the Random Backoff time. When the system employs a shorter average waiting period, 4 nodes network may have better performance than 3 nodes network. At this point, the importance of random backoff time to the system goodput is revealed.

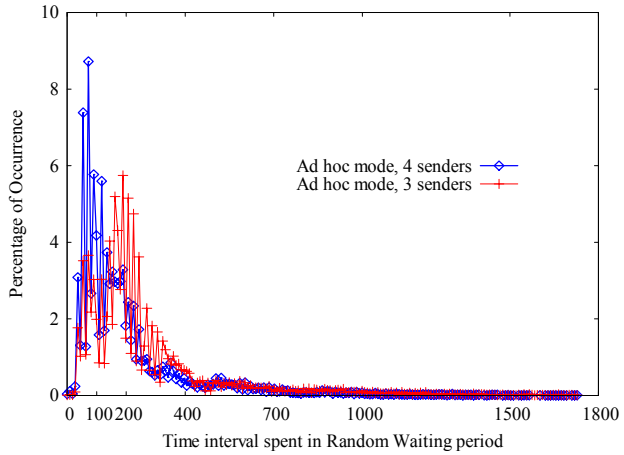


Figure 20 Compare Random waiting time of 3 and 4 sender-Ad hoc network

5. Possible improvement and Future work

According to the above analysis, the competition problem for Access Point in AP network becomes more remarkable as long as the number of nodes increases. To solve this problem, possible means of improvement may include: 1, keep a good ratio of the number of APs and the number of clients. If more clients appear in a WLAN, just set more APs to compete with them. 2, shorten the Access Point's random waiting period, make AP wait for shorter time than other nodes; 3, increase the size of AP's buffer. 4, distinguish different priorities of different services from the clients' requests, as suggested in 802.11e.

Future work could be done on the mathematical modeling and NS2 simulation of large scale network, to prove a convincing way of performance improvement for 802.11 protocols.

6. Conclusion

This paper compares the performance of wireless 802.11 Ad hoc networks and AP networks. Most existing wireless Internet services adopt the AP network mode, which has an inevitable problem that the AP should contend with all other nodes for the channel occupancy. This problem dramatically reduces the system goodput of AP network. Overall performance of Ad hoc network is not influenced by the user scale. In contrast, the increase of number of users is a key factor for the performance of AP networks. To improve overall performance of AP networks, possible means are presented and could be done in future research work.

7. References

[1] IEEE Standard 802.11 - 1999 edition.

[2] IEEE Standard 802.3 - 2005 edition.

[3] IEEE Standard 802.11g - 2003 edition.

[4] F.Calì, M. Conti, E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement", INFOCOM, San Francisco, USA, 1998.

[5] Liam Murphy, "PERFORMANCE OF VOIP OVER IEEE 802.11G DSSS-OFDM MODE WITH IEEE 802.11E QOS SUPPORT"， 2nd International Conference on E-Business and Telecommunication Networks, Reading, U.K., Oct. 3-7, 2005.

Design of Proportional Delay Guarantee Controller of a Cluster-Based Web Server

Chan Ka Ho

Abstract

PI Controller has attracted researchers in industrial control process because of its simplicity and robust performance in a wide range of operating conditions. It has been used to provide proportional delay differentiation on web servers in previous work. Besides PI controller, fuzzy controllers are also proposed on top of single server. The usage of web applications are keep increasing, which cause the increment of pressure to web server. The single server may be overloaded, so the cluster-based web servers are implemented. However, there are no controllers on top of the cluster-based web servers for delays guarantee. This paper will introduce the implementation of controllers of cluster-based web servers and how to guarantee the fairness of the different classes.

Keyword: QoS, Web Servers, Delays, Dispatcher

1 Introduction

With the wide spread usage of all kings of web applications, the access rate of popular web sites are growing rapidly, which causes increasing pressure to web servers. On the other hand, web servers experience an extreme variation in access demand: sometimes very lightly loaded, sometimes suffered from enormous connection requests caused by the flash crowds. It is not economically feasible to design web servers for peak load, because even well-equipped web servers may still be overloaded by the increasing Internet user populations. During overload period, not all requests can be served in a timely manner. However, it is possible to provide a better service to premium users. Performance-enhancing mechanisms that can achieve such QoS properties during server overload are therefore of major importance.

Previous work has proposed the proportional delay differentiation model for web servers. It aims to maintain prespecified delay ratios between different classes of client requests. Lu et al. have proposed a PI controller to guarantee delays among different classes [14]. Due to the difficulty of building an accurate model for web servers with non-linear properties, the authors regarded the web server as a second order system, and determined the system parameters by system identification technique. However, PI controller still cannot get satisfactory results on some performance metrics such as the settling time and oscillation. On the contrary, fuzzy controller is independent of accurate models and could be a good selection for providing delays guarantee [5] [15] [7] [9]. But the main drawback of fuzzy controller is large amount of parameters to be tuned. It is especially difficult to make initial approximate adjustment because there is no cookery book to do the job [9]; and the performance usually depends on the quality of expert knowledge.

We proposed to implement controllers in the cluster-based web servers. The controller, front-end machine, analyzes the requests from web clients. According to the object requested and priority of users, the controller forward the packets to one of the corresponding back-end web servers. The idea of controller and dispatching algorithm will be discussed in the rest of the paper.

The rest of the paper is organized as follows: Section 2 introduces some background information. Section 3 presents the idea of request distribution. The architecture of the system is described in Section 4. Finally, conclusion of the whole paper will be in Section 5.

2 Background

There are two types of computers, front-end dispatching node and back-end web server nodes, in cluster-based web servers as shown in fig. 1.

Some technical background of dispatching node, back-end web servers and quality-of-service will be discussed in this section.

2.1 Front-End Dispatching Node

In the cluster-based web servers, one of the computers, also known as dispatcher, is acted as requests receiver, controller of dispatching requests and the response forwarder.

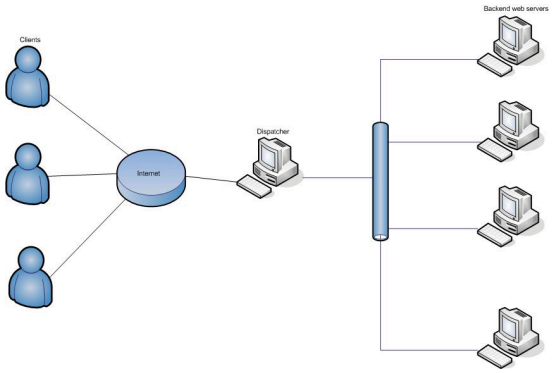


Figure 1: The architecture of cluster-based servers

It plays an important role for the cluster-based web servers. The following parts will discuss each role of the front-end machine thoroughly.

- *Request Receiver*

  The front-end machine starts up number of processes at the very beginning. The processes are used to listen to the well-known port and accepts every incoming connection request.

  When client requests for connection to the cluster-based web servers, one of the processes completes the three-way handshaking for the establishment of the TCP connection. Then, it waits for the request from the client.

- *Controller of Dispatching Requests*

  After the receiving of requests, controller classifies requests into different classes, for example, based on client's IP address. At the same time, a process is responsible to read the request body. The request is forwarded to corresponding web server according to the information of the object requested. The request is forwarded to one of the back-end web servers.

- *Response Forwarder*

  The requests are forwarded to back-end node. After the end of processing of request, the object requested will be transferred to the front-end machine. And the front-end node will forward the response to the specific client. In such design, the server architecture is user-blinded. Therefore, the replacement of back-end nodes is very easy and convenient.

2.2 Back-End Server Nodes

For the back-end nodes, they are only need to handle the requests forwarded by the dispatcher. In order to provide web services for clients, Apache [1] is employed. Apache is a well-known HTTP servers.

Apache is typically structured as a pool of workers, which are threads for the Apache 2.0.59, which handle HTTP requests as discussed before. A worker is responsible to handle one single connection. Therefore, the number of workers is the performance bottleneck of the web server. Besides, harddisk, memory and CPU resources are some limitations for the web server as well. For nowadays technological point of view, it is not significant. In order to boost the quality of services, multiple web servers are required.

Furthermore, the idea of specialized nodes is introduced. Web server nodes are responsible for different partition of objects. When the objects are requested, they are stored in the cache memory for future use. As more objects are stored in the cache memory, the requests of objects from harddisk are reduced. By reducing transfer time from harddisk, a better service can be provided. Therefore, specialized back-end server nodes are beneficial to the cluster-based web servers.

2.3 Quality of Service

Assume the connection delay denote the time interval between the arrival of a connection request and the time the connection is accepted. Let the processing time be the time slot between the web server starting to process a request and sending the response to clients. For the idea of Quality of Service, QoS, connection delay is focused, and we simply define "delay" as the connection delay in the remaining part of this paper.

Research on providing delay differentiated services to web systems is popular in recent years. In [4], an admission control method based on PI controller is proposed. However, the controller is based on the system model which assume the web server can be modeled by a M/G/1/PS model. According to [11], the model is not accurate enough to control web server system.

In addition to PI controller, feedback control theory has been applied to web systems for differentiated services in [6]. Furthermore, queuing theory has been introduced in [14] to provide differentiated delay services.

Other than providing differentiated services, feedback control has been used to adjust KeepAlive and MaxClient of Apache in [13]. As a result, the Apache shows quick convergence and stability. However, the two parameters do not directly address metrics of interest to the Web Site.

3 Request Distribution Strategies

The following assumptions hold for request distribution strategy considered in this paper.

- **Front-end dispatching node** is responsible for handling requests. The data from client is forward to one of the target back-end nodes. As a result, it must keep track of connections information and the distribution of documents. The, it uses these information in making load balancing decision.

- **Back-end server nodes** is capable of serving any clients. As the request distribution strategy, the front-end may direct similar requests only to a subset of back-end server nodes.

3.1 Load Balancing

The basic idea of constructing cluster-based web server is coming from overloading of a single web server. If the requests sent to the cluster-based web server can be distributed evenly to different back-end web servers, the loading of every web servers are approximately the same as the assumption.

In [2] [3], weighted round-robin is introduced in their products. The requests are distributed from front-end node to back-end servers in a round-robin manner and weight parameters are introduced as well. The advantage of this distribution technique is good load balancing among back-end nodes. However, requests are distributed to back-end without considering of the types of services requested. Therefore, each web server receives approximately identical working sets and caches approximately same documents. If the working set exceeds certain amount, for example memory size, cache misses will occur. In short, the round-robin distribution of requests affects the performance of the cluster-based web servers.

3.2 Locality

In order to provide a better cache performance, locality is taken place in front-end distribution strategy. A hashing function can be used to partition the requests. Specific partition of requests is forward to a target web server. Therefore, the cache size can be used in a better way which leads better performance.

The down side of purely concerning the locality is the imbalance of back-end web server loading. It is very clear that if a small set of back-end nodes is responsible to provide specific services, the loading of these servers will be much higher. It is clear that a good hashing function can bring us a evenly loading distribution among the back-end servers. Therefore, it can increase cache hit rate as it can enhance the performance metrics of the whole cluster-based web servers.

3.3 Locality-Aware Request Distribution

In [12], Locality-Aware Request Distribution (LARD) is proposed. The goal of LARD strategy is to combine load balancing and locality. Client requests are distributed to back-end web servers according to back-end loading information and the document distribution. The loading information of back-end web servers is defined as the total number of active connections between client and web server.

3.4 Timeout within Same Connection

HTTP/1.1 introduced the idea of persistent connections [10] which allows multiple requests from the same client to reuse an opened TCP connection. The implementation of HTTP/1.1 introduces a timeout between two consecutive requests received through the same TCP connection. For Apache [1], the timeout is governed by the parameter KeepAliveTimeout, which is set to fifteen seconds. Multiple requests can share same connection, so the number of request establish and terminate is reduced.

However, it also introduces the unfair resources allocation of workers. For example, there are n workers for the web server and $n + 1$ client connections. The first n connections are served by the workers and the last connection are waiting for free worker in the waiting queue. The workers will be free if client terminated the connection or the KeepAliveTimeout reached without any additional requests. However, if all the clients send a request for every fifteen seconds, no workers will be free. In case, the last connection will be timeout and re-established for waiting in waiting queue again. It is not fair to the last connection.

For the controller, it is especially essential to handle the problem of timeout between two consecutive requests of the same TCP connection.

4 Dispatching Algorithm

The whole architecture of the cluster-based web servers includes back-end web servers and dispatching node. Fig. 2 is the diagram to illustrate the setup of our cluster-based web servers. The back-end web servers are the same as single standalone web server, so we will not discuss the working principle. The following sub-sections will be thorough discussion of dispatching algorithms.

4.1 Dispatching and Controlling Modules

There are three modules in the dispatching controller, including monitor, controller and the dispatching mechanism. There will be a brief discussion of the modules.

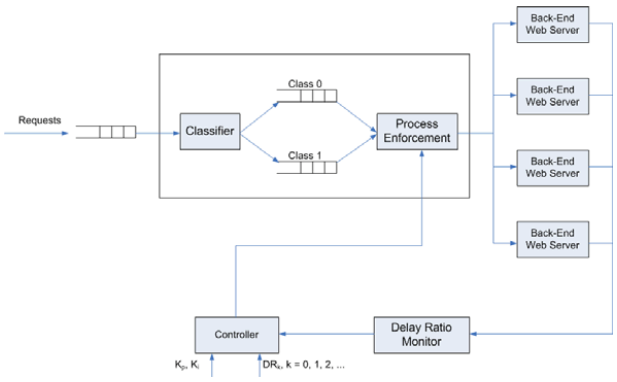


Figure 2: The Cluster-Based Web Server Model

- **Monitor**

  Monitor is responsible to keep track in the proportional delay ratios experienced by requests of adjacent classes k_i and class k_{i+1}, denoted by C_k/C_{k+1}. Therefore, monitor is essential for the the controller. The priority of the users can be defined by the request information, for example, IP address, port requested etc.

- **Controller**

  The proposed controller guarantees our proportional delay ratio. For the premium class, i.e. class 0 and the basic class, i.e. class 1, by assigning suitable number of processes to serve the specific classes of users, say, p_0 and p_1.

  For every control loop, the delay ratio error, which is computed by the difference between the measured delay ratio C_k, C_{k+1} and the desired delay ratio, also known as delay ratio set point, DR_k and DR_{k+1}. By multiplying the errors by proportional gains. The controller decides the number of processes allocated to each class.

- **Dispatching Mechanism**

  There are pre-forked processes on top of the front-end machine. The number of processes to be forked is equaled to the sum of maximum number of workers among all back-end web servers.

  The controller assigns relevant number of the pre-forked processes to specific classes. One of the processes for specific class handles requests from one of the connections of the class. First of all, the process analysis the HTTP request, according to the requested objects, the request is redirected to the corresponding back-end web servers. The response from the back-end web server is received by the process, then the packet will further transfer back to the client. If there are further requests from the same connection, they will be re-directed to the same back-end web server. In short, the dispatching algorithm is client-blinded, that no modifications of clients' browsers are required.

4.2 Non-preemptive Controller

The monitor gathers delays ratio information periodically. With the delay ratio of previous time slot, the numbers of processes for each class can be calculated. The numbers calculated are only valid for the next time slot. And the numbers of further time slot will be calculated with the information of the time slot before it. In short, the delay ratios of current time slot will affect the delays of next time slot and vise versa.

It is very important to find how long a suitable time slot is. If the time slot lasts very long, it is hard for the controller to have a good tuning results because of slow convergence. On the other hand, if the time slot is too short, the controller is over-reactive. It takes vigorous action for a tiny changes of load. Therefore, the system cannot be converged or have great oscillations with large amplitude. Therefore, finding a suitable time slot is essential for controllers. In our experiments, the time slot is determined by the multiples of the timeout between two consecutive requests within the same TCP connection.

For a non-preemptive controller, no connections will be preempted even when the number of processes for a class is less than the expected value. For example, the controller assigns n processes to $class_0$ and $n+1$ processes to $class_1$ at the very beginning of the time slot. If there are only $n-1$ free processes for $class_0$, no processes of $class_1$ will preempt the connection. Indeed, the controller waits for the process to be freed, and it will be assigned to $class_1$. It is very clear that the stored information, such as cookie and session, can be maintained. However, in the worst case, there are requests from all the connection of $class_1$ before the timeout between two consecutive requests within the same connection. The $class_0$ cannot get enough process which leads timeout of the connections of $class_0$. Then, the monitor will measure the delay ratio of the classes incorrectly. Therefore, if non-preemptive controller is employed, we have assumed that the worst case never happens or happens with very low possibilities.

4.3 Preemptive Controller

Preemptive controller is totally different from the non-preemptive controller. If the number of processes for a class is less than the expected value, one of the processes for other priority class preempts the connection. For example, the controller assigns n processes to $class_0$ and $n+1$

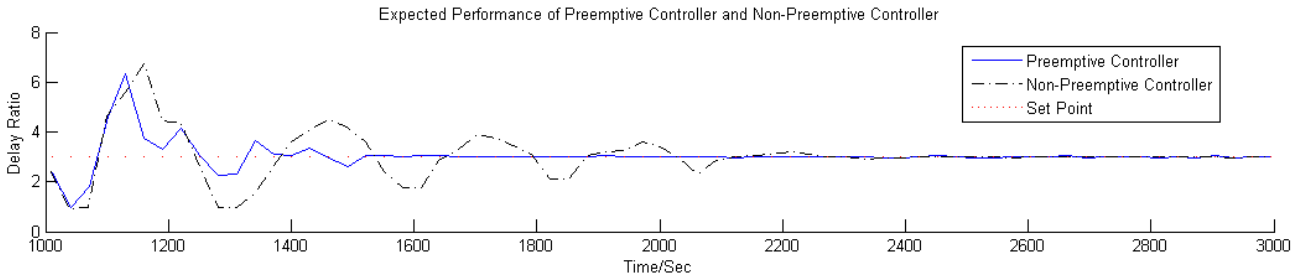


Figure 3: The Expected Performance Difference between Preemptive Controller and Non-Preemptive Controller

processes to $class_1$ at the very beginning of the time slot. If there are only $n-1$ free processes for $class_0$, one of the $class_1$ processes will preempt the connection. It is obvious that the stored information, such as session and cookie, cannot be reused. Therefore, the choice of suitable connection to preempt is essential. The following is a simple comparison of preempting first established connection and last established connection.

- The first establish connection should be preempted. However, it is a disaster to do so, since there are lots of stored information for the first established connection. If the client redo everything but re-establish a connection, all the requests will be redone. In this circumstance, lots of server resources are wasted.
- The last establish connection should be preempted. The amount of stored information is relatively small. In case of redoing everything, it waste less server resources. However, if the stored information is very useful, it is not good to preempt the connection as well.

From the above comparison, we can see that preempting the last established connection is a better choice. However, we have implemented the preemption with a weighting factor. The object to be preempted is chosen from the highest score connection. It can retain the advantage of preempting the last established connection. Moreover, it can safe to prevent preempting some important connection.

5 Experiment

We have developed a program which runs on a Linux platform to implement the adaptive architecture, controller and the dispatching algorithm. The sampling period was set to 30 seconds which is two times of the timeout between two consecutive requests within the same TCP connection. For every sampling period, the controller computes the number of processes assigned to each class.

All experiments were conducted on a test-bed includes a front-end dispatching node with a 2.8GHz Pentium processor and 1 GB RAM running Linux-2.6.12, three back-end web servers with two AMD Athlon 2000+ processors and 2 GB RAM running Linux-2.6.12, and a group of clients running Linux-2.6.12. The server and the clients were connected by a 1-Gbps Ethernet. The Surge workload generator [8] was used to generate web traffic. The percentage of base, embedded, and loner objects were 30%, 38%, and 32%, respectively.

Groups of experiment will be done on top of the test-bed to analysis the performance of preemptive and non-preemptive controllers, the different design of preemptive mechanisms. The experiment is in progress, so there is no exact experimental results. It is foreseeable that the preemptive controller can provide better performance in the matrices of oscillation and the converging time. The expected performance result is as shown in Fig. 3.

6 Conclusion and Future Works

In order to provide better services, the timeout between two consecutive requests from the same connection is very important factor to be concerned. Therefore, preemption is a good decision for the controller. However, for the web contents with stored information, i.e. session and cookie, it is a disaster. All the information stored for the specific clients will be lost.

As we notice the disadvantage of the preemptive approach of the dispatcher, we will have more study on human browsing habit to solve the problem of the downside of preemption. We can have a controller with preemption and try to have shorter sampling period. The second approach is introducing proper algorithm to the controller for choosing the most suitable preempted connection. Lastly, the session or cookie information can be stored or migrated to handle the problem. In short, the following analysis is to reduce the harmful effect caused to the clients.

References

[1] Apache Software Foundation. http://www.apache.org/.

[2] Cisco Systems Inc. LocalDirector. http://www.cisco.com.

[3] IBM Corporation. IBM Interactive network dispatcher. http://www.ics.raleigh.ibm.com/ics/isslearn.htm/.

[4] A. Kamra, V. Misra, and E. Nahum. Yaksha: A controller for Managing the Performance of 3-Tiered Websites. *In Proceedings of the 12th IEEE International Workshop on Quality of Service*, 2004.

[5] B.-G. Hu, G.K.I. Mann, and R.G. Gosine. A Systematic Study of Fuzzy PID Controllers - Function-based Evaluation Approach. *IEEE Transactions on Fuzzy Systems, Vol.9(5):699-712*, 2001.

[6] C. Lu, Y. Lu, T. F. Abdelzaher, J. A. Stankovic, and S. H. Son. Feedback Control Architecture and design Methodology for Service Delay Guarantees in Web Servers. *IEEE Transactions on Parallel and Distributed Systems 2006, vol. 17, no.9, pp.1014-1027*, 2006.

[7] Jianbin Wei, and Cheng-Zhong Xu. A Self-tuning Fuzzy Control Approach for End-to-End QoS Guarantees in Web Servers. *In Proceedings of IWQoS 2005, Lecture Notes in Computer Science, Volume 3552, pp: 123-135*, 2005.

[8] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. *In Proceedings ACM SIGMETRICS'98, Madison WI*, 1998.

[9] P. Pivonka. Comparative Analysis of Fuzzy PI/PD/PID Controller Based on Classical PID Controller Approach. *In Proceedings of tje 2002 IEEE World Congress on Computational Intelligence, USA, 2002, pp.541-546*, 2002.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol-HTTP/1.1. *IETF RFC 2616*, 1999.

[11] V. Paxson and S. Floyd. Wide area traffic: The failure of passion modeling. *IEEE/ACM Transactions On Networking*, 1995.

[12] V. S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum. Locality-Aware Request Distribution in Cluster-based Network Servers. *In Proceedings of the ACM 8th International Conference on Architectural Support for Programming Languages and Operating Systems, pp.205-216*, October, 1998.

[13] X. Liu, L. Sha, Y. Diao, J. L. Hellerstein, and S. Parekh. Online Response Time Optimization of an Apache Web Server. *In Proceedings of 11th International Workshop on Quality of Service, pp. 461-478*, 2003.

[14] Y. Lu, T. F. Abdelzaher, C. Lu, L. Sha, and X. Liu. Feedback Control with Queuing-Theoretic Prediction for Relative Delay Guarantees in Web Server. *Proceedings of IEEE Real Time and Embedded Technology and Applications Symposium*, 2003.

[15] Y. Wei, C. Lin, X.-W. Chu, T. Voigt, and F. Ren. Fuzzy Control for Guaranteeing Absolute Delays in Web Servers. *International Journal of High Performance Computing and Networking*, to appear.

Agent based testbed for relax criteria negotiation

Ng ka-fung

Abstract

Nowadays it is common to see tremendous number of computers are being interconnected to form a grid in order to provide enormous and virtually unlimited computational and storage capacity with arbitrary services available. Efficient negotiation mechanism and strategies would thus provide mutual benefits to both consumer and service providers in the grid, and finally boosting utilization of the grid as a whole. This paper reports experiment results of applying fuzzy logic can be applied in on both side of negotiation agents.

1 Introduction

Computers are often interconnected in large-scale to form a grid to provide enormous and virtually unlimited computational and storage capacity to users. The idea of applying microeconomic in grid then emerges as grid is essentially a resources market which perfectly suit into the traditional microeconomic demand and supply model, and negotiation is central to the idea of economic grid, which in return facilitates automated allocation of resources, a traditionally NP-hard matching problem in combinatorial optimization.

This paper reports the simulation results of incorporating fuzzy logic controller into negotiation kernel of market agents representing and negotiating on behalf of either service providers or service consumers, where the negotiation kernel implementation is based on the Rubinstein's alternating offers protocol [1], and in each offer apart from the initial proposal, they make certain amount of concession by considering the spread between its proposal and the counteroffer based on opportunity, time and competition factors described in [2]. The difficulty lies on the fact that resources negotiation in grid is large-scale multilateral instead of bilateral. The simulation focuses on how fuzzy logic controller helps relaxing stringent negotiation criteria, boosting negotiation success rate without loosing much utility and in addition increasing negotiation speed (reducing number of negotiation rounds required).

This paper is based on [3].

2 Fuzzy Decision Controller

Grid negotiation agents have to deal with wide variety of dynamic market situations to give optimal concession, and the aforementioned opportunity, time and competition (OCT) factors described in [2] are implemented and facilitate agents determining optimal concession to give in each negotiation round. Fuzzy Decision Controller (FDC) is then incorporated into the negotiation kernel to relax stringent concession by OCT. FDC is a prominent choice since agents are facing different levels of negotiation pressures and different market situations, and FDC is a rule-based system to handle all predefined situations.

FDC is deployed on both service consumer and provider agents. It generates a relaxing interval [0, k]. Agents with FDC will consider the counter proposal spread as sufficiently small and acceptable, if the utility difference of the agent proposal and the counter proposal is less then k.

2.1 Consumer FDC

The consumer agent FDC takes two inputs, the failure to success ratio fs_t and demand factor df_t to determine the output, the relaxation degree η, with fs_t $[0, \infty]$ and $df_t, \eta \in [0, 1]$. As agents are not assumed the knowledge of global grid utilization level, the failure to success ratio serves as a possible indicator of recent grid resources competition and utilization – high failure to success ratio may probably due to high grid utilization or strong competition.

Table I. Consumer Fuzzy Decision Rules

No	If fs_t	And df_t	Then η	No	If fs_t	And df_t	Then η
1	N	N	N	9	M	N	N
2	N	L	N	10	M	L	L
3	N	M	L	11	M	M	M
4	N	H	L	12	M	H	H
5	L	N	N	13	H	N	N
6	L	L	L	14	H	L	M
7	L	M	L	15	H	M	H
8	L	H	M	16	H	H	H

N—Negligible L—Low M—Moderate H—High

As different utilization and competition will put different pressure on negotiating agents to reach agreement with trading partners, the FDC guides agents to relax their

trading position according to recent relative demand and negotiation results.

Each consumer agent calculates its own failure to success ratio fs_t at current round t, by taking previous n rounds trading history given as follows:

$$fs_t = \sum_{i=t-n}^{t-1} f_i \Big/ \sum_{i=t-n}^{t-1} s_i$$

where fs_t is the number of request the agent failed to reach a deal with any provider agents before the request deadline is reached, at round i. Similarly s_i is the number of request successfully reached an agreement with provider agent at round i, and n = x / P_m where P_m is the agent's mean event rate, which is the probability of the arrival of a task to a consumer agent in each round (one of the experiment input parameter), and x be the number of negotiation results to consider based on experimental tuning, a typical value ranging from 10 to 20 is used. Hence n controls the number of rounds taken into consideration for recent statistics, such that fs_t reflects only recent failure history.

Each consumer agent computes its demand factor df_t at round t is defined as:

$$df_t = d_t / \max(d_{t-n}, d_{t-n+1}, ..., d_t)$$

where d_i is the total capacity demand at round i, and n in df_t is identically defined as in fs_t, controlling the number of rounds taken into consideration for recent demand statistics, making df_t reflects only the recent demand information.

2.2 Provider FDC

The provider agent FDC also takes two inputs, the current resource utilization level u_t and request factor rf_t to determine the output, the relaxation degree η, with $u_t, rf_t, \eta \in [0, 1]$.

Table II. Provider Fuzzy Decision Rules

No	If u_t	And rf_t	Then η	No	If u_t	And rf_t	Then η
1	N	N	H	9	M	N	M
2	N	L	M	10	M	L	L
3	N	M	M	11	M	M	L
4	N	H	L	12	M	H	N
5	L	N	H	13	H	N	N
6	L	L	M	14	H	L	N
7	L	M	L	15	H	M	N
8	L	H	L	16	H	H	N

N—Negligible L—Low M—Moderate H—High

Utilization level is defined as:

$$u_t = u / p$$

where u is the utilized capacity and p is the total possessed capacity. For the simulated grid environment, agents trade computational resources, thus capacity is measured in million instruction per second (MIPS). In this design of the FDC, provider agents tend to relax more in lower utilization level, in order to obtain agreement utility and avoid resources being idled. On the contrary it relaxes less in low to moderate utilization level, and no relaxation for high utilization, for maximizing utility from users. Agent considers only its own resource utilization level since it has no information of other agents, nor aggregated grid utilization.

Request factor is defined as

$$rf_t = r_t / \max(r_{t-n}, r_{t-n+1}, ..., r_t)$$

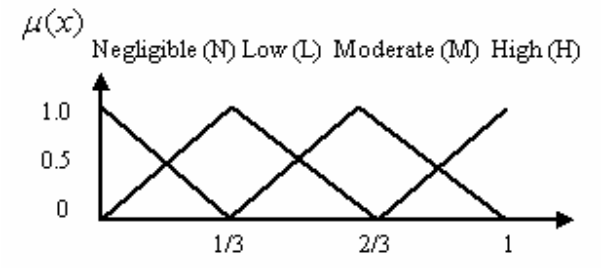
where r_t is the total capacity request at round t. It considers over the previous n rounds, where n = xd. Let $d = \frac{l+u}{2}$ be the average number of rounds in one negotiation session, where l is the lower bound of negotiation deadline, and u is the upper bound of negotiation deadline, so d is provider's mean negotiation deadline for incoming requests. Once a request arrives to provider agent, the capacity demand will persist until reaching a deal or the negotiation deadline, thus mean negotiation deadline is taken as a factor of d as an approximation of number of rounds that a request will persist in provider agent. x is an experimental value typically ranging from 10 to 20, such that rf_t covers recent x negotiation sessions, and reflects the current request level relative to the peak capacity request in previous *n* rounds. rf_t shows the demand relative to the recent peak, and is an indicator of possible future loads in the agent and the grid.

2.3 FDC Internals

The FDC comprises of three components: a fuzzification interface, a fuzzy rule base, and a defuzzification interface.

Fuzzification: Fuzzification interface converts crisp input value into fuzzy representation. Demand factor, utilization level and request factor are evaluated using the percentage membership function defined as below

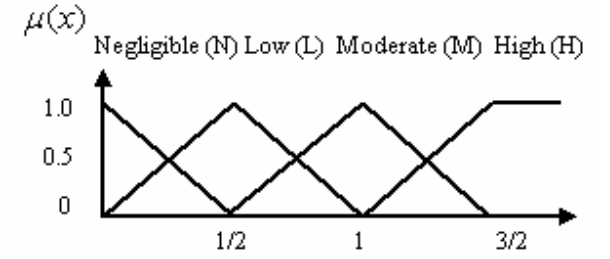
a) Percentage membership function



$$\mu(x) = \begin{cases} -3x + 1 & x \in [0, 1/3] \\ p_1(3x) + (1 - p_1)(-3x + 2), & x \in [0, 2/3] \\ p_2(3x - 1) + (1 - p_2)(-3x + 3), & x \in [1/3, 1] \\ 3x - 2 & x \in [2/3, 1] \end{cases}$$

where $p_1 = 1$ when $x \in [0, 1/3]$, $p_1 = 0$ when $x \in [1/3, 2/3]$
$p_2 = 1$ when $x \in [1/3, 2/3]$, $p_2 = 0$ when $x \in [2/3, 1]$

b) Failure To Success Ratio fs_t membership function



Failure Rate

$$\mu(x) = \begin{cases} -2x + 1, & x \in [0, 1/2] \\ p_1(2x) + (1 - p_1)(-2x + 2), & x \in [0, 1] \\ p_2(2x - 1) + (1 - p_2)(-2x + 3), & x \in [1/2, 3/2] \\ \min(1, 2x - 2) & x \in [1, \infty] \end{cases}$$

where $p_1 = 1$ when $x \in [0, 1/2]$, $p_1 = 0$ when $x \in [1/2, 1]$
$p_2 = 1$ when $x \in [1/2, 1]$, $p_2 = 0$ when $x \in [1, 3/2]$

For example, the fuzzification interface would converts $df_t = 0.60$ to 20% low and 80% moderate according to the membership function defined.

Fuzzy Rule Base: The rule base is a decision controller which aggregates input into output by proportion $fs_t * df_t$. For example if fs_t is 10% negligible and 90% low, with df_t is 40% low and 60% moderate, then by rule 2, it is 0.1 * 0.4 = 4% negligible, by rule 7 it is 0.9 * 0.6 = 54% low, and so on. Relaxation degree η is calculated by $\Sigma\ fs_t * df_t$. In this example, η is determined as 4% negligible and 96% low.

Defuzzification: The defuzzification interface converts the linguistic value of η into crisp value, using *weighted average method* [4], employing also the percentage membership function for defuzzifying relaxation output η. Whether agent will agree with the proposals utility differences can ultimately be determined from η.

3 Testbed

The grid negotiation testbed consists of five major components, (i) a market generator which generates tasks, resources and represented by respective agents. It assigns attributes to generated objects such as negotiation price set of tasks and agents lifespan, (ii) an agent registry provides and manages agent directory service to active agents in the grid space. It execute instruction from market generator like deploy new agent or terminate active agents on their deadline in the grid space, and delisting those expired from the directory, (iii) transaction repository which records succeeded deals and failed negotiation for statistics purposes, such as recent success to failure ratio, (iv) a messaging gateway buffers agent proposals and route to its trading partners in the next round. Agents avoid direct contact with other agents by placing their offers through the messaging gateway, and (v) the grid agent space.

Resource Provider and Consumer Agents: Consumer agents negotiate with provider agents to lease resources, with the objective of reaching a deal with any of its trading partner before the negotiation deadline is reached. Negotiation starts once a task arrives to the agent, and it assigns the initial offer and send to provider agents. Provider agents negotiate by sending counter-offers to consumer agent proposals. Negotiation continues until either an agreement is reached or the negotiation deadline on either side is reached.

Grid Agent Space: The grid agent space is responsible for simulating the entrance of resource consumer agents and provider agents to the market, with assistant from the agent registry. It provides the negotiation framework to agents such as managing rounds to support alternative offers.

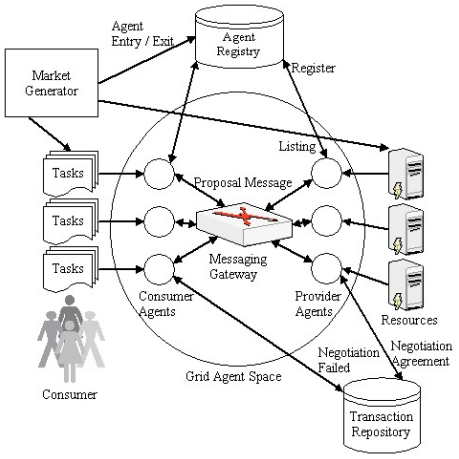


Figure 1. The architecture of the testbed

3.1 Experiment Parameters

The grid controller simulate the negotiation with four input parameters following uniform distribution, (i) market density, the probability an agent will enter the market in all negotiation round, (ii) consumer to provider ratio which determines the probability that determines whether a producer or consumer agent will be generated, (iii) negotiation deadline, the number of rounds a task negotiation cannot exceed deadline, and (iv) the service provider capacity. An additional input, (v) mean event rate, the probability of a task arrival to a consumer agent in each round, following the Poisson distribution. Input (iv) and (v) eventually control the grid utilization.

3.2 Performance Measure

Performance measures are success rate, average utility, expected utility and speed of acquiring resources.

Success Rate	$R_{success} = N_{success} / N_{total}$
Expected Utiltiy	$U_{expected} = (\Sigma U_{success} + \Sigma U_{fail}) / N_{total}$ $= \Sigma U_{success} / N_{total}$
Average Utility	$U_{average} = \Sigma U_{success}) / N_{success}$
$N_{success}$	Number of tasks reached consensus
N_{total}	Total number of tasks negotiated
$U_{success}$	Average utility of a task that reached consensus
$U_{fail} = 0$	Average utility of a task that no consensus reached

Utility function: Let l_{min} and l_{max} be the initial price and reserve price respectively for tasks in consumer agent, (the reverse for provider agent), and l be the price that consensus is reached by both side, then

Consumer: $U_{success} = v_{min} + (1 - v_{min}) (l_{max} - l) / (l_{max} - l_{min})$
Provider: $U_{success} = v_{min} + (1 - v_{min}) (l - l_{min}) / (l_{max} - l_{min})$

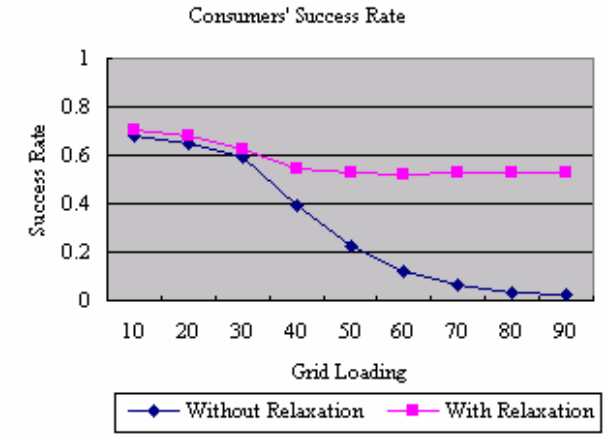
where v_{min} is the minimum utility that agents will get for reaching a deal at reserve price. In this experiment the value of v_{min} is defined to be 0.1. A value that is too close to zero would imply there is little differences with not reaching a deal, where a value too high would make agents making concession easily to aim for success rate.

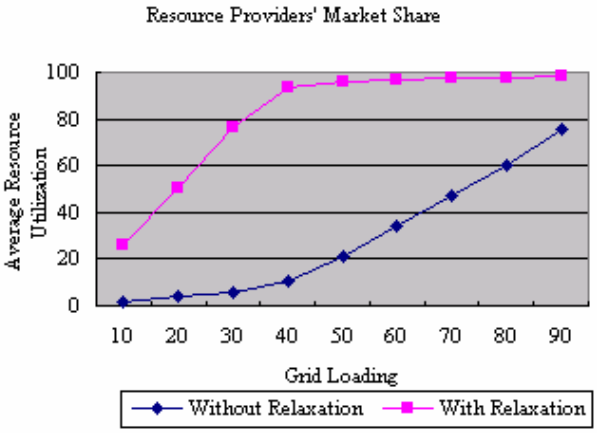
3.3 Simulation Results

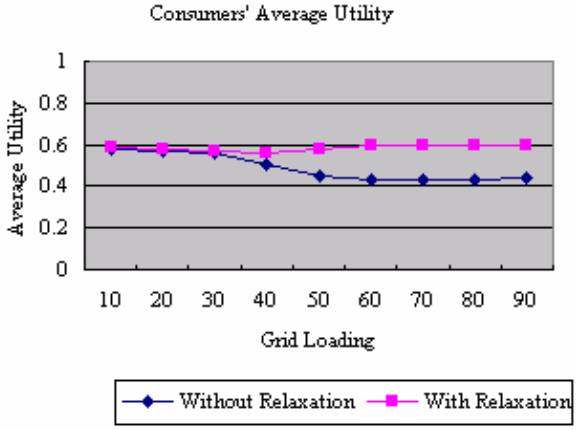
Table III. Input Data Source

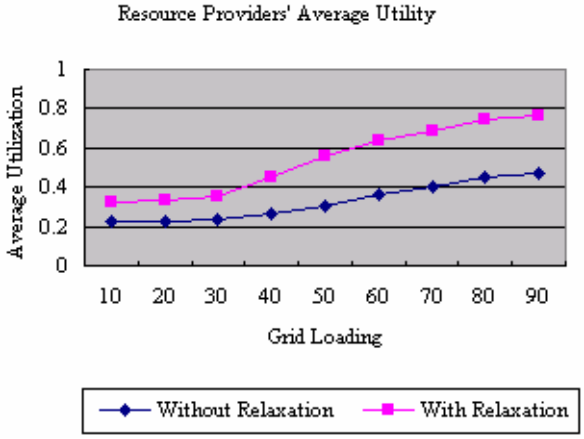
Input Data	**Possible Values**		
Market Type	*Consumer favorable*	*Balanced*	*Provider favorable*
Setting: $\mathbf{P}_{cons}$	<0.5	0.5	>0.5
$\mathbf{P}_{cons}$: *Probability* of an agent *being a consumer agent*			
Characteristic: Market Ratio	*Consumer favorable*	*Balanced*	*Provider favorable*
consumer-to-provider ratio	{1:2 1:5,1:10 }	{1:1}	{10:1,5:1,2:1}
Market Density	*Sparse*	*Moderate*	*Dense*
Setting: $\mathbf{P}_{gen}$	0.2	0.5	0.8
$\mathbf{P}_{gen}$: *Probability* of generating an agent *per round*			
Characteristics: - *Avg. no. of agents*/round	225	560	890
- *Avg. no of rounds an agent stay in market*	750-1500		
- *Max. no. of agents*/round	250	650-680	1140
- *Total no. of agents* (N_{total})	≈100,000	≈250,000	≈400,000
	(generated in entire trading)		
Deadline (No. of rounds)	*Short* (5-15)	*Moderate* (15-30)	*Long* (30-60)
Job Capacity (MIPS)	10-100		
Resource Capacity (MIPS)	200-3000		

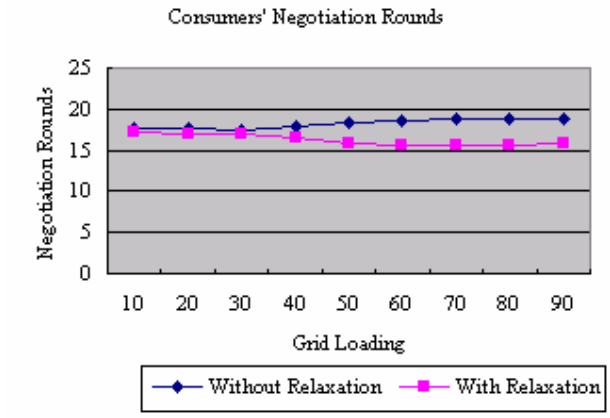
3.3.1 Dense Market Results

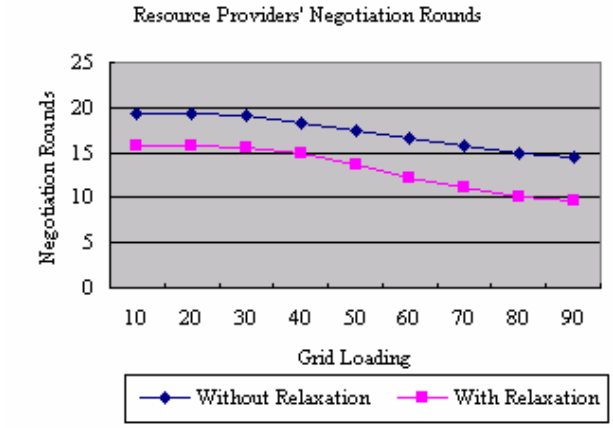












4 Conclusion

Although experiment results are still partial in terms of utilization range, preliminary results shows agents with FDC incorporated into negotiation kernel perform better in success rate, utility and speed – the performance measure indicators. Future works will be on optimizing FDC.

References

1. M. Osborne and A. Rubinstein, Bargaining and Markets, The Academic Press, 1990.
2. Kwang Mong Sim and Shi Yu Wang, Flexible Negotiation Agent with Relaxed Decision Rules, IEEE Trans. On SMC Part B, Vol. 34 No. 3, June 2004.
3. K. M.Sim and K. F. Ng. Relaxed-criteria Negotiation for G-commerce. In Proceedings of the Workshop on Business Agents and the Semantic Web (BASeWEB'06), held in conjunction with the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan, pp, 53-61.
4. T. J. Ross, Fuzzy Logic with Engineering Applications. NewYork: Mc-Graw-Hill, 1995

Creating Service Flows Using Semantic Approach

Kai Kin Chan

Abstract

Service flows connect different services together to form a complex procedure. There are a number of stages for service flows. We should first create a service flow, and then execute the service flow. We give a general approach for creating service flows. For automatic creation, we add semantic on services and flows that help the machines understand the relationships between them.

1 Introduction

Service flows connect a number of services in a procedural order. The service flows comprise a number of logical steps. The system performs creation and execution of service flows. The execution of service flows can be done by some workflows language, such as BPEL. The creation of service flows can divided into Manual Process and Automated Process. Manual Process is that defines and creates service flows manually before execution. Automated Process composes service flows automatically before execution.

In distributed environment, many services are placed in different locations. For different purposes, different service flows are used. Existing service flows can come from background knowledge. For example, ice tea with lemon is a service flow, it is the flow of adding tea service, adding ice service, adding lemon service and adding sugar service.

In this paper, we introduce the service flows composition automatically.

2 Background

In this section, we will introduce some background on creation of scientific. They creation of service flows is similar to scientific workflows. The creation of scientific workflows can be divided into three stages [**?**].

1. To create *workflow templates*. Workflow templates specify the abstract structure of a workflow. It is a high level structure that without identify any particular data and resources.
2. To create *workflow instances*. Workflow instances specify the data and resources used for the workflow template, such as the input data and output data.
3. To create an *executable workflow*. It involves the resources managements and data movements in the distributed services environment.

There are many existing workflow languages, such as BPEL, or Pegasus system [**?**] can perform the third stage of creating the executable workflow.

In this paper, we focus on the first and the second stages. To perform automation of workflow composition, we are using semantic approach.

3 Approach

There is a number of services. Each service may have several inputs and outputs. Figure 1 shows the general structure of ontologies. In our approach, there is a background knowledge ontology that contains service ontology, domain ontology, flow ontology and process ontology.

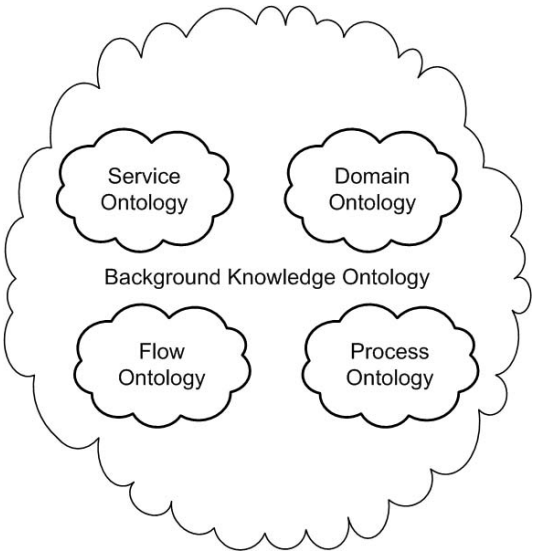


Fig 1. The General Structure of Ontologies

3.1 Creating Service Ontology

The first step is to create a service ontology for each services. The definitions of the service ontology are:

- Name: It represents the name of the service.
- Input: It represents the inputs of the service.
- Output: It represents the outputs of the service.
- Prerequisite: It represents the prerequisite of the service.
- Effect: It represents the effect after executed the service.

The service ontology is needed because the function prototype of each services only provides the basic service name, how many inputs and outputs are needed, and what are the input and output types. The function prototype does not provide any semantic meaning.

In service ontology, the name field used to describe the semantic meaning of the name of the service. The input field and output field are used to provide semantic for inputs and outputs. The prerequisite field specifies what are the preconditions of executing the service. For example, the system must execute the service after executed another service. The effect field specifies the affect to other services or effect in this system after executing the service.

3.2 Creating Domain Ontology

After created the service ontologies, the second step is to create a domain ontology. The domain ontology describe the knowledge structure of a specific area. It is machine-understandable. Using the domain ontology, the system may know the relationship between services, between inputs and outputs.

Figure 2 shows an African wildlife ontology. It is an example of the domain ontology. If there is a workflow with two services: Service A and Service B. Service A outputs Lion, and Service B takes Animal as input. Using the domain ontology, the machine may understand that Lion is a subclass of Animal. As Service A's output type matches Service B's input type, link can be created between Service A and Service B.

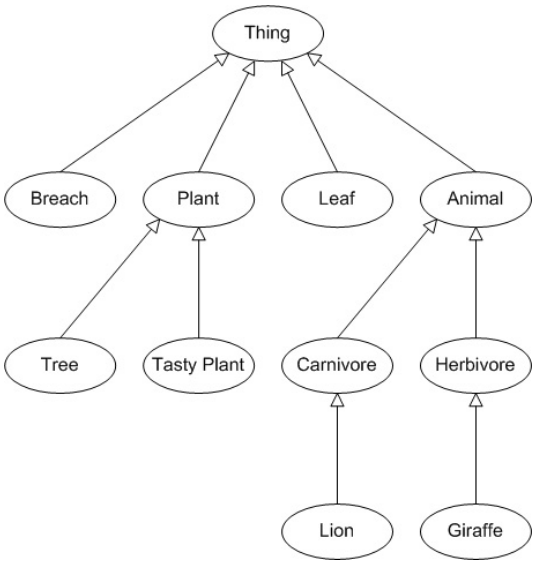


Fig 2. An African Wildlife Ontology

As we have described the semantic meaning of the services, the relationship between services, between inputs and outputs are missing. The domain ontology is used for this purpose.

3.3 Creating Flow Ontology

Making use of domain ontology, a service flow can be composed. Flow Ontology is used for describe a composed service flow semantically.

The definitions of the flow ontology are:

- Name: It represents the name of the flow.
- Services: It represents the services in the flow.
- Links: It represents the links between services.
- Frequency: It represents the frequency used of the flow.

3.4 Creating Process Ontology

If many people construct very similar service flows, we group these service flows together.

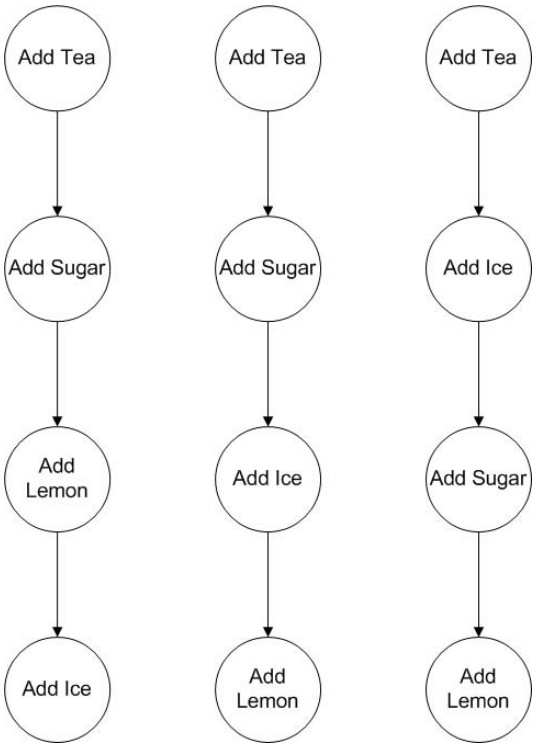


Fig 3. Three Different Service Flows of Making Ice Tea with Lemon

Figure 3 shows three different service flows of making ice tea with lemon. The first flow adds tea first, adds sugar, then adds lemon, and finally adds ice. The second flow adds tea first, adds sugar again, adds ice and then adds lemon. The third flow adds tea first, adds ice, then adds sugar, finally adds lemon. This all belongs to making ice tea with lemon. It is waste memory to store all the service flows. It is better to combine all the service flows of making ice tea with lemon together with one abstract structure. Figure 4 shows the combined service flows of making ice tea with lemon. These service flows are combined to one service flow.

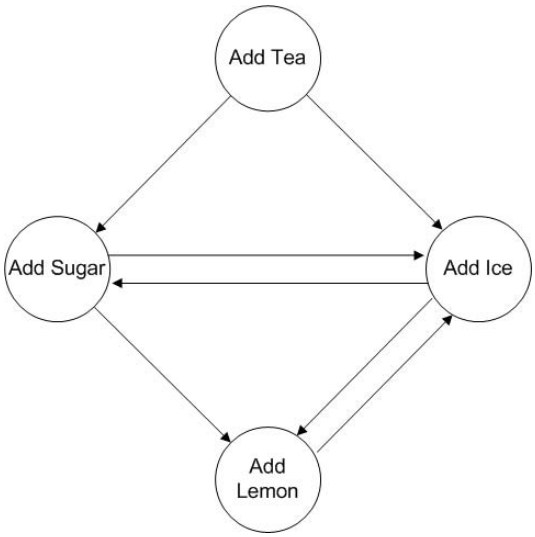


Fig 4. Combined Service Flow of Making Ice Tea with Lemon

The combined service flow are added to process ontology. This combined service flow can also be the workflow template of other service flow.

4 Related Work

There are some related works from Information Sciences Institute, University of Southern California. They created a framework call Wings, based on their execution platform Pegasus, see [**?**, **?**].

5 Conclusions and Future Works

In this paper, we introduced the general idea of creation of service flows. For automated process, we design the system using semantic approach. Using semantic to describe all services, domains and flows. It helps the machine to understand the relationships between services, domains and flows.

In the future works, we will focus on this area, provide more details design and implementation.

References

[1] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping Scientific Workflows onto the Grid. 2004.

[2] Jihie Kim, Yolanda Gil, and Varun Ratnakar. Semantic Meatdata Generation for Large Scientific Workflows. *Proceeding of 5th International Semantic Web Conference, ISWC-2006*, 2006.

[3] Yolanda Gil, Varun Ratnakar, Ewa Deelman, Marc Spraragen, and Jihie Kim. Wings for Pegasus: A Semantic Approach to Creating Very Large Scientific Workflows. *Proceeding of OWL: Experiences and Directions 2006*, 2006.

Data Hiding on 3D Geometry: A Perspective from ICA to Orthogonal Transformation

Hao-tian Wu

Abstract

In the literature, the technique of independent component analysis (ICA) has been used for watermarking of digital images and audio signals. In this paper, it is applied to 3D geometry to improve the security by using the ICA de-mixing matrix for transformation. Then the transformation matrix is replaced by an orthogonal one so that a new data hiding scheme is generated based on orthogonal transformation (OT). By randomly generating an orthogonal matrix whereas avoiding the identity matrix and its permutations, the security of the OT-based data hiding scheme is assured. A reversible algorithm is employed for data embedding in both of the ICA-based and OT-based schemes. In particular, the original geometry can be blindly recovered by improving the precision of the watermarked geometry. The numerical results show that the capacities of the two schemes are the same while the distortions of the watermarked geometries are close to each other. Besides, the OT-based data hiding algorithm performs no worse than the ICA-based one in terms of blind extraction and security, but with less complexity.

1 Introduction

In the field of digital watermarking [1] and information hiding [2], the technique of independent component analysis (ICA) [3]-[4] has been used for digital images (e.g. [6], [7] and [8]) and audio signals (e.g. [9] and [10]). ICA is a versatile technique that was initially used for blind signal separation. It can be represented in the following model by $\mathbf{x} = A\mathbf{s}$, where $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ denotes the observed vector. Each of its elements is the mixture of n independent components $\{s_1, s_2, \ldots, s_n\}$ that form the random vector $\mathbf{s}$. A is the mixing matrix with elements a_{ij} for $i, j \in \{1, 2, \ldots, n\}$ so that $x_i = a_{i1}s_1 + a_{i2}s_2 + \ldots + a_{in}s_n$. All we observe is the vector $\mathbf{x}$, with which we wish to estimate A and $\mathbf{s}$. To make the estimation possible, a simple assumption is that the components s_i are statistically independent over all orders of statistics. Various methods allow ICA estimation to find the mixing matrix A, or its inverse as denoted by W, such as the FastICA algorithm [5]. With the de-mixing matrix W, the independent components can be obtained by $\mathbf{s} = W\mathbf{x}$.

In the literature, the ICA technique has been applied to digital watermarking mainly in two different ways. One is to use the ICA de-mixing matrix for transformation to embed data in the feature space, such as the algorithms for digital images (e.g. [6],[7]) and audio signals (e.g. [9]). If we regard the media data $\mathbf{x}$ as the mixture of statistically independent components $\mathbf{s} = \{s_1, s_2, \ldots, s_n\}$, i.e., $\mathbf{x} = A\mathbf{s}$, we can use the de-mixing matrix W to map $\mathbf{x}$ to the ICA feature space. After that, data embedding can be performed in the feature space by changing the values of the components $\{s_1, s_2, \ldots, s_n\}$ to $\mathbf{t} = \{t_1, t_2, \ldots, t_n\}$. The watermarked media data $\mathbf{y}$ will be obtained after we transform $\mathbf{t}$ with the mixing matrix A, i.e., $\mathbf{y} = A\mathbf{t}$. To extract the embedded data, we need to transform $\mathbf{y}$ back to the ICA feature space by $\mathbf{t} = W\mathbf{y}$. So the de-mixing matrix W acts as a key to both data embedding and extraction and a new form of security is enabled by keeping the matrix W and its inverse secret. In [6], part of the independent components are replaced by those of the watermark image, similar to the method of least significant bits (LSB) replacement. By replacing the components with less energy and weighting the components of the watermark, the perceptual quality of the image can be preserved. However, the data hiding capacity is low at the expense of storing two mixing matrices as the keys, respectively for the original image and the watermark so that the watermarking algorithm is not blind. In a more recent work [7], ICA is used for watermarking of digital images by modulating the independent components with the Quantization Index Modulation (QIM) method [11]. By this means, high data hiding capacity is reachable, especially in the fragile watermarking scheme where a large set of components are selected. In [7], the ICA de-mixing matrix is estimated from a training set of 11 natural scene images so that it is hard to estimate them from the watermarked image. In addition, low computational cost is required by using the pre-computed matrix. The same method is also applied to the audio signals as shown in [9].

Other applications of ICA to digital watermarking are mainly based on the idea of viewing watermark extraction as a problem of blind signals separation. In [8], the key and the watermark, which are special images with the same size as the original image, are linearly mixed with the original image to generate the watermarked image. By regarding the watermarked image as an observed mixture, the watermark recovery can be viewed as a blind signals separation problem and its accuracy depends on the statistical independence between the key, the watermark, and the original image. Since both the key and the original image are required for the watermark extraction, the watermarking algorithm is restricted to private applications such as copyright protection and its robustness to the attacks is examined. The same idea has also been adopted for multimedia authenticity protection. In [10], a covert independent watermark signal serves as a "vaccine" against a dormant digital "bacteria" to protect the multimedia data. Once the unauthorized removal of the watermark is detected after blind de-mixing with ICA, a program is triggered to take appropriate action, such as degrading of quality. The robustness of the ICA digital watermarking with respect to lossy compression, such as the compatibility with the popular MP3 format for digital music files, has been measured. The dynamic range of the digital watermark as clutter versus the original music, which is a signal coded under the MP3 format, has also been investigated.

In this paper, we concentrate on the method of using the ICA de-mixing matrix for transformation to improve the security of data hiding on 3D geometry. To make it suitable for blind watermarking and reduce the complexity as well, the de-mixing matrix is not estimated from the original geometry, but from a set of geometries other than it. The popular FastICA algorithm is employed to estimate the ICA mixing and de-mixing matrices. To make the transformation matrix independent from the media content, a randomly generated orthogonal matrix is further used instead of the ICA de-mixing matrix. Since the ICA matrix is also orthogonal after the whitening preprocessing in the estimation process, the requirement on the transformation matrix is actually relaxed. And the data hiding scheme via orthogonal transformation (OT) may have some properties as same as the ICA-based one. For instance, the capacity will not be changed if the ICA de-mixing matrix is replaced by another orthogonal matrix. Note that an orthogonal transformation is a linear transformation that preserves the inner product of two vectors. In particular, both the lengths of the vectors and the angles between them are preserved by orthogonal transformation so that the shape of 3D geometry is preserved as well. Therefore, the distortion of 3D shape caused by data embedding in the transform domain is exactly that in the spatial domain. In other words, the distortion introduced to 3D geometry by the OT-based or ICA-based data hiding scheme can be directly controlled in the transform domain. A reversible algorithm is employed for data embedding to generate the OT-based or ICA-based algorithms for 3D geometry, respectively. To guarantee the exact recovery of the original geometry, every coordinate in the watermarked geometry should be stored with one more decimal digit. The two generated algorithms are applied to some testing models to compare their performances in terms of imperceptibility, capacity and complexity. As for security and blind detection, some comments are given after comparing the characteristics of the two algorithms.

The rest of this paper is organized as follows: In the next section, the principles of data hiding on 3D geometry using ICA and orthogonal transformation will be introduced, respectively. In Section III, a reversible algorithm will be employed in both of them to generate the ICA-based and OT-based algorithms for 3D geometry, respectively. The generated algorithms will be applied to some testing models to compare their performances in Section VI. Finally, some concluding remarks are drawn in Section V.

2 Two Data Hiding Schemes for 3D Geometry

In the literature, a lot of watermarking algorithms (e.g. [12]-[19]) have been proposed for 3D geometries, polygonal meshes in particular. Many transformation tools have been utilized to embed data into polygonal meshes in the transform domain. Typical methods include the construction of a set of basis functions [13] to employ the spread spectrum approach [20], multi-resolution wavelet decomposition [16], principal component analysis [17], etc. Those algorithms try to improve the robustness of the embedded watermark, especially without the presence of the original mesh. In contrast, we aim to improve the security of data hiding on 3D geometry by using ICA and orthogonal transformation, respectively.

2.1 The Data Hiding Scheme Using ICA

Although the ICA-based data hiding scheme has been applied to digital images and audio signals, how to apply it to 3D geometry has not been previously reported. Given a 3D geometry consists of N position vectors, the geometry can be represented by $\mathbf{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_N}\}$, where a position vector $\mathbf{p_i}$ specifies the coordinates $\{p_{ix}, p_{iy}, p_{iz}\}$ in the 3D space R^3 for $i = 1, 2, \ldots, N$. For a position vector $\mathbf{p_i}$, the coordinates $\{p_{ix}, p_{iy}, p_{iz}\}$ within it can be regarded as a linear combination of three statistically independent sources

$\mathbf{s_i} = \{s_{i1}, s_{i2}, s_{i3}\}$, i.e.

$$\mathbf{p_i} = \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{pmatrix} = A \times \mathbf{s_i} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \end{pmatrix}. \quad (1)$$

By using the ICA algorithm such as FastICA, we can estimate the mixing matrix A and its inverse, i.e. the de-mixing matrix W, from a set of the chosen geometries. If we directly estimate A from the set of position vectors $\mathbf{P}$ in the original geometry, the obtained matrix will be dependent on the original geometry so that the de-mixing matrix W estimated from each geometry is different. As a result, the data extraction will be non-blind so that it cannot be used in blind watermarking. In contrast, the de-mixing matrix W should be estimated from a set of geometries other than the original one. In this way, low computational cost is required because the only operation to be carried out for transformation is a simple matrix multiplication. With the de-mixing matrix W, every position vector can be transformed to the independent components by $\mathbf{s_i} = W\mathbf{p_i}$, i.e.

$$\mathbf{s_i} = \begin{pmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \end{pmatrix} = W \times \mathbf{p_i} = A^{-1} \times \begin{pmatrix} p_{ix} \\ p_{ix} \\ p_{iy} \end{pmatrix}. \quad (2)$$

Although the obtained components will not be as independent as possible, the proposed scheme can be applied to the blind watermarking of 3D geometry with less complexity.

Data embedding can be performed in the ICA feature space by changing the component vector $\mathbf{s_i}$ to a new one $\mathbf{t_i} = \{t_{i1}, t_{i2}, t_{i3}\}$. The position vector $\mathbf{g_i}$ in the watermarked geometry can be generated by

$$\mathbf{g_i} = \begin{pmatrix} g_{ix} \\ g_{iy} \\ g_{iz} \end{pmatrix} = A \times \mathbf{t_i} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} t_{i1} \\ t_{i2} \\ t_{i3} \end{pmatrix}. \quad (3)$$

The whole watermarked geometry $\mathbf{G} = \{\mathbf{g_1}, \ldots, \mathbf{g_N}\}$ will be formed after every position vector in it has been generated.

2.2 The Data Hiding Scheme via Orthogonal Transformation

In the ICA-based data hiding scheme, the de-mixing matrix W is estimated from a set of the chosen geometries to transform the original geometry to the ICA feature space. Here we relax the requirement by performing the transformation with an orthogonal matrix independent from 3D geometry. As a result, the generation of the transformation matrix is greatly simplified while the new scheme is suitable for blind watermarking of 3D geometry. For an orthogonal matrix Q, its transpose Q^T is just its inverse:

$$QQ^T = I, \quad (4)$$

where I is the identity matrix. An orthogonal matrix Q corresponds to an orthogonal transformation, which is a linear transformation that preserves the inner product of any two vectors in the Euclidean space R^n, which is considered here for the 3D space R^3 is a typical example of it.

Orthogonal transformation of 3D geometry is performed by multiplying every vector $\mathbf{p_i}$ with a 3×3 orthogonal matrix B so that a new vector $\mathbf{f_i} = (f_{ix}, f_{iy}, f_{iz})^T$ in R^3 is obtained by

$$\mathbf{f_i} = \begin{pmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{pmatrix} = B \times \mathbf{p_i} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \times \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{pmatrix}. \quad (5)$$

Data embedding is performed in the transform domain by changing $\mathbf{f_i}$ to $\mathbf{k_i} = (k_{ix}, k_{iy}, k_{iz})^T$ for $i = 1, 2, \ldots, N$. The watermarked geometry can be therefore obtained by multiplying $\mathbf{k_i}$ with B^T so that a new set of position vectors $\{\mathbf{g_1}, \ldots, \mathbf{g_N}\}$ are generated, respectively by

$$\mathbf{g_i} = \begin{pmatrix} g_{ix} \\ g_{iy} \\ g_{iz} \end{pmatrix} = B^T \times \mathbf{k_i} = \begin{pmatrix} b_{11} & b_{21} & b_{31} \\ b_{12} & b_{22} & b_{32} \\ b_{13} & b_{23} & b_{33} \end{pmatrix} \times \begin{pmatrix} k_{ix} \\ k_{iy} \\ k_{iz} \end{pmatrix}. \quad (6)$$

Suppose $\mathbf{k_i} = \mathbf{f_i} + \delta_\mathbf{i}$ where $\delta_\mathbf{i} = (\delta_{ix}, \delta_{iy}, \delta_{iz})^T$ is the change of $\mathbf{f_i}$ caused by data embedding. It can be seen that the difference between $\mathbf{g_i}$ and $\mathbf{p_i}$ is $B^T\delta_\mathbf{i}$. Since B^T is also an orthogonal matrix, the length of $B^T\delta_\mathbf{i}$ is always equal to that of $\delta_\mathbf{i}$, as denoted by

$$|B^T\delta_\mathbf{i}| = |\delta_\mathbf{i}|. \quad (7)$$

As a matter of fact, both the lengths of the vectors and the angles between them are preserved by orthogonal transformation so that the shape of 3D geometry is preserved as well. Therefore, the distortion of 3D shape introduced by data embedding in the transform domain is exactly that in the spatial domain so that it can be directly controlled. In some special cases, the security of data embedding cannot be improved via orthogonal transformation. For example, if the identity matrix is used, there is no difference before and after the transformation so that the security level is the same. When the permutations of the identity matrix are used for transformation, the security cannot be improved either. In practice, an orthogonal matrix is randomly generated whereas avoiding those special ones. To perform the task of data embedding, a reversible algorithm will be introduced in the following section.

3 Employing A Reversible Algorithm in the Schemes

3.1 A Reversible Data Hiding Algorithm

In the following, a reversible data hiding algorithm will be introduced by adopting the idea of reserving the modulation information in the watermarked object. Different from the reversible algorithms proposed for digital images (e.g. [21]-[23]), the following one is applicable to the media content represented by numbers with decimal point, such as 3D coordinates in the geometrical models. Note that a string of binary numbers $\mathbf{W} = \{w_1, w_2, \ldots, w_N\}$ will be embedded into a set of one-dimensional signals $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_N\}$. The embedding process is based on the QIM scheme while the modulation information is further reserved in the watermarked signals, as denoted by $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_N\}$.

There are two parts of information that should be contained in the watermarked signals $\{Z_1, Z_2, \ldots, Z_N\}$. One part is the embedded values $\mathbf{W}$, and the other part is the modulation information, which is defined as the differences between the original signals $\{Y_1, Y_2, \ldots, Y_N\}$ and a set of special values $\{Y_1', Y_2', \ldots, Y_N'\}$ generated from $\{Y_1, Y_2, \ldots, Y_N\}$, $\{w_1, w_2, \ldots, w_N\}$ and the quantization step Δ. A set of variables $\mathbf{E} = \{e_1, e_2, \ldots, e_N\}$ are further defined with a parameter a to represent the modulation information by $e_i = \frac{Y_i' - Y_i}{a}$ for $i = 1, \cdots, N$. The procedure to hide the two parts of information is as follows:

Step 1: For $i = 1, \cdots, N$, initialize the integer quotient Q_i by $Q_i = \lfloor \frac{Y_i}{\Delta} \rfloor$, where $\lfloor \cdot \rfloor$ represents the low floor. A binarization function $B(\cdot)$ is defined by $B(Int) = Int - \lfloor \frac{Int}{2} \rfloor \times 2$, where the input Int is an integer and the output $B(Int)$ is a binary number. For example, if Int equals to -7, then $\lfloor \frac{Int}{2} \rfloor = -4$ so that $B(-7) = 1$. Then $B(Q_i)$ and the remainder R_i are further calculated by

$$\begin{cases} B(Q_i) = Q_i - \lfloor \frac{Q_i}{2} \rfloor \times 2 \\ R_i = Y_i - Q_i \times \Delta \end{cases}. \tag{8}$$

It can be seen from the definition of Q_i that R_i is nonnegative.

Step 2: A binary number w_i is embedded in Y_i by

$$Y_i' = \begin{cases} Q_i \times \Delta + \frac{\Delta}{2} & \text{if } B(Q_i) = w_i \\ Q_i \times \Delta - \frac{\Delta}{2} & \text{if } B(Q_i) \neq w_i \ \& \ R_i \leq \frac{\Delta}{2} \\ Q_i \times \Delta + \frac{3\Delta}{2} & \text{if } B(Q_i) \neq w_i \ \& \ R_i > \frac{\Delta}{2} \end{cases} \tag{9}$$

so that $Y_i' = \lfloor \frac{Y_i'}{\Delta} \rfloor \times \Delta + \frac{\Delta}{2}$ and $B(\lfloor \frac{Y_i'}{\Delta} \rfloor) = w_i$. Subsequently, $e_i = \frac{Y_i' - Y_i}{a}$ is obtained.

Step 3: For $i = 1, \cdots, N$, the value of e_i is further added to Y_i' by

$$Z_i = Y_i' + e_i = \lfloor \frac{Y_i'}{\Delta} \rfloor \times \Delta + \frac{\Delta}{2} + e_i. \tag{10}$$

If the difference between Y_i' and Y_i is denoted by γ_i, it can be seen from Eq.(9) that $\gamma_i \in [-\Delta, \Delta)$. Since $e_i = \frac{\gamma_i}{a}$ for $i = 1, \cdots, N$, the variables $\mathbf{E} = \{e_1, e_2, \ldots, e_N\}$ will be distributed within $[-\frac{\Delta}{a}, \frac{\Delta}{a})$. Therefore, the parameter a can be assigned with a value greater than 2 so that $e_i \in (-\frac{\Delta}{2}, \frac{\Delta}{2})$ for $i = 1, \cdots, N$. Thus the adding of e_i in Eq.(10) will not change the embedded value w_i, i.e. $B(\lfloor \frac{Z_i}{\Delta} \rfloor) = B(\lfloor \frac{Y_i'}{\Delta} \rfloor) = w_i$.

The values of the quantization step Δ and the parameter a used in the embedding process are required to extract the embedded data and recover the original signals from the watermarked signals $\{Z_1, Z_2, \ldots, Z_N\}$. With the quantization step Δ, the embedded binary numbers $\{w_1, w_2, \ldots, w_N\}$ can be extracted from $\{Z_1, Z_2, \ldots, Z_N\}$ by

$$w_i = B(\lfloor \frac{Z_i}{\Delta} \rfloor). \tag{11}$$

In practice, the precision of the watermarked signals needs to be taken into account. Suppose the watermarked signal Z_i is stored at the precision level of 10^{-m}, the roundoff error that happens to Z_i is within $(-5 \times 10^{-(m+1)}, 5 \times 10^{-(m+1)}]$. To ensure that the value of $\lfloor \frac{Z_i}{\Delta} \rfloor$ is not affected by the limited precision so that the value of w_i can be correctly extracted, the following condition should be satisfied

$$0 \leq \frac{\Delta}{2} + e_i \pm 5 \times 10^{-(m+1)} < \Delta, \tag{12}$$

which is equivalent to $\frac{\Delta}{2} - \frac{\Delta}{a} \geq 5 \times 10^{-(m+1)}$ since $e_i \in [-\frac{\Delta}{a}, \frac{\Delta}{a})$. It can be seen that the parameter a should be assigned with a value greater than 2 because the quantization step Δ must be a positive value. Given $a > 2$, the embedded data can be correctly extracted if

$$\Delta \geq \frac{a}{a-2} 10^{-m}, \tag{13}$$

where 10^{-m} is the precision level of the watermarked signals.

To recover the original signals $\{Y_1, Y_2, \ldots, Y_N\}$, the reserved modulation information represented by $\mathbf{E} = \{e_1, e_2, \ldots, e_N\}$ should be retrieved from the set of watermarked signals $\{Z_1, Z_2, \ldots, Z_N\}$. For $i = 1, \cdots, N$,

$$e_i = Z_i - (\lfloor \frac{Z_i}{\Delta} \rfloor \times \Delta + \frac{\Delta}{2}) = Z_i - Y_i', \tag{14}$$

where $Y_i' = \lfloor \frac{Z_i}{\Delta} \rfloor \times \Delta + \frac{\Delta}{2}$ can be easily obtained from Z_i and the provided quantization step Δ. For $i = 1, \cdots, N$, the original signal Y_i can be obtained by

$$Y_i = Y_i' - e_i \times a. \tag{15}$$

However, the recovered signal only approximates to the original one due to the limited precision. Without

loss of generality, we suppose that the original signals $\{Y_1, Y_2, \ldots, Y_N\}$ is at the precision of 10^{-n}. When the watermarked signal Z_i is also stored at the precision level of 10^{-n}, the roundoff error that happens to the reserved modulation information e_i is within $(-5 \times 10^{-(n+1)}, 5 \times 10^{-(n+1)}]$. Given Eq.(13), the value of Y_i' can be correctly generated by $\lfloor \frac{Z_i}{\Delta} \rfloor \times \Delta + \frac{\Delta}{2}$ with the provided quantization step Δ. Then the error introduced to the recovered signal will be within $(-5a \times 10^{-(n+1)}, 5a \times 10^{-(n+1)}]$ as shown in Eq. (15). Since the parameter a should be assigned with a value no less than 2, the difference between the recovered and original signals may be out of the range of $(-10^{-n}, 10^{-n})$. So the recovered signal may be unequal to the original one if the watermarked signal is at the same precision level as the original one. To overcome the round-off error so that the original signal can be exactly recovered, the precision of the watermarked signals should be improved.

3.2 The Generated Algorithms

One may argue the security of the reversible algorithm is poor, because the embedded data can be easily extracted or modified once the quantization step is known. Actually, it can be combined with the ICA-based or the OT-based scheme for 3D geometry in Section II to enhance the security. Note that the difference between the two schemes lies in the generation of the transformation matrix: one is an orthogonal matrix except the identity matrix and its permutations, while the other is the de-mixing matrix estimated from a set of chosen geometries by the ICA algorithm. In the following, only the generation of the OT-based algorithm will be detailed by employing the reversible algorithm, for the principles and steps to generate the ICA-based one are the same. As the coordinates in the 3D geometry are represented by numbers with decimal point, every of them can be stored with one more decimal digit without necessarily changing the data format so that the original geometry can be exactly recovered by assigning the appropriate values to the parameters.

Given a set of N position vectors $\{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_N}\}$ in a 3D geometry, a new set of N vectors $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_N}\}$ will be generated after multiplying every position vector with an orthogonal matrix B. Since a vector $\mathbf{f_i}$ consists of three elements, i.e. f_{ix}, f_{iy} and f_{iz}, the reversible data embedding algorithm can be applied to three sets of elements $\{f_{1x}, f_{2x}, \ldots, f_{Nx}\}$, $\{f_{1y}, f_{2y}, \ldots, f_{Ny}\}$ and $\{f_{1z}, f_{2z}, \ldots, f_{Nz}\}$ with the same quantization step Δ and parameter a as shown in Section III. A, respectively. In total, 3 strings of binary numbers with the length of N can be embedded by modulating $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_N}\}$ to $\{\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_N}\}$. The watermarked geometry is obtained by multiplying every vector in $\{\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_N}\}$ with B^T to generate a new set of position vectors $\{\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_N}\}$. Suppose the coordinates in the original geometry are stored at the precision level of 10^{-n}. Then every coordinate in the watermarked one should be stored at the precision level of $10^{-(n+1)}$ to guarantee the exact recovery.

To extract data from the position vectors $\{\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_N}\}$ in the watermarked geometry, we need to transform them with B to generate every vector in the set of $\{\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_N}\}$, respectively. With the quantization step Δ used in the embedding process, three strings of binary numbers can be retrieved from $\{k_{1x}, k_{2x}, \ldots, k_{Nx}\}$, $\{k_{1y}, k_{2y}, \ldots, k_{Ny}\}$ and $\{k_{1z}, k_{2z}, \ldots, k_{Nz}\}$, respectively. Besides, three sets of elements $\{f_{1x}, f_{2x}, \ldots, f_{Nx}\}$, $\{f_{1y}, f_{2y}, \ldots, f_{Ny}\}$ and $\{f_{1z}, f_{2z}, \ldots, f_{Nz}\}$ can be respectively recovered from $\{k_{1x}, k_{2x}, \ldots, k_{Nx}\}$, $\{k_{1y}, k_{2y}, \ldots, k_{Ny}\}$ and $\{k_{1z}, k_{2z}, \ldots, k_{Nz}\}$ with the quantization step Δ and the parameter a used in the embedding process, as shown in Section III. B. The original geometry is recovered by multiplying every vector in the set of $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_N}\}$ with B^T to generate the original position vectors $\{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_N}\}$.

If the coordinates in the watermarked geometry are stored at the precision level of $10^{-(n+1)}$, the roundoff error happens to every of them is within $(-5 \times 10^{-(n+2)}, 5 \times 10^{-(n+2)}]$. After the orthogonal transformation, which can be viewed as a rotation or a rotation followed by a flip, the error vector introduced to the vectors in the transform domain is within the sphere with the crossing of the axes as the centroid and $5\sqrt{3} \times 10^{-(n+2)}$ as the radius. So the error introduced to every element among the vector set of $\{\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_N}\}$ is within $(-5\sqrt{3} \times 10^{-(n+2)}, 5\sqrt{3} \times 10^{-(n+2)}]$. To ensure the embedded data can be correctly extracted from the vectors in the transform domain, it should be satisfied that

$$\frac{\Delta}{2} - \frac{\Delta}{a} < 5\sqrt{3} \times 10^{-(n+2)}, \tag{16}$$

i.e. $\Delta \geq \frac{\sqrt{3}a}{a-2} 10^{-(n+1)}$, where $10^{-(n+1)}$ is the precision level of the coordinates in the watermarked geometry.

Suppose the roundoff error happens to a position vector $\mathbf{g_i}$ in the watermarked geometry is denoted by $\mathbf{d_i} = \{d_{ix}, d_{iy}, d_{iz}\}$ for $i = 1, 2, \ldots, N$. After transforming $\mathbf{g_i}$ with B to generate a vector $\mathbf{k_i}$, the error introduced to $\mathbf{k_i}$ is $B\mathbf{d_i}$. The error vector is multiplied by a when recovering $\mathbf{f_i}$ from $\mathbf{k_i}$, i.e. $aB\mathbf{d_i}$. At last, the error introduced to the position vector $\mathbf{p_i}$ is $B^T aB\mathbf{d_i} = a\mathbf{d_i}$ after transforming $\mathbf{f_i}$ with B^T. Since each element in $\mathbf{d_i}$ is within $(-5 \times 10^{-(n+2)}, 5 \times 10^{-(n+2)}]$, the error introduced to each coordinate in the recovered geometry will be in $(-5a \times 10^{-(n+2)}, 5a \times 10^{-(n+2)}]$. By assigning a value between 2 and 10 to the parameter a and choosing the quantization step Δ according to Eq.(16), every coordinate in the recovered geometry will be identical to that in the original geometry at the precision level of 10^{-n}. Thus far, a re-

Table 1. THE VRML MODELS USED IN THE EXPERIMENTS

Model	Vertices	Polygons	Capacity(bits)
lamp	676	1288	2028
pear	891	1704	2673
sgilogo	1224	1620	3672
pavilion	7334	5338	22002
indigo	8389	10187	25167
gears	24546	8182	73638

versible data hiding algorithm has been generated for 3D geometry via orthogonal transformation.

Similarly, the ICA-based data hiding algorithm is acquirable by replacing the orthogonal matrix B and its inverse with the ICA de-mixing matrix W and the mixing matrix A, respectively. According to the symbols used in Section II. A, the sets of vectors $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_N}\}$ and $\{\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_N}\}$ in the OT-based algorithm should also be replaced with the components $\{\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_N}\}$ and $\{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_N}\}$ to generate the ICA-based algorithm, respectively. Obviously, the two generated algorithms are applicable to 3D geometry without any constraint. In the following, the experimental results implementing them on some testing models will be given to compare their performances.

4 Performance Analysis & Comparison

The two generated algorithms were implemented on several VRML models listed in Table I, where the capacity is given. A string of binary numbers were randomly generated as the watermark so that its length can be chosen according to the capacity of each model. As for the OT-based algorithm, an orthogonal matrix B was randomly generated, avoiding the identity matrix and its permutations. As for the ICA-based algorithm, the de-mixing matrix W was estimated from a set of geometries other than the original one.

In the experiments, the coordinates in the original geometry were at the precision level of 10^{-6}, while the coordinates in the watermarked mesh model were set at the precision level of 10^{-7}. If the coordinates are stored in floating point numbers, the file size of the watermarked geometry will not be increased by the precision improvement. It should be noted that the distortion introduced by watermark embedding can still be reduced by the recovery process if the precision of the watermarked geometry cannot be improved. Since a value between 2 and 10 should be assigned to the parameter a, we chose 2.1 as its value so that the quantization step Δ should be no less than $\frac{2.1\sqrt{3}}{2.1-2} \times 10^{-7} \simeq 3.64 \times 10^{-6}$ as required in Eq. (16). By setting the quantization step $\Delta = 0.000005$ and the parameter $a = 2.1$, the pictures rendered from the original, watermarked and recovered models of "sgilogo" and "gears" using the OT-based data hiding algorithm are shown in Fig. 1.

4.1 Imperceptibility & Reversibility

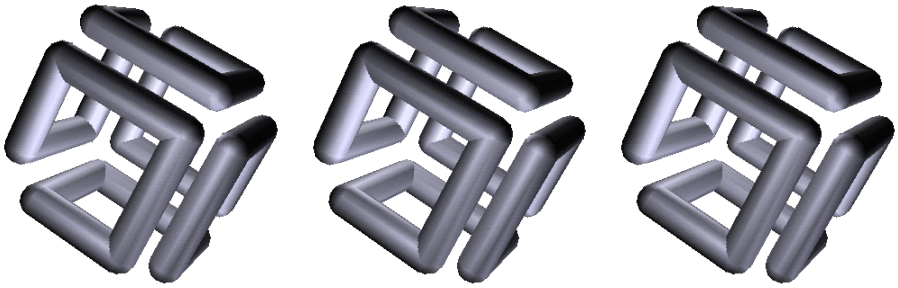
To represent the geometrical distortion of the mesh content, the 3D signal-to-noise ratio (3D SNR) is defined as follows: Given L vertices in a 3D mesh model, the vertex positions in the original mesh are represented by $\mathbf{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_L}\}$, while the vertex positions in the watermarked mesh are denoted as $\mathbf{G} = \{\mathbf{g_1}, \ldots, \mathbf{g_L}\}$. By using the mean square function $MS(\cdot)$, the 3D SNR is defined by

$$SNR = 10 \log_{10} \frac{MS(\mathbf{P} - \overline{\mathbf{P}})}{MS(\mathbf{G} - \mathbf{P})}, \tag{17}$$

where $\overline{\mathbf{P}} = \{\overline{p}_x, \overline{p}_y, \overline{p}_z\}$ is the mean of $\mathbf{P}$ so that $\mathbf{P} - \overline{\mathbf{P}}$ denotes the signal, whereas $\mathbf{G} - \mathbf{P} = \mathbf{G} - \overline{\mathbf{P}} - (\mathbf{P} - \overline{\mathbf{P}})$ represents the noise. In particular,

$$\begin{cases} MS(\mathbf{P} - \overline{\mathbf{P}}) = \frac{\sum_{i=1}^{N} (p_{ix} - \overline{p}_x)^2 + (p_{iy} - \overline{p}_y)^2 + (p_{iz} - \overline{p}_z)^2}{N} \\ MS(\mathbf{G} - \mathbf{P}) = \frac{\sum_{i=1}^{N} (g_{ix} - p_{ix})^2 + (g_{iy} - p_{iy})^2 + (g_{iz} - p_{iz})^2}{N} \end{cases}. \tag{18}$$

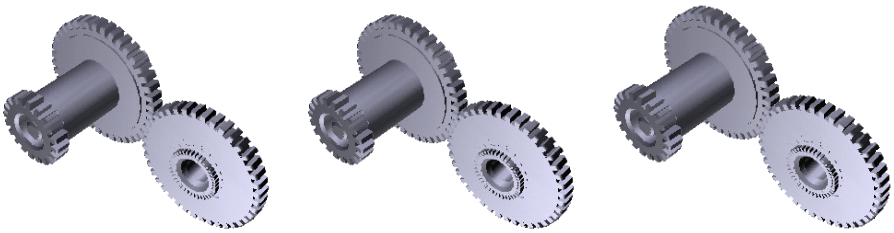
In the experiments, the impact of data embedding could be tuned by the quantization step Δ. In the OT-based algorithm, if 0.0001 and 0.00001 were assigned to Δ, the obtained 3D SNRs of the watermarked geometry "pear" were about 52.78 and 72.60, respectively. As shown in Fig. 2 (a), the 3D SNR of the watermarked geometry decreases if the quantization step Δ is increased by setting the parameter $a = 2.1$. The difference in 3D SNR between the two watermarked geometries generated by the OT-based and ICA-based algorithms is shown in Fig. 2 (b). It can be seen from Fig. 2 that the distortions of the watermarked geometries in the OT-based and ICA-based algorithms were close to each other, because the difference in 3D SNR between the two watermarked geometries fluctuated around zero and was relatively small. The 3D SNR of the recovered geometry was calculated in the same way to test the reversibility of the OT-based and ICA-based algorithms. By setting the parameter $a = 2.1$ and the quantization step $\Delta = 0.000005$, the obtained SNR values of the recovered geometries in both of the OT-based and ICA-based algorithms were *infinite* if the precisions of the watermarked geometries had been improved.



(a) The original mesh model of "sgilogo"

(b) The watermarked mesh with its precision improved

(c) The recovered mesh model

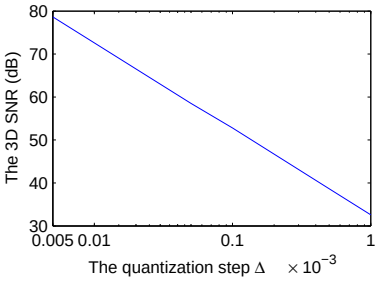


(d) The original mesh model of "gears"

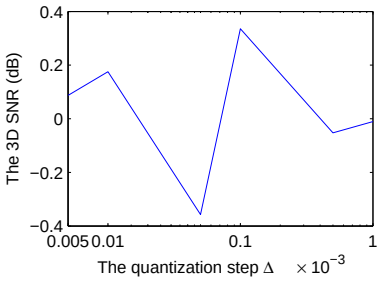
(e) The watermarked mesh with its precision improved

(f) The recovered mesh model

Figure 1. The original mesh models of "sgilogo" and "gears", the watermarked ones generated by the OT-based data hiding algorithm with $\Delta = 0.000005$ and $a = 2.1$, and the mesh models recovered from the watermarked ones, respectively.



(a) The 3D SNR of the watermarked geometry generated by the OT-based algorithm



(b) The difference in 3D SNR between the two watermarked geometries generated by the OT-based and ICA-based algorithms, respectively

Figure 2. By setting the parameter a at 2.1, the 3D SNRs of the two watermarked geometries "pear" generated by the OT-based and ICA-based data hiding algorithms vary with the quantization step Δ, respectively.

4.2 Data Hiding Capacity

Since a position vector has three coordinates in X, Y and Z axes, respectively, three binary numbers can be embedded in it. Given N position vectors in a 3D geometry, such as N vertices in a polygonal mesh or N points in a point cloud, the data hiding capacities of the OT-based and ICA-based data hiding algorithms are both $3N$ bits, higher than the previous works such as $3N - 2$ bits in [15] and $N - 1$ bits in [19]. Different from the algorithm in [15], the upper bound of capacity is reachable for any polygonal mesh, despite whether it is manifold or not.

4.3 Complexity & Blind Extraction

Compared with estimating the ICA de-mixing matrix from a set of 3D geometries, it is much simpler to randomly generate an orthogonal matrix whereas avoiding the identity matrix and its permutations. Since the difference between the OT-based and the ICA-based data hiding algorithms lies only in the generation of the transformation matrix, the OT-based algorithm has less complexity than the ICA-based algorithm. In the OT-based algorithm, the randomly generated orthogonal matrix is independent from the original geometry so that it can be used for any geometry. In addition, no information regarding the original geometry is required to extract the embedded data from the watermarked geometry. In the ICA-based algorithm, the ICA de-mixing matrix can also be used for blind extraction applications if it is estimated from a set of geometries other than the original one, although the obtained components will not be as independent as possible.

4.4 Security

Security is an important aspect of digital watermarking in that the embedded data cannot be extracted or modified without authorization. The common features of the OT-based and ICA-based data hiding algorithms in security can be summarized as follows. (1). By transforming the media data with a invertible matrix, it is hard for an opponent to access the transform domain by keeping the transformation matrix secret. As shown in the experiments, the extracted values were quite different from the embedded ones when the extraction process was performed without transforming the watermarked geometry. (2). By setting the parameter a at 2.1, the remainders after dividing the elements of the vectors in the transform domain by the quantization step Δ are distributed in $(0, \Delta)$. Therefore, it is hard to estimate the quantization step Δ from the watermarked geometry. (3). The security of the OT-based and ICA-based algorithms can be further enhanced by scrambling the order of vectors in the transform domain with a secret key. Without the correct order, one cannot correctly extract the embedded watermark, given the accurate quantization step. In summary, it is hard for an opponent to detect or extract the embedded data from the watermarked geometry without the transformation matrix and the quantization step, given the specific watermarking algorithm.

The difference between the OT-based and ICA-based data hiding algorithms lies in the generation of the matrix used for transformation. In the OT-based algorithm, the orthogonal matrix is randomly generated so that it is independent from the 3D geometry. Thus, it is hard to infer the orthogonal matrix used for transformation from the watermarked geometry. As for the ICA-based algorithm, it is also hard to infer the de-mixing matrix, which is estimated from a set of geometries other than the original one. But the de-mixing matrix may be disclosed if the set of the chosen geometries are known to an opponent. Therefore, the security of the ICA-based data hiding algorithm is no better than that of the OT-based one.

5 Conclusion

In this paper, the technique of ICA has been applied to data hiding on 3D geometry. By transforming the position vectors with the de-mixing matrix estimated from a set of the chosen geometries, data embedding and extraction can be performed in the transform domain. We have further relaxed the requirement on the transformation matrix by using an orthogonal matrix instead so that a new data hiding scheme has been generated based on orthogonal transformation (OT). To improve the security, the orthogonal matrix is randomly generated while avoiding the identity matrix and its permutations. A reversible algorithm has been employed for data embedding in both of the ICA-based and OT-based schemes.

The two generated algorithms have been implemented on several VRML models to compare their performances. The numerical results have shown that the capacities of the two algorithms are the same while the distortions of the watermarked geometries in the two algorithms are close to each other. Although the OT-based data hiding algorithm is much simpler, it performs no worse than the ICA-based one in terms of blind extraction and security. It can be seen that the OT-based data hiding algorithm is more suitable for the blind watermarking of the 3D geometrical models.

Acknowledgment

The VRML models are from http://www.martinreddy.net/ukvrsig/vrml.html.

References

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, "Digital Watermarking," *New York: Morgan Kaufmann*, 2001.

[2] S. Katzenbeisser and F. A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking," *Artech House*, 2000.

[3] P. Comon, "Independent Component Analysis - A New Concept?" *Signal Processing*, Vol. 36, pp. 287-314, 1994.

[4] A. Hyvarinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, pp. 411-430, 2000.

[5] A. Hyvarinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, Vol.9, pp. 1483-1492, 1997.

[6] F. J. Gonz'alez-Serrano, H. Y. Molina-Bulla, and J. J. Murillo-Fuentes, "Independent component analysis applied to digital watermarking," *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Vol.3, pp. 1997-2000, 2001.

[7] S. Bounkong, B. Toch, D. Saad, D. Lowe, "ICA for Watermarking Digital Images," *Journal of Machine Learning Research*, Vol. 4, pp. 1471-1498, 2003.

[8] D. Yu, F. Sattar, and K.-K. Ma, "Watermark detection and extraction using independent component analysis method," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 92-104, 2002.

[9] B. Toch, D. Lowe, and D. Saad, "Watermarking of audio signals using independent component analysis," *International conference on WEB delivering of music*, pp. 71-74, 2003.

[10] H. Szu, S. Noel, S.-B. Yim, J. Willey, and J. Landa, "Multimedia authenticity protection with ICA watermarking and digital bacteria vaccination," *Neural Networks, special issue for International Joint Conference on Neural Networks*, 2003.

[11] B. Chen and G. W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Transactions on Information Theory*, vol. 47, pp. 1423-1443, May 2001.

[12] O. Benedens, "Geometry-Based Watermarking of 3-D Models," *IEEE Computer Graphics and Application, Special Issue on Image Security*, pp. 46-55, Jan./Feb. 1999.

[13] E. Praun, H. Hoppe and A. Finkelstein, "Robust Mesh Watermarking," *Proceedings of ACM SIGGRAPH*, pp. 69-76, 1999.

[14] B. L. Yeo and M. M. Yeung, "Watermarking 3-D Objects for Verification," *IEEE Computer Graphics and Application*, pp. 36-45, Jan./Feb. 1999.

[15] F. Cayre, O. Devillers, F. Schmitt, and H. Maitre, "Watermarking 3D Triangle Meshes for Authentication and Integrity," *Research Report RR-5223, INRIA*, 2004.

[16] F. Uccheddu, M. Corsini and M. Barni, "Wavelet-Based Blind Watermarking of 3d Models," *Proceedings of ACM Multimedia & Security Workshop*, pp. 143-154, Magdeburg, Germany, 2004.

[17] S. Zafeiriou, A. Tefas and I. Pitas, "Blind Robust Watermarking Schemes for Copyright Protection of 3D Mesh Objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596-607, Sept/Oct, 2005.

[18] A. G. Bors, "Watermarking Mesh-Based Representations of 3-D Objects Using Local Moments," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 687- 701, 2006.

[19] H. T. Wu and Y. M. Cheung, "A High-Capacity Data Hiding Method for Polygonal Meshes," *Proceedings of the 8th Information Hiding International Conference (IH'2006)*, Virginia, USA, July, 2006.

[20] I. J. Cox, J. Killian, T. Leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, Vol. 12, pp. 1673C1687, 1997.

[21] C. De Vleeschouwer, J. E. Delaigle and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Transactions on Multimedia*, vol. 5, pp. 97-105, 2003.

[22] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEE Electronics Letters*, vol. 38, no. 25, pp. 1646-1648, 2002.

[23] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.

An Maximum Weighted Likelihood Estimation for Unsupervised Model Selection and Feature Weighting on Gaussian Mixture *

Yiu-ming Cheung, Hong Zeng

Abstract

This paper aims at dealing with two challenging issues in the model-based clustering: the model selection (also referred to as the number clusters determination) and the feature weighting automatically and simultaneously in a single optimization paradigm. This is achieved by introducing two sets of weight functions into the conventional likelihood under the maximum weighted likelihood *(MWL) estimation framework, with one set to reward or penalize the significance of each Gaussian in the entire likelihood, and another set to discriminate the relevance of each feature to the clustering process, both sets are derived dynamically in the competitive learning process according to the intrinsic clustering nature of the samples, providing an adaptive complexity and fitness control scheme. The promising performance is shown through the experiments conducted on both synthetic and real-world data sets.*

1 Introduction

The model-based clustering analysis is performed on the finite mixture composed of underlying probability distributions, trying to partition the objects represented by feature vectors into several homogeneous groups so that objects in the same cluster are similar to each other [1]. In this field, two main issues attribute to the problem's complexity. One is the determination of an appropriate number of clusters for the model, because there is a trade-off between the model complexity and the goodness-of-fit: the true model may not be reflected accurately with too few clusters, whereas the estimated model may seem "over-fitting" the data with too many clusters [8]. Another is the discrimination of inhomogeneous relevance for each feature in respect to the partitioning task. Since the features are collected for a general purpose, they cannot be equally useful in distinguishing one cluster from others. As a result, the partitioning performance of the conventional clustering algorithms which hold the assumption that all features are of the same importance may be degraded. The exacerbating effect may become more depressing when analyzing the high dimensional data sets.

Apparently, the aforementioned two issues are strongly dependent of each other. In a common sense, we always want to use as many features as possible to describe the differences among objects, which will result in a more complex model, consequently, influencing the choice of cluster numbers. It is believed that these two basic clustering issues should be jointly taken into account. In the model-based clustering literature, several approaches which simultaneously address these two problems have been proposed. A recent main advance was presented in [3], where the *saliency* for a feature is utilized as feature weight to describe the capability of discriminating the feature's distributions dependent of clusters or independent of clusters, as an extension to their earlier work [8], it integrates the statistical model selection criterion *Minimum Message Length* (MML) into the maximum likelihood estimation for the simultaneous learning of the feature saliency, finally a revised EM algorithm is derived to iteratively update the parameter set. However, it displays an unrobustness property to sparse data sets. Hence several algorithms [4], [5], [6], [7] are developed under the variational Bayesian or the expectation propagation framework, trying to provide a more robust solution. In [4], [5], [6], the conjugate prior distributions over the model parameters are introduced on the same general mixture model in [3], and the Jensen's inequality is utilized to derive a variational approximation to the true marginal likelihood, eventually an EM-like update scheme is obtained. Nevertheless, it is a rather involved operation to derive the cost function. Furthermore, the slow convergence as a result of the strongly coupled updates makes the them less appealing.

In such a circumstance, we propose a novel approach for Gaussian mixture based clustering with automatic and simultaneous model selection and feature weighting. The basic clustering problems stated above are formulated into a single *maximum weighted likelihood* (MWL) optimization first presented in [9]. Two sets of weight functions are proposed under the MWL framework, with one to reflect the penalty to the model complexity and the other to embody the discrimination for individual features with respect to the clustering process. Both of the two sets of weight functions are derived

* Y.M. Cheung and H. Zeng are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, China. Email:{ymc, hzeng}@comp.hkbu.edu.hk.

dynamically and adaptively in the competitive learning process according to the intrinsic clustering nature. Before moving on to the proposed method, we will review its background of the MWL framework in the next section.

2 The MWL framework

Suppose that the observation data set $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2} \ldots \mathbf{x_N}\}$ is derived from the mixture model, where each observation $\mathbf{x_t}(1 \leq t \leq N)$ is a vector of d-dimensional features: $[x_{1t}, \ldots, x_{dt}]^T$. Then the model is defined by:

$$p(\mathbf{x}_t|\Theta^*) = \sum_{j=1}^{k^*} \alpha_j^* p(\mathbf{x}_t|\theta_j^*) \tag{1}$$

with

$$\sum_{j=1}^{k^*} \alpha_j^* = 1 \quad \text{and} \quad \forall 1 \leq j \leq k^*, \quad \alpha_j^* > 0,$$

where θ_j^* denotes the genuine parameters of the jth probability density function in the mixture model, and k^* is the true cluster number, α_j^* represents the the mixing coefficient, or the proportion of the jth component in the mixture. The main purpose of clustering analysis is to estimate $\Theta^* = \{\alpha_j^*, \theta_j^*\}_{j=1}^{k^*}$ from N observations, a general approach is to search a set of parameters which could reach a maxima of the fitness in terms of *maximum likelihood* (ML) defined below:

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \{\log p(\mathbf{x}_t|\Theta)\}. \tag{2}$$

The usual search strategy to the above ML estimation is the *Expectation Maximization* (EM) algorithm [2]. However, there isn't any penalty function in the likelihood, which means the model order k^* cannot be automatically determined and has to be pre-specified. Thus EM algorithm is not feasible for the real-world data set as the exact number of clusters are often unknown in advance. Recently, an inspiring approach called *Rival Penalized Expectation Maximization* (RPEM for short) has been proposed under the *maximum weighted likelihood* (MWL) framework in [9], which can automatically select the model order. In [9], a set of weight functions $g(j|\mathbf{x}_t, \Theta)$ is introduced into the maximum likelihood as regularization terms, which insure the weighted likelihood does not increase monotonically with the candidate model order k. The weighted likelihood is written below:

$$\begin{aligned} Q(\Theta, \mathbf{X}_N) &= \frac{1}{N} \sum_{t=1}^{N} \log p(\mathbf{x}_t|\Theta) \\ &= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log p(\mathbf{x}_t|\Theta) \\ &= \frac{1}{N} \sum_{t=1}^{N} \mathcal{M}(\Theta, \mathbf{x}_t) \end{aligned} \tag{3}$$

$$\begin{aligned} \mathcal{M}(\Theta, \mathbf{x}_t) &= \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log[\alpha_j p(\mathbf{x}_t|\theta_j)] \\ &\quad - \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log h(j|\mathbf{x}_t, \Theta) \end{aligned} \tag{4}$$

where

$$h(j|\mathbf{x}_t, \Theta) = E[z_{tj}] = \frac{\alpha_j p(\mathbf{x}_t|\theta_j)}{p(\mathbf{x}_t|\Theta)} \tag{5}$$

is the posterior probability that $\mathbf{x}_t$ belongs to the jth component in the mixture.

The weight function $g(j|\mathbf{x}_t, \Theta)$ satisfies the constraint below:

$$\sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) = 1, 1 \leq t \leq N. \tag{6}$$

They are constructed from the following equation:

$$g(j|\mathbf{x}_t, \Theta) = (1 + \varepsilon_t) I(j|\mathbf{x}_t, \Theta) - \varepsilon_t h(j|\mathbf{x}_t, \Theta) \tag{7}$$

with

$$I(j|\mathbf{x}, \Theta) = \begin{cases} 1 & \text{if } j = c \equiv \arg\max_{1 \leq i \leq k} h(i|\mathbf{x}, \Theta); \\ 0 & j = r \neq c. \end{cases} \tag{8}$$

The ranges for k and α_j in (3) are slightly different from those in (1) by: $k > k^*$ and $0 \leq \alpha_j < 1$, for $1 \leq j \leq k$. The iterative update details of the following Maximum Weighted Likelihood estimation can be found in [9]:

$$\hat{\Theta}_{MWL} = \arg\max_{\Theta} \{Q(\Theta, \mathbf{X}_N)\} \tag{9}$$

The construction of weight functions reflects the competitive learning scheme: when a sample $\mathbf{x}_t$ comes from a component which indeed exists in the mixture, the value of $h(j|\mathbf{x}_t, \Theta)$ is likely to be the greatest, thus this component becomes the winner, a positive weight $g(j|\mathbf{x}_t, \Theta)$ placed on it will keep it in the temporary model; on the contrary, all other components are seemly to lose in the competition, then negative weights are assigned to the "pseudo-components", penalizing them proportional to their probabilities to generate the observation. Eventually, only genuine clusters would survive, whereas the "pseudo-clusters" will varnish. A pleasable performance is always obtained with ideal data sets [9], where there is no redundance in the input space, all features are equally useful in clustering process. However, it may not be

the case at all for real data set, thus it is doomed to present a degraded performance for real data. Hence we extend this basic clustering approach with the feature relevancy analysis to overcome this limitation in this paper.

3 Simultaneous Clustering and Feature Weighting

The relevance of the raw feature space can be explained as follows: most features are generated by the components with clustering structures, yet some features cannot be described by the distributions of these clusters, so these features seem to be relevant with respect to the task of recovering clusters from observations, and they are likely to confuse the competition. To discriminate the importance of each feature in the recovering process, we utilize the concept of feature saliency defined in [3] as our feature weight measurement: the lth feature is relevant with a probability $w_l (0 \leq w_l \leq 1, \forall 1 \leq l \leq d)$ that its density distribution is dependent of the density parameters of clusters in the mixture. Assume features are independent of each other, the probability density function of a more general Gaussian mixture model can be written below as in [3]:

$$p(\mathbf{x}_t|\Theta) = \sum_{j=1}^{k} \alpha_j \prod_{l=1}^{d} [w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l)] \quad (10)$$

where $p(x_{lt}|\theta_{lj}) = \mathcal{N}(m_{lj}; s_{lj}^2)$ denotes the Gaussian density function of relevant feature x_{lt} with the mean m_{lj}, and standard deviation s_{lj}; $q(x_{lt}|\lambda_l)$ is the common distribution of the irrelevant features which cannot be described by the specific components, in this paper, we shall limit it to be a Gaussian for a general purpose: $q(x_{lt}|\lambda_l) = \mathcal{N}(cm_l, cs_l^2)$. The full parameter set of the general Gaussian mixture model is redefined as $\Theta = \{\{\alpha_j\}_{j=1}^k, \Phi\}$ and $\Phi = \{\{\theta_{lj}\}_{l=1,j=1}^{d,k}, \{w_l\}_{l=1}^d, \{\lambda_l\}_{l=1}^d\}$. Note that

$$p(x_{lt}|\Phi) = w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l) \quad (11)$$

is a coupling form with two possible densities for each feature, and the feature weight w_l acts as a regulator to determine which distribution is more appropriate to describe the features. It inspires us to regard this form as a *low level* Gaussian mixture, which resembles the *high level* Gaussian mixture on which the proportion of the genuine clusters is estimated, hence the feature weight w_l can be considered as the counterpart of mixing coefficient α_j.

Motivated by the weight functions ($\tilde{g}(.|\mathbf{x}_t, \Theta^{old})$) intended to reduce the model complexity, we designed another set of weight functions ($\tilde{f}(.|x_{lt}, \Phi)$) to discriminate the relevancy of a individual feature in a similar competitive learning manner.

$$\tilde{Q}(\Theta, \mathbf{X}_N) = \frac{1}{N} \sum_{t=1}^{N} \tilde{\mathcal{M}}(\Theta, \mathbf{x}_t); \quad (12)$$

For each time step, the log-likelihood is:

$$\tilde{\mathcal{M}}(\Theta, \mathbf{x}_t) = \sum_{j=1}^{k} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \log p(\mathbf{x}_t|\Theta) \quad (13)$$

According to the Bayes rule:

$$p(\mathbf{x}_t|\Theta) = \frac{\alpha_j p(\mathbf{x}_t|\Phi)}{\tilde{h}(j|\mathbf{x}_t, \Theta^{old})}, \quad (14)$$

$$p(x_{lt}|\Phi) = \frac{w_l p(x_{lt}|\theta_{lj})}{h^{'}(1|x_{lt}, \Phi)} = \frac{(1-w_l) q(x_{lt}|\lambda_l)}{h^{'}(0|x_{lt}, \Phi)}, \quad (15)$$

$$h^{'}(1|x_{lt}, \Phi) = \frac{w_l p(x_{lt}|\theta_{lj})}{w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l)}, \quad (16)$$

$$h^{'}(0|x_{lt}, \Phi) = \frac{(1-w_l) q(x_{lt}|\lambda_l)}{w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l)}. \quad (17)$$

The $h^{'}(1|x_{lt}, \Phi)$ represents the posterior probability that the lth feature conforms to the Gaussian distribution with the parameters for clustering structure, it reflects the confidence for the relevance of the lth feature to the clustering structure, while $h^{'}(0|x_{lt}, \Phi)$ is for the reverse case.

$$\begin{aligned} \tilde{h}(j|\mathbf{x}_t, \Theta) &= \frac{\alpha_j p(\mathbf{x}_t|\Phi)}{p(\mathbf{x}_t|\Theta)} \\ &= \frac{\alpha_j \prod_{l=1}^{d} [w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l)]}{\sum_{j=1}^{k} \alpha_j \prod_{l=1}^{d} [w_l p(x_{lt}|\theta_{lj}) + (1-w_l) q(x_{lt}|\lambda_l)]}, \end{aligned} \quad (18)$$

indicating the probability that some features in the data points come from the jth density component in the subspace. By inserting above all to (13), the final log-likelihood can be written below:

$$\begin{aligned} \tilde{\mathcal{M}}(\Theta, \mathbf{x}_t) &= \sum_{j=1}^{k} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \log \alpha_j + \\ &\sum_{j=1}^{k} \sum_{l=1}^{d} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \Big\{ \tilde{f}(1|x_{lt}, \Phi) \log[w_l p(x_{lt}|\theta_{lj})] + \\ &\tilde{f}(0|x_{lt}, \Phi) \log[(1-w_l) q(x_{lt}|\lambda_l)] \Big\} - \\ &\sum_{j=1}^{k} \sum_{l=1}^{d} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \tilde{f}(1|x_{lt}, \Phi) \log h^{'}(1|x_{lt}, \Phi) - \\ &\sum_{j=1}^{k} \sum_{l=1}^{d} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \tilde{f}(0|x_{lt}, \Phi) \log h^{'}(0|x_{lt}, \Phi) - \\ &\sum_{j=1}^{k} \tilde{g}(j|\mathbf{x}_t, \Theta^{old}) \log \tilde{h}(j|\mathbf{x}_t, \Theta^{old}) \end{aligned} \quad (19)$$

In order to utilize the discriminative model selection power by the RPEM algorithm, we still design the $\tilde{g}(j|\mathbf{x}_t, \Theta)$ as a function of $\tilde{h}(j|\mathbf{x}_t, \Theta)$, and the ε_t is set at a constant of 0.001 in this paper.

The new weight functions $\{\tilde{f}(1|x_{lt}, \Phi), \tilde{f}(0|x_{lt}, \Phi)\}$ satisfy the following constraint:

$$\tilde{f}(1|x_{lt}, \Phi) + \tilde{f}(0|x_{lt}, \Phi) = 1. \quad (20)$$

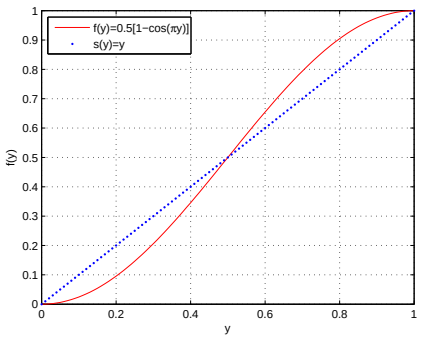


Figure 1. f(y) vs. s(y). f(y) is plotted with "-"; s(y) is plotted with ".";

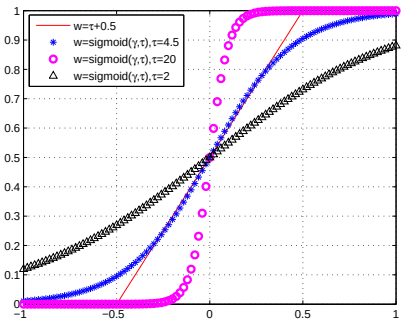


Figure 2. the sigmoid functions with different τ s; $w = \gamma + 0.5$ ($\frac{dw}{d\gamma} \equiv 1$) is plotted with "-".

In order to avoid the collision of the sign of $\tilde{g}(.|\mathbf{x}_t, \Theta^{old})$, $f(.|x_{lt}, \Phi)$ should be kept nonnegative, thus $f(.|x_{lt}, \Phi)$s are not designed in the same way of $\tilde{g}(.|\mathbf{x}_t, \Theta^{old})$, we suggest taking the following forms:

$$\tilde{f}(1|x_{lt}, \Phi) = f(h^{'}(1|x_{lt}, \Phi)), \tag{21}$$

$$\tilde{f}(0|x_{lt}, \Phi) = 1 - f(h^{'}(1|x_{lt}, \Phi)), \tag{22}$$

with the $f(y)$ based on the cosine function:

$$f(y) = 0.5[1 - \cos(\pi y)], y \in [0, 1]. \tag{23}$$

$f(y)$ is plotted in Fig.1. An interesting property of $f(y)$ can be observed from the Fig.1:

$$y > f(y) > 0, \text{for } 0 < y < 0.5;$$
$$y < f(y) < 1, \text{for } 0.5 < y < 1.$$

so when $h^{'}(1|x_{lt}, \Phi) > h^{'}(0|x_{lt}, \Phi)$, which means the feature seems useful in the clustering, then its log-likelihood ($\log[w_l p(x_{lt}|\theta_{lj})]$) is amplified by a higher coefficient $\tilde{f}(1|x_{lt}, \Phi)$, whereas the log-likelihood of the "common" distribution is suppressed by a lower coefficient $\tilde{f}(0|x_{lt}, \Phi)$. A reverse assigning scheme is also held for the case when the feature seems less useful. In this sense, the function $f(y)$ represents an automatic feature discriminating tool, emphasizing on the greater contributions made by the important features to the likelihood and lowering down those of insignificant features. What's more, the dynamical selection is relied on the data's inherent property reflected by the posterior probability $h^{'}(.|x_{lt}, \Phi)$, no statistical or empirical criterion is used.

The Maximum Weighted Likelihood estimation for the whole parameter set is given by:

$$\hat{\Theta}_{MWL} = \arg\max_{\Theta}\{\tilde{Q}(\Theta, \mathbf{X}_N)\} \tag{24}$$

An EM-like iterative updating by gradient ascent technique is used to estimate the parameter set:

step 1 : calculate $h^{'}$, $\tilde{f}$, $\tilde{h}$ and $\tilde{g}$;

$$step\ 2 : \Theta^{new} = \Theta^{old} + \Delta\Theta = \Theta^{old} + \eta \left.\frac{\partial \tilde{\mathcal{M}}(\mathbf{x}_t; \Theta)}{\partial \Theta}\right|_{\Theta^{old}};$$

after a maximum number of epoches is reached, the estimated parameters will be obtained. During the adaptive learning, $\{\alpha_j\}_{j=1}^k$s must be kept conforming to $\sum_{j=1}^{k} \alpha_j = 1$ and $0 \le \alpha_j < 1$, therefore, we also update $\{\beta_j\}_{j=1}^k$s instead of $\{\alpha_j\}_{j=1}^k$ as in [9], and $\{\alpha_j\}_{j=1}^k$ are obtained by:

$$\alpha_j = \frac{e^{\beta_j}}{\sum_{i=1}^{k} e^{\beta_i}},\ for\ 1 \le j \le k. \tag{25}$$

As for another parameter w_l with the constraint: $0 \le w_l \le 1$, we update γ_l instead of w_l, and w_l is obtained by the sigmoid function of γ_l:

$$w_l = \frac{1}{1 + e^{-\tau \cdot \gamma_l}},\ for\ 1 \le l \le d. \tag{26}$$

The constant τ ($\tau > 0$) is used to tune the shape of the sigmoid function. As shown in Fig. 2, if τ is relatively large, a small fluctuation in γ_l will result in a significant change in the value of w_l around 0.5; and when the τ is comparatively small, a relative large change in γ_l only lead to a much smaller drift in the value of w_l around 0.5. Therefore, an approximately equal increasing or decreasing quantity in both γ_l and w_l is more appropriate, in another word, the slope of the sigmoid function around 0 w.r.t. γ_l (namely, around 0.5 w.r.t. w_l) should be roughly equal to 1. By rule of thumb, we find that a linear fitting of the two curves: $w_l = \frac{1}{1+e^{-\tau \cdot \gamma_l}}$ and $w_l = \gamma_l + 0.5$ around 0 w.r.t γ_l happens around $\tau = 4.5$. In the following, we will therefore set τ at 4.5.

For each observation $\mathbf{x}_t$, we calculate the changes in the parameters set as:

$$\begin{aligned}\Delta\beta_j &= \eta_\beta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial\beta_j}\right|_{\Theta^{old}} = \eta_\beta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial\alpha_j}\cdot\frac{\partial\alpha_j}{\partial\beta_j}\right|_{\Theta^{old}} \\ &= \eta_\beta\tilde{g}(j|\mathbf{x}_t,\Theta)(1-\alpha_j^{old}),\end{aligned} \tag{27}$$

$$\begin{aligned}\Delta m_{lj} &= \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial m_{lj}}\right|_{\Theta^{old}} \\ &= \eta\tilde{g}(j|\mathbf{x}_t,\Theta)\tilde{f}(1|x_{lt},\Phi)\frac{x_{lt}-m_{lj}^{old}}{(s_{lj}^{old})^2},\end{aligned} \tag{28}$$

$$\begin{aligned}\Delta s_{lj} &= \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial s_{lj}}\right|_{\Theta^{old}} \\ &= \eta\tilde{g}(j|\mathbf{x}_t,\Theta)\tilde{f}(1|x_{lt},\Phi)\left[\frac{(x_{lt}-m_{lj}^{old})^2}{(s_{lj}^{old})^3}-\frac{1}{s_{lj}^{old}}\right],\end{aligned} \tag{29}$$

$$\begin{aligned}\Delta cm_l &= \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial cm_l}\right|_{\Theta^{old}} \\ &= \eta\sum_{j=1}^{k_{max}}\tilde{g}(j|\mathbf{x}_t,\Theta)\tilde{f}(0|x_{lt},\Phi)\frac{x_{lt}-cm_l^{old}}{(cs_l^{old})^2},\end{aligned} \tag{30}$$

$$\begin{aligned}\Delta cs_l &= \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial cs_l}\right|_{\Theta^{old}} \\ &= \eta\sum_{j=1}^{k_{max}}\tilde{g}(j|\mathbf{x}_t,\Theta)\tilde{f}(0|x_{lt},\Phi)\ \cdot \\ &\quad\left[\frac{(x_{lt}-cm_l^{old})^2}{(cs_l^{old})^3}-\frac{1}{cs_l^{old}}\right],\end{aligned} \tag{31}$$

$$\begin{aligned}\Delta\gamma_l &= \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial\gamma_l}\right|_{\Theta^{old}} = \eta \left.\frac{\partial\tilde{\mathcal{M}}(\mathbf{x}_t;\Theta)}{\partial w_l}\cdot\frac{\partial w_l}{\partial\gamma_l}\right|_{\Theta^{old}} \\ &= \eta\cdot\tau\cdot\sum_{j=1}^{k_{max}}\tilde{g}(j|\mathbf{x}_t,\Theta)\ \cdot \\ &\quad\left[\tilde{f}(1|x_{lt},\Phi)(1-w_l^{old})-\tilde{f}(0|x_{lt},\Phi)w_l^{old}\right].\end{aligned} \tag{32}$$

4 Experimental Results

4.1 Synthetic Data

We generated $N = 1,000$ points of 2 dimensions, they come from a Gaussian mixture of relatively sparse clustering structure:

$$\begin{aligned}&0.3*\mathcal{N}\left[\begin{pmatrix}1.0\\1.0\end{pmatrix};\begin{pmatrix}0.15&0\\0&0.15\end{pmatrix}\right]+\\ &0.4*\mathcal{N}\left[\begin{pmatrix}1.0\\2.5\end{pmatrix};\begin{pmatrix}0.15&0\\0&0.15\end{pmatrix}\right]\\ &+0.3*\mathcal{N}\left[\begin{pmatrix}2.5\\2.5\end{pmatrix};\begin{pmatrix}0.15&0\\0&0.15\end{pmatrix}\right].\end{aligned}$$

where 0.3, 0.4, 0.3 being their proportions in the mixture, respectively. In order to illustrate the ability of proposed algorithm to perform automatic model selection and feature weighting jointly, we appended two independent features to the original set to form a 4-dimensional one. The last two features are sampled from the Gaussian noise covering the entire data set:

$$\mathcal{N}\left[\begin{pmatrix}1.5\\1.5\end{pmatrix};\begin{pmatrix}5.0^2&0.00\\0.00&5.0^2\end{pmatrix}\right]$$

apparently, the last two dimensions are not as significant as the first two dimensions in the clustering process.

Then the robustness of the proposed approach was testified on the sparse data, we initialized k_{max} to 15, and all β_j's and γ_l's to 0, the remaining parameters were randomly initialized. The learning rate are set at $\eta = 10^{-4}$, $\eta_\beta = 10^{-3}$. After we conducted the proposed algorithm for 200 epoches, the mixing coefficients and feature weights are reported below:

$$\begin{aligned}&\hat{\alpha}_6 = 0.4149\ \hat{\alpha}_8 = 0.2963\ \hat{\alpha}_{15} = 0.2888\ \hat{\alpha}_j = 0, j \neq 6, 8, 15\\ &\hat{w}_1 = 0.9973\ \hat{w}_2 = 0.9969\ \hat{w}_3 = 0.0031\ \hat{w}_4 = 0.0033.\end{aligned}$$

The feature weights of the first two dimensions converge close to 1, while those of the last two dimensions are assigned close to 0. To show the learning progress for visualization, we plot a snapshot of the clustering process in the first two features space, along with the learning curve of α_j's, the learning curve of w_l's and the likelihood $Q(\Theta, \mathbf{x}_N)$ in Fig. 3. It can be observed that the algorithm accurately identifies the underlying clustering structures in the first two dimensions, and the proper model order and component parameters are simultaneously estimated.

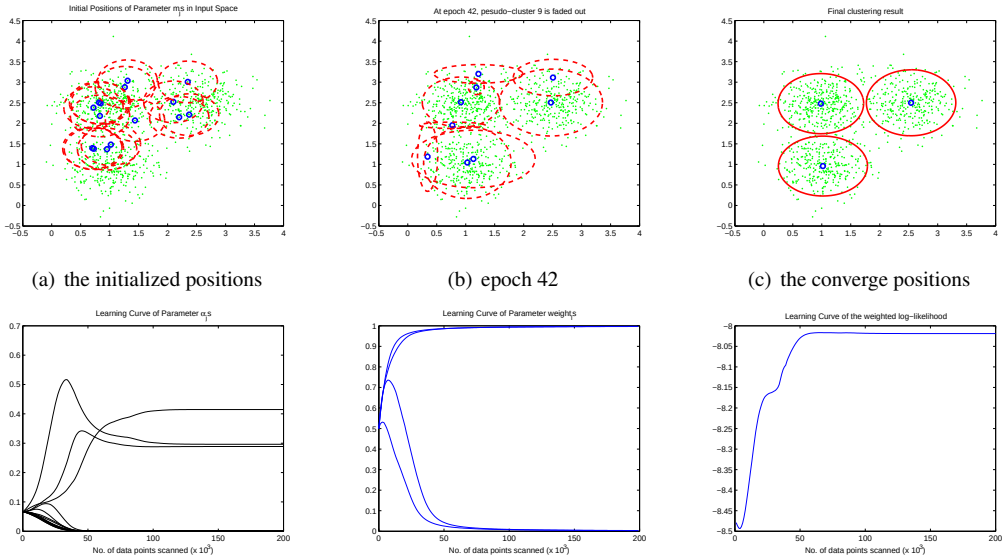
For comparison, the algorithm proposed in [3] was also applied to the same sparse data, it detected only two clusters, and a poor estimation of the weights for the four features was made:

$$\begin{aligned}&\hat{\alpha}_1 = 0.3601\ \hat{\alpha}_2 = 0.6399;\\ &\hat{w}_1 = 0.5179\ \hat{w}_2 = 0.6467\ \hat{w}_3 = 0.1364\ \hat{w}_4 = 0.0470.\end{aligned}$$

4.2 Real-world Data

We further validate the performance of our proposed algorithm on serval benchmark databases [10] for data mining. Five data sets were used for validating our methodology, their characteristics are listed in Table 1.

The partitional accuracy of the algorithm without the prior knowledge of the underground class labels and the relevancy of each features was measured by the *error rate* index. When



(a) the initialized positions (b) epoch 42 (c) the converge positions

(d) the learning curve of α_j (e) the learning curve of w_l (f) the learning curve of $Q(\Theta, X_N)$

Figure 3. The 3-component, 4-dimensional sparse Gaussian mixture data are projected on the first two dimensions, with the "o" marking the center of each cluster, and the circle marking each cluster. In (b), a snapshot of the clustering process shows that the pseudo-cluster ($j = 9$) is faded out ($\alpha_9 < 1/1000$). Through Fig. (d), Fig. (e) and (f), we also observe that the likelihood reaches its maximum when the α_j and w_l get converged.

Table 1. The Real World Data Sets

Data set	d	N	k^*
thyroid	5	215	3
wine	13	178	3
Australian	14	690	2
German	24	1000	2
wdbc	30	569	2

Each data set has N data points with d features from k^* classes.

computing the error rate, the data sets are required to be divided into training and test sets. Hence we randomly split those raw data sets into equal size for the training sets and the testing sets. The process was repeated 30 times, yielding 30 pairs of different training and test sets. For comparison, we conducted the proposed algorithm as well as the RPEM, the approach in [3] (denoted as EMFW). To examine the usefulness of the feature weight function $f(.|x_{lt}, \Phi)$, we also conducted the algorithm (denoted as NFW) with the feature weight function in (23) setting to $f(y) = y, y \in [0, 1]$, which is tantamount to no penalization or promotion for the *low level* Gaussian mixture likelihood for informative feature estimation. We report means and standard deviations of the results obtained on the five sets summarized in Table 2.

Remark 1: In Table 2, it is noted that the proposed method produces much lower error rate than the RPEM algorithm

Table 2. Results of the 30-fold Runs on the Test Sets for Each Algorithm

Data Set	Method	Model Order $Mean \pm Std$	Error Rate $Mean \pm Std$
thyroid	RPEM	2.1 ± 0.2	0.4930 ± 0.0100
	EMFW	4.0 ± 0.9	0.3132 ± 0.2405
	NFW	3.0 ± 0.1	0.1375 ± 0.0559
	proposed method	2.8 ± 0.3	$\mathbf{0.0484 \pm 0.0108}$
wine	RPEM	2.5 ± 0.7	0.0843 ± 0.0261
	EMFW	3.3 ± 1.4	0.0673 ± 0.0286
	NFW	2.8 ± 0.7	0.0955 ± 0.0186
	proposed method	**fixed at 3**	$\mathbf{0.0425 \pm 0.0138}$
Australian	RPEM	3.2 ± 0.4	0.3210 ± 0.0536
	EMFW	4.0 ± 0.0	0.4594 ± 0.1211
	NFW	fixed at 2	0.4855 ± 0.0329
	proposed method	**fixed at 2**	$\mathbf{0.2525 \pm 0.0294}$
German	RPEM	2.1 ± 0.2	0.4620 ± 0.0531
	EMFW	1.7 ± 0.5	0.3510 ± 0.0716
	NFW	fixed at 2	0.3964 ± 0.0543
	proposed method	**fixed at 2**	$\mathbf{0.3440 \pm 0.0215}$
wdbc	RPEM	1.7 ± 0.4	0.2610 ± 0.0781
	EMFW	5.7 ± 0.3	0.1005 ± 0.0349
	NFW	3.0 ± 0.8	0.4871 ± 0.2312
	proposed method	2.5 ± 0.7	$\mathbf{0.0993 \pm 0.0182}$

Table 3. The Average Weighting Results of the 30 Fold-Runs on the Real Data

Data set	Feature													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
thyroid	0.0009	0.8415	0.3345	0.8131	0.9048									
wine	0.9990	0.8799	0.0256	0.2831	0.2354	0.9990	0.9990	0.0010	0.9900	0.9869	0.7613	0.9990	0.0451	
Australian	0.8026	0.0010	0.3953	0.8137	0.2975	0.6412	0.4325	0.5633	0.9990	0.8760	0.0010	0.9082	0.0010	0.9604

The feature weights for German *and* wdbc *are not included as the number of their features is too large to accommodate in this Table.*

which is without the capacity for feature redundance reduction. Table 3 reports the mean feature weights of the sets obtained by the proposed algorithm in the 30 runs, indicating that only several features have good discriminating power.

*Remark 2:*The performance of the NFW algorithm is unstable on the data sets , it indicates that without the feature weight functions to assign unequal emphases on the likelihood for the *low level* Gaussian mixture, the estimation may be ruined by premature decisions on the proper "winner" cluster in the *high level* Gaussian mixture. The results from our algorithm validate the design of proposed feature weight functions.

Remark 3: Relative higher error rates can be observed almost on all the data sets by the MML-criterion based algorithm in [3], whereas the proposed method demonstrates a general significant improvement. Furthermore, the former tends to use more "components" for the mixture, while the proposed algorithm mainly gives a more close order estimation as the sets have specified. From the experimental results above, we infer that the proposed algorithm outperforms its MML-criterion based counterpart.

5 Conclusion

We propose a novel approach to tackle the two basic challenges for Gaussian mixture based clustering problem under the Maximum Weighted Likelihood framework. The model order for the mixture model can be automatically determined, and it is also robust to the more complex and difficult situation when there is redundance in the feature space, where we introduce the feature weights and identify them simultaneously with the model selection process, hence more meaningful clusters can be detected there. The performance advantages over empirical or statistical criterion-based counterpart algorithm is demonstrated through the experiments performed on both the synthetic and the real-world data sets.

References

[1] G. McLachlan and K. Basford, *Mixture Model: Inference and Application to Clustering.* New York: Marcel Dekker, 1998.

[2] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. (B)*, vol.39, no.1, pp.1–38, 1977.

[3] M.H.C. Law, M.A.T Figueiredo and A.K.Jain, "Simultaneous Feature Selection and Clustering Using Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no.9, pp.1154–1166, 2004.

[4] C. Constantinopoulos, M.K. Titsias and A Likas, "Bayesian Feature and Model Selection for Gaussian Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp.1013-1018, 2006.

[5] F. Valente, C.J. Wellekens, "Variational Bayesian Feature Selection for Gaussian Mixture Models", *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp.513–516, 2004.

[6] F. Valente, C.J. Wellekens, "Variational Bayesian Feature Saliency for Audio Type Classification", *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp.513–516, 2005.

[7] Chang, S. Dasgupta, N. Carin, L. "A Bayesian Approach to Unsupervised Feature Selection and Density Estimation Using Expectation Propagation", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp.1043–1050, 2005.

[8] M.A.T. Figueiredo and A.K. Jain,"Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.24, no. 3, pp.381–396, 2002.

[9] Y.M. Cheung,"Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection," *IEEE Trans. Knowledge and Data Engineering*, vol.17, no.6, pp.750–761, 2005.

[10] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/mlearn/MLRepository.html, University of California at Irvine, 1998.

Privacy-Preserving Location-based Queries in Mobile Environments

Jing Du

Abstract

In location based services (LBS), users with location-aware mobile devices can query their surroundings anywhere and at any time. While this ubiquitous computing paradigm brings great convenience for information access, it raises a concern of potential intrusion on users' location privacy, which has hampered the widespread use of LBS. In this paper, we present iPDA, a client-based framework to facilitate privacy-preserving location-based data access in mobile environments. The main idea is to reduce the resolution of user location based on location cloaking and transform a location-based query to a region-based query. We develop an optimal location cloaking technique that is immune to mobility analysis attack. We also propose efficient algorithms to process region-based queries. Extensive experiments are conducted to demonstrate the effectiveness of the proposed algorithms.

1 Introduction

Location based service (LBS) is emerging as a killer application in mobile data services with the rapid development in wireless communication and location positioning technologies [6, 14, 23]. Users with location-aware mobile devices can query their surroundings (e.g., finding all shopping centers within 5 miles or the nearest two gas stations from my current location) anywhere and at any time. However, although this ubiquitous computing paradigm brings great convenience for information access, the disclosure of user locations to service providers raises a concern of intrusion on location privacy, which has hampered the widespread use of LBS [12, 21]. Thus, how to enjoy LBS with preservation of location privacy has been gaining increasing research attention recently [8, 9, 15, 17].

In the literature, there are mainly two categories of approaches to preserve location privacy for LBS. The first is through information access control [6, 7, 24]. User locations are sent to the service providers as usual. It relies on the service providers to restrict access to stored location data through rule-based polices. The second is to employ a trustworthy middleware running between the clients and the service providers. Each time a client makes a location-based request, its location is anonymized by the middleware before being forwarded to the service provider(s) [3, 8, 17]. However, both of these two approaches are vulnerable to misbehavior of the third party. They offer little protection when the service provider / middleware is owned by an untrusted party. There has been private data inadvertently disclosed over the Internet in the past.

In this paper, we present iPDA, an alternative client-based solution to enable privacy-preserving location-based data access for scenarios in which trust in third party is limited. In iPDA, a user can specify for each location-based query the privacy requirement with a minimum spatial area she wants to hide her location. For example, a user can specify it is acceptable to be located within an area of 1 square mile when she is in a shopping center or within an area of 10 square miles when she is in the Disneyland. Upon a location-based query (i.e., a range query or kNN query), iPDA cloaks user's current location with a region and transforms the location-based query to a region-based query. Upon receiving the region-based query, the server evaluates and returns a *result superset* containing the query results for all location points in the cloak region. From the result superset, iPDA refines the actual result.

There are a number of challenging technical issues presented in iPDA, including 1) how to *effectively* cloak user locations to meet user-specified privacy requirements, and 2) how to *efficiently* evaluate result supersets for region-based spatial queries. The idea of using cloaking to reduce location resolution is not new; it has been studied for years (e.g., [8, 9, 17]). However, all existing studies cloak user locations on a snapshot basis and have ignored the spatial locality of client movement. Hence, the existing location cloaking algorithms can be attacked by mobility analysis. A user's location can be easily inferred if she makes queries frequently. We develop a mobility-aware cloaking technique to address this issue. Our contributions made in this paper can be summarized as follows:

- We introduce a client-based framework to empower mobile users to query location-based information with privacy preservation in an autonomous manner.
- We study the *optimal* representation of cloak regions that results in a minimum-size result superset.

- We develop an *optimal* mobility-aware location cloaking technique that resists mobility analysis attack.
- We develop efficient algorithms to evaluate *exact* result supersets for region-based queries (i.e., the query results for all location points in the region and only those results are returned).
- We conduct extensive experiments to demonstrate the effectiveness of the proposed algorithms.

The rest of this paper is organized as follows. Section 2 reviews the work related to our research. In Section 3, we give an overview of the proposed iPDA framework. Section 4 presents how to cloak user locations based on user-specified privacy requirements. The processing of region-based queries is discussed in Section 5. Section 6 evaluates the proposed algorithms via experiments. Finally, this paper is concluded in Section 7.

2 Related Work

Location Privacy Preservation. Preserving data privacy has been extensively studied for general database applications (e.g., [5, 16, 22]). However, relatively fewer works have studied protection of location privacy for location-based services. Most of the existing studies focused on object location tracking. A typical solution is to employ a trustworthy third-party middleware to collect exact locations from moving clients and anonymize location data through de-personalization before release [3, 8, 9]. Beresford and Stajano [3] define some geographical regions as *mix zones*. Once a client enters into a mix zone, its identity is mixed with all other users in the zone. Gruteser and Grunwald [8] provide location anonymity by spatio-temporal cloaking based on the k-anonymity model. A Quad-tree like algorithm is used to perform spatial cloaking. Gedik and Liu [9] extended it to a personalized k-anonymity model. Users can also specify the minimum acceptable spatial resolution and temporal tolerance. A new cloaking algorithm called *CliqueCloak* was developed. However, these previous studies did not consider the spatial locality of client movement in location cloaking. Moreover, the query processing issue has been left out in these studies.

Spatial Query Processing. There is a large body of research work on spatial query processing, in particular kNN search. Most kNN search algorithms were developed based on the R-tree and its variants [10], which index object locations recursively using minimum bounding rectangles (MBRs). To process kNN search, a branch-and-bound approach is employed to traverse the R-tree. At each step, a heuristic is applied to order the index nodes to be visited. At the same time, information is collected to prune the future search space. Various search algorithms differ in terms of the search order and the metric used to prune the search space [11, 18, 20]. While all previous works studied kNN search for a single query point or a line segment, only our recent work investigated kRNN which retrieves the nearest neighbors for all points in a range [13]. However, the focus of our previous work is on high-dimensional data and the query range is limited to a cuboid. In this paper, we consider circle-shaped query ranges and optimize the query performance for 2-dimensional spatial data. Another related work is [4], in which Cheng *et al.* developed algorithms for evaluating probabilistic queries over imprecise object locations. In contrast, we are interested in using imprecise locations to retrieve result supersets for spatial queries.

While the above studies concentrated on location privacy preservation and query processing separately, this paper presents a systematic study on location-based queries with privacy preservation. In concurrent to our work, Mokbel *et al.* [17] presented a framework named Casper for the problem. Our proposal differs from Casper in many aspects. First, Casper relies on a third-party middleware to cloak user locations, whereas in our solution location cloaking is done by the client autonomously. Second, Casper employs the k-anonymity model for privacy requirements, which is not appropriate for a client-based framework as argued in Section 3.1. Instead, we use the minimum cloak area to specify privacy requirements. Third, like [8, 9], the location cloaking algorithm in Casper does not take into account consecutive queries and client movement. In contrast, our proposed algorithm maximizes the cloaking quality and is resistant to mobility analysis attack. Last, the region-based query processor in Casper only returns *inclusive* results (i.e., may return extra unnecessary results) for 1NN queries, whereas we propose general kNN query processing algorithms that efficiently retrieve the *exact* result superset.

3 iPDA Framework

We consider a client-server architecture, where the clients are mobile and are equipped with wireless interfaces to communicate with the server. We assume the clients are location-aware; they can position their own locations (e.g., using GPS or WLAN-based positioning techniques). Users are interested in querying public spatial objects (e.g., hotels, restaurants, gas stations etc.) related to their current locations. These objects are maintained by a spatial database in the server.

As shown in Figure 1, iPDA consists of three main components: location cloaker, query processor, and result refiner. The location cloaker in the client accepts location-based queries $Q(q)$ from users and transform them to region-based queries $Q(D)$. Thus, the resolution of user location is reduced before it leaves the client, thereby protecting location privacy. Upon receiving a region-based query, the query processor evaluates a result superset and

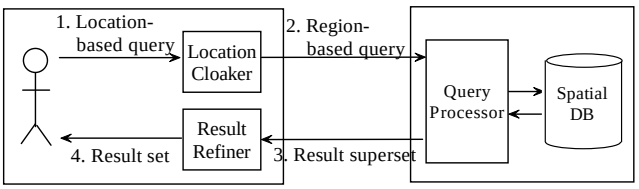


Figure 1. Client-based iPDA Framework

returns it to the client. Assuming the result set of a point-based query is $\mathcal{R}(Q(q))$, the (exact) result superset is thus $\cup_{q \in D} \mathcal{R}(Q(q))$. Finally, the actual result is computed by the result refiner in the client.

Figure 2a shows an example for nearest neighbor (NN) search. Instead of providing its current location q, the client submits a cloak region D_q covering q to the server. The server then returns the set of objects that are potentially an NN of some point in D_q, i.e., b, c, d. Then, the client uses its exact location q to find out the actual NN, i.e., b. Some challenging issues arise in this framework, including 1) how to effectively cloak locations, and 2) how to efficiently evaluate result supersets. Before going into these two issues in detail, we present the privacy measure adopted in the framework.

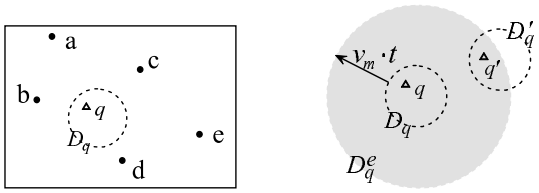
3.1 Privacy Measure

The k-anonymity is a commonly used model to specify privacy requirements in object tracking systems [8, 9]. In this model, an object location is cloaked with a region such that there exist at least $k-1$ other objects in the same region. However, the k-anonymity model is not appropriate for location-based queries due to several reasons. First, it may incur excessive delay if less than k clients issue queries in a short period [9]. Second, it cannot work without knowing neighboring clients' locations in such a client-based framework as iPDA. As such, in this paper, we adopt a simple yet practical privacy measure, i.e., the spatial area of the cloak region. A user can specify a minimum acceptable cloak area with each query. Note that the cloak area requirement can achieve the same level of privacy protection as the k-anonymity model when the user density ρ is available. In this case, we can set the area to be $\frac{k}{\rho}$.

The quality of location cloaking is measured by *entropy*, a well-known metric for representing the amount of uncertainty in information theory [1]. Suppose we can infer that the likelihood of the client being at location (x, y) in cloak region D is $prob(x, y)$, the entropy is defined by:

$$-\iint_D prob(x,y)\ln(prob(x,y))\,\mathrm{d}x\mathrm{d}y. \quad (1)$$

It is obvious that the entropy is maximized when the probability of the client being at any location in the region is equal.



(a) Dynamic cloaking (b) A consecutive query

Figure 2. Dynamic Location Cloaking

4 Location Cloaking

In this section, we present how to cloak a point location by a region to meet user-specified privacy requirements. In the following, we first discuss the representation of cloak regions in Section 4.1 and then the location cloaking techniques in Section 4.2.

4.1 Representation of Cloak Region

This section studies, given a spatial area requirement of cloak region, how to represent the cloak region in terms of shape such that the size of result superset is minimized. A smaller result superset is favorable so as to save wireless bandwidth and reduce the complexity of the final refinement process.

Consider a region-based kNN query, which retrieves the kNNs for all points in the region. The following theorem shows that a region-based kNN query can be decomposed into a range query and a kNN query of the region perimeter:

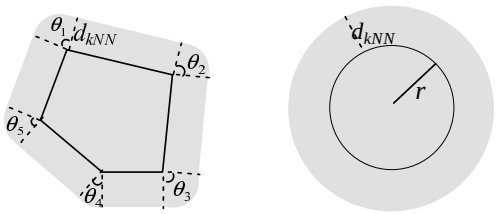
Theorem 1 *An object o is a kNN of region D if one of the following conditions is satisfied: i) $o \in D$ or, ii) o is a kNN of some point on the perimeter of region D.*

Proof Sketch: i) Any object inside the region is the NN of the same point it occupies; ii) If the ith-NN of a point inside the region is an object outside the region, this object must be a jth-NN ($j \leq i$) to some point on the region perimeter. □

For uniform data distribution and unit workspace, the average distance between a query point and its k-th NN can be estimated by [2, 20]:

$$d_{kNN} = \sqrt{k/(\pi N)}. \quad (2)$$

Thus, following Theorem 1, the result search area for a region-based kNN query can be approximated by the area extended from the query region by a distance of d_{kNN} (see shaded areas in Figure 3). We estimate the size of a result superset $\mathcal{R}$ by the number of objects lying in the result search area. Let A and P respectively be the area and perimeter length of the query region, and ρ be the object



(a) Convex Region (b) Circular Region

Figure 3. Search Area of a Region-based kNN Query

density. For a general convex region (Figure 3a), we have

$$|\mathcal{R}| = (A + P \cdot d_{kNN} + \sum_i \frac{1}{2}\theta_i d_{kNN}^2) \cdot \rho$$
$$= (A + P \cdot d_{kNN} + \pi d_{kNN}^2) \cdot \rho. \quad (3)$$

Theorem 2 *A circular region (Figure 3b) results in the smallest $|\mathcal{R}|$.*

Proof: Given different shapes with the same area, from (3), the relative size of $|\mathcal{R}|$ is determined by the perimeter length P. It is well known that a circle has the shortest perimeter under a fixed area. □

This theorem implies that a circular cloak region always produces the minimum-size result superset for kNN queries.[1] We have conducted some experiments over a dataset with 300K randomly distributed objects. The results shown in Figure 4 verify our analysis, in which the result size by a circular cloak region is about 10% less than that by a square region. In the remaining of this paper, we shall use circle to represent a cloak region.

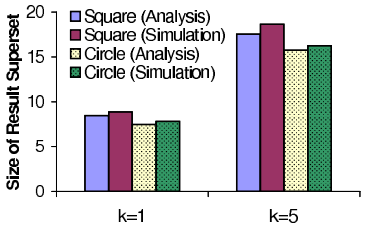


Figure 4. Size of Result Superset ($A = 10^{-5}$)

4.2. Optimal Location Cloaking

There are two basic location cloaking techniques:

- **Static Cloaking [17].** The service area is pre-partitioned into a set of grid cells. Given the current location q, the cell covering q is selected as the cloak region.
- **Dynamic Cloaking.** A random region covering q is dynamically generated to serve as the cloak region, e.g., D_q in Figure 2a.

[1]The same conclusion can be obtained for range queries.

However, both of these two approaches are vulnerable to mobility analysis attack. We use the dynamic cloaking to illustrate the problem. Consider the example shown in Figure 2b. Suppose the client issues another query at location q' with a cloak region D'_q. If the server knows the maximum moving speed of the object, v_m, it can draw an ever-expanding region (D_q^e) from the old region D_q based on the product of v_m and the elapsed time t since the last query. Thus, one can figure out that the chance of the client being in the white area of D'_q (see Figure 2b) is zero, which leads to a low cloaking quality (i.e., entropy as defined in (1)). The performance may further deteriorate with more consecutive queries.

In the following, we develop an *optimal* mobility-aware cloaking technique based on dynamic cloaking. We would like to control the generation of dynamic regions such that the client is equally likely to be at any point in the newly generated region, thereby maximizing the cloaking quality. We consider a general mobility pattern, which both the client and the server are aware of. Denote by O the center of the old cloak region produced for the last query (with a radius of r).[2] Let $u(x)$ be the probability density function of the new object location being distance x away from O at the time of the new query. It follows that

$$\int_0^R u(x)\mathrm{d}x = 1, \quad (4)$$

where R is the farthest distance that the client can travel since the last query, $R = \min\{y \mid u(x) = 0, x \geq y\}$.

Recall that our objective is to generate the new cloak region in such a way that given the new region, the new object location is equally likely to be any point in the new region. To mathematically characterize the goal of optimal cloaking, we define $q(y|z)$ as the probability density function of the new object location being distance y away from O given that the center of the new region is distance z away from O. We first analyze the value of $q(y|z)$ under optimal cloaking.

Assume $z \geq r$. In the left case in Figure 5 (i.e., $0 \leq y \leq z - r$), $q(y|z)$ should be 0. In the middle case in Figure 5 (i.e., $z - r \leq y \leq z + r$), $q(y|z)$ should be proportional to $2\alpha y = 2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y$. In the right case in Figure 5 (i.e., $z + r \leq y$), $q(y|z)$ should be 0. Therefore, we have

$$q(y|z) = \begin{cases} \frac{2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y}{\int_{z-r}^{z+r} 2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y \mathrm{d}y} & \\ & \text{if } z - r \leq y \leq z + r, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Assume $z < r$. In the left case in Figure 6 (i.e., $0 \leq y \leq r - z$), $q(y|z)$ should be proportional to $2\pi y$. In the middle

[2]For clarity of presentation, we assume the old and new cloak regions have a same radius r. Nevertheless, the analysis can be straightforwardly extended to the case where they have different radii.

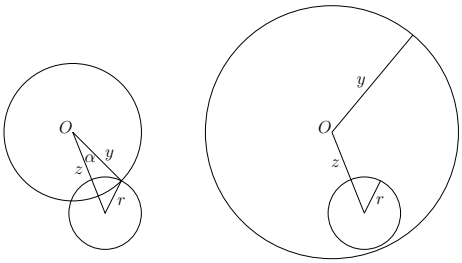


Figure 5. $z \geq r$

case in Figure 6 (i.e., $r - z \leq y \leq z + r$), $q(y|z)$ should be proportional to $2\alpha y = 2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y$. In the right case in Figure 6 (i.e., $z + r \leq y$), $q(y|z)$ should be 0. Therefore, we have

$$q(y|z) = \begin{cases} \frac{2\pi y}{\pi(r-z)^2 + \int_{r-z}^{z+r} 2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y \mathrm{d}y} \\ \qquad \text{if } 0 \leq y \leq r - z, \\ \frac{2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y}{\pi(r-z)^2 + \int_{r-z}^{z+r} 2 \cdot \arccos \frac{y^2+z^2-r^2}{2yz} \cdot y \mathrm{d}y} \\ \qquad \text{if } r - z \leq y \leq z + r, \\ 0 \qquad \text{otherwise.} \end{cases} \tag{6}$$

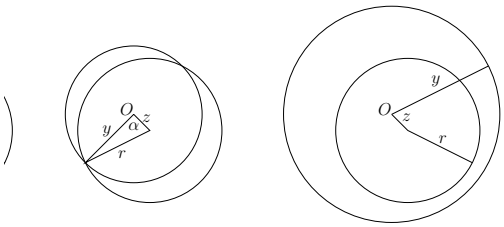


Figure 6. $z < r$

Remember that the cloak region for the new query is generated randomly. To mathematically characterize the random generation, we define $p(z|y)$ as the probability density function of the center of the new cloak region being distance z away from O given that the new object location is distance y away from O. Since the new object location must be in the new cloak region, it follows that

$$\int_{\max\{0,y-r\}}^{\min\{R-r,y+r\}} p(z|y)\mathrm{d}z = 1. \tag{7}$$

Thus, our objective is to find $p(z|y)$ to satisfy the constraints of $q(y|z)$ (i.e., (5) and (6)). Note that the relation between $p(z|y)$ and $q(y|z)$ is given by the Bayes' rule, i.e.,

$$q(y|z) = \frac{p(z|y) \cdot u(y)}{\int_{\max\{0,x-r\}}^{\min\{R-r,x+r\}} p(z|x) \cdot u(x)\mathrm{d}x}.$$

Unfortunately, it is difficult to find a closed-form solution for $p(z|y)$. In the following, we present a numerical method to solve $p(z|y)$ using a discretization technique. We divide the plane into a set of rings of sufficiently small width Δ. The rings are centered at O. As shown in Figure 7, ring 1 is enclosed by a circle centered at O with radius Δ, i.e., ring 1 contains all points that are within distance Δ from O. For each $i > 1$, ring i is enclosed by two circles centered at O with radiuses $(i-1)\Delta$ and $i\Delta$ respectively, i.e., ring i includes all points that are $(i-1)\Delta$ to $i\Delta$ away from O.

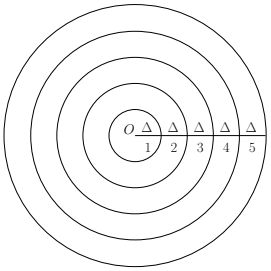


Figure 7. A Set of Rings

Without loss of generality, we assume the radius of a region $r = K\Delta$, and $R = L\Delta$, where K and L are integers. Based on the assumption of mobility pattern, the probability $U(i)$ of the new object location being in ring i is given by $U(i) = \int_{(i-1)\Delta}^{i\Delta} u(x)\mathrm{d}x$, and it follows that

$$U(1) + U(2) + \cdots + U(L) = 1.$$

We define $Q(i|j)$ as the probability of the new object location being in ring i given that the center of the new region is in ring j. Following (5) and (6), $Q(i|j)$ should satisfy:

If $j \geq K$,

$$Q(i|j) = \begin{cases} \frac{\arccos \frac{(i-\frac{1}{2})^2+(j-\frac{1}{2})^2-K^2}{2(i-\frac{1}{2})(j-\frac{1}{2})} \cdot (i-\frac{1}{2})}{\sum_{m=j-K+1}^{j+K-1} \arccos \frac{(m-\frac{1}{2})^2+(j-\frac{1}{2})^2-K^2}{2(m-\frac{1}{2})(j-\frac{1}{2})} \cdot (m-\frac{1}{2})} \\ \qquad \text{if } j - K + 1 \leq i \leq j + K - 1, \\ 0 \qquad \text{otherwise.} \end{cases} \tag{8}$$

If $j < K$,

$$Q(i|j) = \begin{cases} \frac{\pi(i-\frac{1}{2})}{\sum_{m=1}^{K-j} \pi(m-\frac{1}{2}) + \sum_{m=K-j+1}^{j+K-1} \arccos \frac{(m-\frac{1}{2})^2+(j-\frac{1}{2})^2-K^2}{2(m-\frac{1}{2})(j-\frac{1}{2})} \cdot (m-\frac{1}{2})} \\ \qquad \text{if } 1 \leq i \leq K - j, \\ \frac{\arccos \frac{(i-\frac{1}{2})^2+(j-\frac{1}{2})^2-K^2}{2(i-\frac{1}{2})(j-\frac{1}{2})} \cdot (i-\frac{1}{2})}{\sum_{m=1}^{K-j} \pi(m-\frac{1}{2}) + \sum_{m=K-j+1}^{j+K-1} \arccos \frac{(m-\frac{1}{2})^2+(j-\frac{1}{2})^2-K^2}{2(m-\frac{1}{2})(j-\frac{1}{2})} \cdot (m-\frac{1}{2})} \\ \qquad \text{if } K - j + 1 \leq i \leq j + K - 1, \\ 0 \qquad \text{otherwise.} \end{cases} \tag{9}$$

What we want to find out is $P(j|i)$ – the probability of the center of the new region being in ring j given that the new object location is in ring i. Following (7), the definable probabilities are listed as a matrix in Figure 8. The sum of each row in the matrix $\sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} P(j|i) = 1$. After discretization, our problem becomes to find $P(j|i)$ to satisfy the constraints of $Q(i|j)$ (i.e., (8) and (9)).

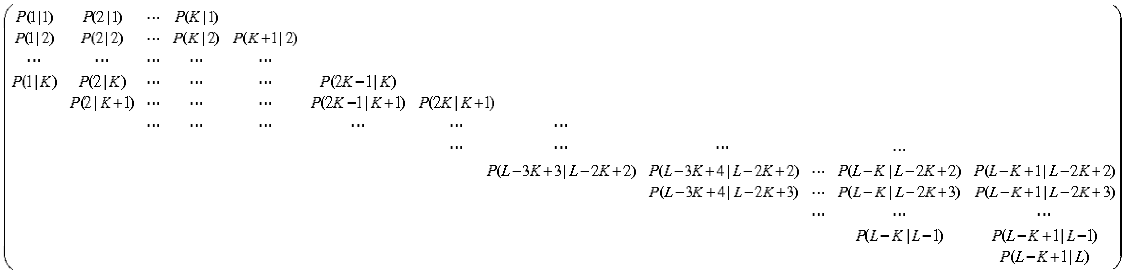


Figure 8. Probability Matrix (showing the upper-left corner and lower-right corner only)

Following the Bayes' rule, for any i and j,

$$P(j|i) = \frac{Q(i|j)}{U(i)} \cdot \sum_m (P(j|m) \cdot U(m)).$$

Let $v_j = \sum_m (P(j|m) \cdot U(m))$. The matrix we want to compute can be rewritten as shown in Figure 9. Recall that for any i, $\sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} P(j|i) = \sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} \frac{Q(i|j)}{U(i)} v_j = 1$. This linear equation set does not always have a feasible solution. To relax the constraints, we propose to allow queries to be *blocked*. If a query from ring $i \leq K$ (i.e., from the old cloak region; called *inner query*) is blocked, the query sending cost is saved and the last result superset can be reused to compute the actual result without sacrificing the accuracy. Thus, a high blocking probability is desired for inner queries. On the other hand, if a query from ring $i > K$ (called *outer query*) is blocked, we can only obtain an approximate query result based on the last result superset and, hence, the query accuracy might be sacrificed.

Thus, by allowing blocked queries, the set of linear equations is relaxed to:

$$\begin{cases} \sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} \frac{Q(i|j)}{U(i)} v_j \leq 1, & i = 1, 2, \cdots, L, \\ v_j \geq 0, & j = 1, 2, \cdots, L-K+1. \end{cases}$$

The v_j's satisfying the above constraints are not unique. To find the "best" solution, we formulate two linear programs with different objective functions.

MaxAccu_Cloak:

$$\text{maximize} \quad \sum_{i=K+1}^{L} \sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} \frac{Q(i|j)}{U(i)} v_j.$$

MinCost_Cloak:

$$\text{maximize} \quad \sum_{i=K+1}^{L} \sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} \frac{Q(i|j)}{U(i)} v_j - \sum_{i=1}^{K} \sum_{j=\max\{1,i-K+1\}}^{\min\{L-K+1,i+K-1\}} \frac{Q(i|j)}{U(i)} v_j.$$

The first objective function MaxAccu_Cloak attempts to minimize the outer query blocking probability and, hence, maximize the query accuracy. In contrast, the second MinCost_Cloak trades bandwidth cost for query accuracy. It aims to maximize the inner query blocking probability as well to increase the result reuse rate and reduce the remote query cost. Their performance will be evaluated by experiments in Section 6.4.

On obtaining $v_1, v_2, \cdots, v_{L-K+1}$, we can compute $P(j|i)$'s. Then, given a new object location, the distance between the center of the new region and O can be randomly generated based on the probabilities $P(j|i)$'s. Given the distance, the center location of the new region can be randomly generated on the corresponding ring segment centered at O.

5 Region-based Query Processing

This section discusses circular-region-based query processing algorithms. The evaluation of a region-based range query is straightforward since it is still a range query (with extended range). Thus, we concentrate on the evaluation of circular-region-based kNN queries. Following Theorem 1, a circular-region-based kNN query can be decomposed into a range query and a kNN query of the circle's perimeter (denoted by Ω). The kNN query of Ω retrieves the set of kNNs for every point on Ω (hereafter called *k-circular-range-NN — k*CRNN query). In the following, we first propose kCRNN processing on an in-memory dataset, and then present three heuristics to prune unnecessary node accesses when applied to a spatial index of the dataset. Finally, we briefly discuss how to integrate the evaluation of a range query and a kCRNN query.

5.1 In-Memory Processing

Different from traditional kNN queries that evaluate an *ordered* list of k objects for a point or a line segment/subsegment, kCRNN only needs to return the set of kNN objects of Ω. Thus, the idea is to scan the objects one by one, and during each scan we maintain the set of arcs for each object on Ω to which this object is the 1st, 2nd, $\cdots$, and k-th NN. Figure 10 shows an example for a 2CRNN

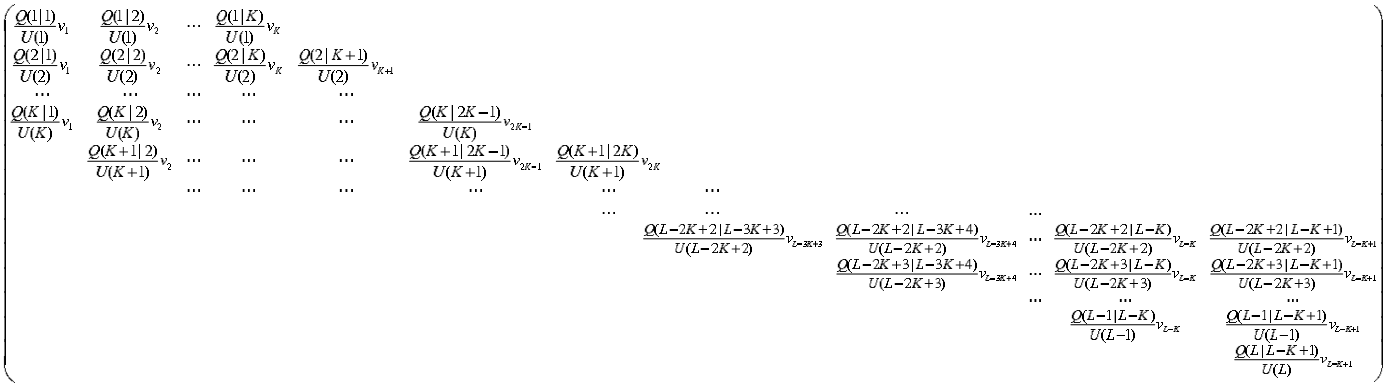
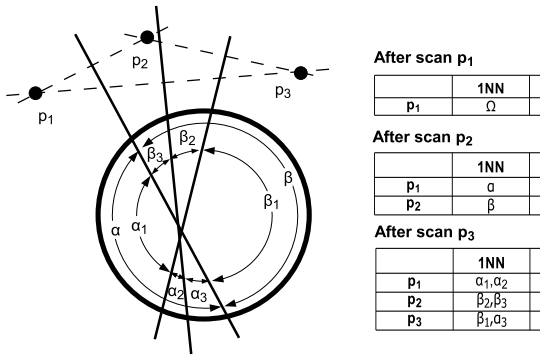


Figure 9. Rewritten Probability Matrix (showing the upper-left corner and lower-right corner only)



After scan p_1

	1NN
p_1	Ω

After scan p_2

	1NN
p_1	α
p_2	β

After scan p_3

	1NN
p_1	α_1,α_2
p_2	β_2,β_3
p_3	β_1,α_3

Figure 10. Finding 2CRNN Results in Memory

query. Suppose there are three objects p_1, p_2, and p_3. Initially p_1 is the 1st NN to the entire perimeter Ω. Then, p_2 is scanned, and the perpendicular bisector (pb) of p_1 and p_2 splits Ω into arcs α and β. Thus, p_2 takes over p_1 to be the 1st NN to β — p_1 is the 1st NN to α and the 2nd NN to β and, for p_2, the order is reversed. Next, p_3 is scanned and we check it against p_1 and p_2. The pbs further split α into α_1, α_2, and α_3, and split β into β_1, β_2, and β_3. Now p_3 takes over p_1 to be the 1st NN for α_3 and the 2nd NN for β_2; p_3 takes over p_2 to be the 2nd NN for α_2 and the 1st NN for β_1. In general, when object p_i is scanned, for any p_j ($j < i$), the pb of p_j and p_i splits the arcs of p_j at two points at most, and p_j becomes the next NN (e.g., the 1st NN becomes the 2nd NN) for those arcs located at the p_i side of the pb. In addition, if the next NN already exceeds k, this arc is removed for p_j. In this example, the arcs for the 3rd NN are removed because only 2CRNN is requested. After each scan, those objects which have at least one l-th-NN arc ($l \leq k$) constitute the kCRNN candidate result set. The algorithm works by scanning the entire set of objects to obtain the final kCRNN results. Each scan checks for a new object against every object in the candidate result set.

Given a kCRNN query with S objects, the total number of arcs for all candidates is bounded by $\mathcal{O}(S^2)$. For each arc, the algorithm downgrades it by k times at most. Thus, the worst-case time complexity is $\mathcal{O}(kS^2)$. For the storage complexity, each arc can appear in the arc sets of at most k objects. Hence, it is also $\mathcal{O}(kS^2)$.

In order to speed up the convergence of the kCRNN candidate set, we apply a heuristic (Heuristic 1) to scan the closest objects to Ω first because they are most likely to bear the final kCRNN results.

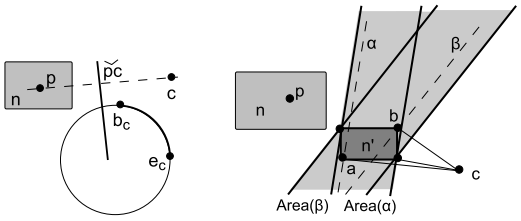
Heuristic 1 *Objects are scanned in the ascending order of their minimum distances to Ω. And those objects whose minimum distances to Ω are $2r$ (r is the radius of Ω) farther than the minimum distance of the k-th NN to Ω are removed from scanning.*

The second sentence of Heuristic 1 sets a stop condition for the scan, because those objects who are $2r$ farther from Ω than the k-th NN must be farther than the maximum distance of the k-th NN to Ω.

5.2 Secondary-Memory Pruning

Intuitively, the overall kCRNN processing is to traverse the spatial index tree (e.g., an R-tree) of the dataset in either depth first order [18] or best first order [11], feed the objects in each leaf index node into the in-memory algorithm, and compute the new kCRNN candidate set until the entire index is visited. However, this searching paradigm is costly for a large dataset. In this section, we develop three heuristics to eliminate most of the nodes from being accessed during the index traversal.

The key observation is that, if the minimum distance between the MBR of a node n and Ω, denoted by $MinDist(n, \Omega)$, is larger than the maximum distance of any kCRNN candidate to its corresponding arcs on Ω, node n can be pruned because the current kCRNN candidate set C will not be updated by visiting n. If n intersects with Ω, $MinDist(n, \Omega) = 0$. Otherwise, $MinDist(n, \Omega)$ is the minimum distance between the MBR of n and the center of Ω (denoted by $MinDist(n, \Omega.o)$), subtracted by Ω's radius r. To compute the maximum distance of kCRNN candidates to their corresponding arcs, we call an arc with a candidate c being the k-th NN as a *k-arc*, and denote it by $\widehat{b_c e_c}$.



(a) The Basic Idea (b) Bound of $\check{pc}$ with Respect to n

Figure 11. Heuristic 3

Let t_c be the far-end intersection of Ω and the line connecting c and $\Omega.o$. The maximum distance between c and its k-arc, denoted by $MaxDist(c, \widehat{b_c e_c})$, occurs only at endpoints b_c and e_c, or t_c if $t_c \in \widehat{b_c e_c}$. As such, the maximum distance of any kCRNN candidate to its corresponding arcs is the maximum of $MaxDist(c, \widehat{b_c e_c})$ for all k-arcs. The first heuristic is formulated as follows.

Heuristic 2 *An index node n with $MinDist(n, \Omega.o) - r > \max_{k-arc}\{MaxDist(c, \widehat{b_c e_c})\}$ should not be visited.*

Heuristic 2, though fast to execute, may not be aggressive enough in pruning. In the next heuristic, we elaborate the relationship between an index node and each k-arc. Figure 11a shows Ω, a k-arc $\widehat{b_c e_c}$, the candidate kCRNN c, and the MBR of node n. p is an arbitrary point in n, and $\check{pc}$ is the perpendicular bisector of line segment $\overline{pc}$. In order for node n to be pruned, for each k-arc $\widehat{b_c e_c}$, there should be no point $p \in n$ such that $\check{pc}$ intersects with this arc.

More specifically, we first obtain a bound of $\check{pc}$ with regard to the MBR of n. The dark grey rectangle n' in Figure 11b shows the bound of the midpoint of line segment $\overline{pc}$, while the slope of $\check{pc}$ is bounded by the slopes of α, β, which are the perpendicular bisectors of $\overline{ac}$ and $\overline{bc}$, respectively (a, b are the two vertices of n'). As such, $\check{pc}$ is bounded by two infinite areas (see the light grey area in Figure 11b). The first (second) of these two infinite areas is formed by two lines that are parallel to α (β) and are touching two vertices of n' (the bold lines in Figure 11b). These two areas are denoted by $Area(\alpha)$ and $Area(\beta)$.

As such, checking if n has possible candidates for Ω is reduced to checking, for each k-arc $\widehat{b_c e_c}$, whether it overlaps the bounded area of $Area(\alpha)$ and $Area(\beta)$, whose sufficient and necessary condition is that, both the two parallel lines that form each area do not intersect with this arc and this arc lie on the same side of the two areas. Formally, Heuristic 3 is shown as:

Heuristic 3 *An index node n satisfying that any k-arc $\widehat{b_c e_c}$ does not intersect with the bounded area of $Area(\alpha)$ and $Area(\beta)$ should not be visited.*

The last heuristic is similar to Heuristic 1. Besides the reason of quick convergence, a more close-to-optimal candidate set has a tighter bound on the maximum distance between a k-arc and its corresponding object and, thus, Heuristics 2 and 3 would be more effective.

Parameter	Setting
Query Interval (I)	4 s
Privacy Requirement (r)	0.001
kNN Search (k)	5

Table 1. Default Parameter Settings

Heuristic 4 *Nodes are visited in the ascending order of their minimum distance to Ω.*

5.3 Evaluating Region-based kNN Queries

Now that the kCRNN query processing is developed, a naive approach to evaluate a region-based kNN query is to separately evaluate the corresponding range query and kCRNN query and combine their results. An obvious optimization is to integrate the evaluation of these two queries. Since the results of the range query are likely to be the kCRNN results, we first use the range query to retrieve the objects with their index nodes' MBRs intersecting with the query range. These objects and their nodes are fed to the range query processor, as well as to the kCRNN query processor as candidate objects/nodes. Thus, repeated accesses of index nodes are avoided. Moreover, the convergence of kCRNN candidate set can be quicker.

6 Performance Evaluation

6.1 Experiment Setup

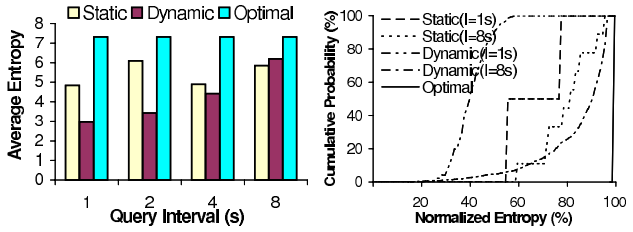
We have developed a testbed to evaluate the performance of the iPDA framework. The client-side program was implemented on a HP iPAQ rx3417 with Samsung S3C2440 300MHz processor and 77MB RAM. The database server was implemented on a Redhat 7.3 Linux server with Intel Xeon 2.80GHz processor and 2GB RAM. The client and the server communicate through 802.11b Wireless LAN.

The spatial dataset used in the experiments contains the centroids of 2,249,727 MBRs representing the street segments in California [19]. We normalize the data space to a unit space and index the centroids by the R-tree [10]. For the R-tree, we set the fanout of an index node (page) at 200 and the page occupancy at 70%. The size of a data record is 512 bytes.

We assume the client moves following a well-known random walk model. It moves in steps. In each step, the client selects a random speed from the range [0.0001 /s, 0.0005 /s] and travels along an arbitrary direction for a duration of 2 s. The client makes kNN queries with privacy requirements from time to time. The average query interval I is set at 4 s by default. The client specifies the privacy requirement with a radius r (i.e., the minimum acceptable cloak area is πr^2). For the numerical method of optimal location cloaking, Δ is set at 0.0001, and the Simplex method is employed to solve

the linear program. The default parameter settings are summarized in Table 1. The experimental results are obtained based on 1,000 random queries.

6.2 Cloaking Effectiveness

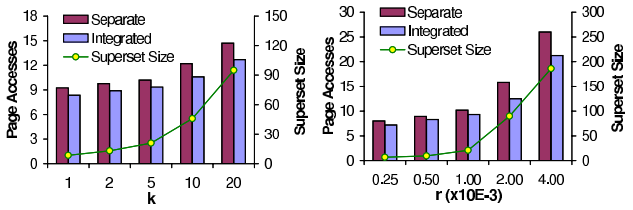


(a) Average Entropy (b) Percentile Entropy

Figure 12. Cloaking Quality (Entropy)

In this section, we compare the proposal optimal cloaking algorithm against static cloaking [17] and dynamic cloaking (described in Section 4.2). For static cloaking, the space is partitioned into a grid such that the size of a grid cell equals the required cloak area. We assume initially the client is equally likely at any location in the cloak region. We measure the cloaking quality in terms of entropy for the next query based on 1500 sample locations. As shown in Figure 12b, when the query interval is small (i.e., 1 s), dynamic cloaking and static cloaking are 40% worse than the optimal cloaking for all cases and 50% of the cases respectively. With increasing query interval, their average performance improves (see Figure 12a) but is still beyond satisfaction. When the query interval is 8 s, over 20% of the cases are less than 80% of the optimal and over 50% of the cases are less than 90% of the optimal. Note that the results shown here are for one consecutive query only. With more consecutive queries coming, the quality of dynamic cloaking and static cloaking will further degrade.

6.3 kNN Query Performance

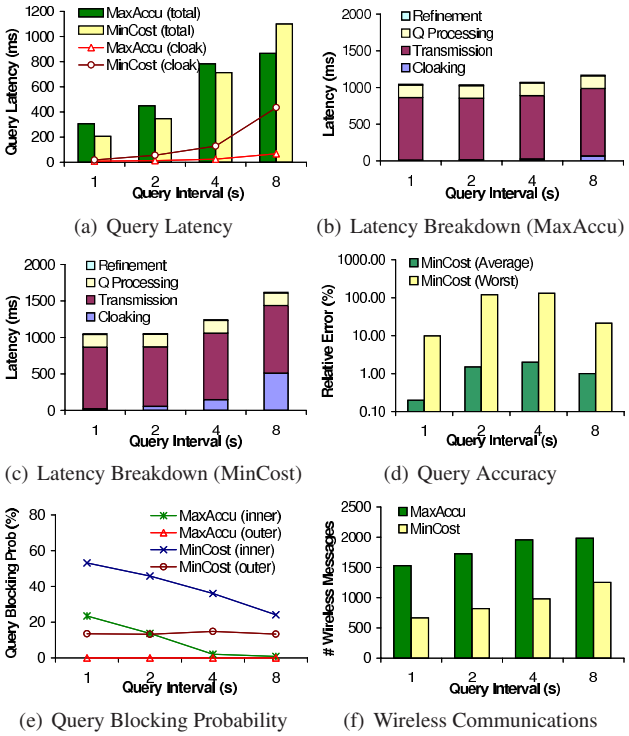


(a) Varying k ($r = 0.001$) (b) Varying Region Size ($k = 5$)

Figure 13. Scalability of kNN Query Processing

In this section, we evaluate the scalability of the proposed region-based kNN query processing algorithm. The best first search [11] of the index tree is applied. Figure 13 shows the disk I/O cost for both separate and integrated evaluation strategies (Section 5.3). It can be observed that the proposed pruning heuristics are efficient such that only about 10 index pages are accessed for each query. When k is exponentially increased from 1 to 20 (Figures 13a), although the result superset is dramatically enlarged, the number of index page accesses is increased linearly. A similar trend is observed when the region radius is increased from 0.25E-3 to 4E-3 (Figures 13b). This indicates that the proposed pruning heuristics are scalable to k and the region size. We can also see that by integrating the evaluation of the range query and kCRNN query, repeated page accesses are saved and quick convergence of the kCRNN candidate set is achieved, which results in a 10%-20% improvement of the integrated evaluation against the separate evaluation.

6.4 iPDA System Performance



(a) Query Latency (b) Latency Breakdown (MaxAccu)

(c) Latency Breakdown (MinCost) (d) Query Accuracy

(e) Query Blocking Probability (f) Wireless Communications

Figure 14. iPDA System Performance

This section evaluates the performance of our proposed iPDA framework for the two objectives, MaxAccu_Cloak (simply *MaxAccu*) and MinCost_Cloak (simply *MinCost*), designed for location cloaking (Section 4.2). Recall that inner queries (i.e., queries inside the old cloak region) are sent to the server for evaluation merely for the purpose of optimal cloaking. For fast response, such queries compute their answers immediately from the last result superset instead of waiting for the server response.

Figure 14 shows the results for these two approaches when the query interval varies from 1 s to 8 s. From Figure 14a, we can see that the query latency is increased with

increasing query interval under both approaches. This is mainly because with a larger query interval, more query points are beyond the old cloak region. Thus, more queries get their answers from remote server access. In general, although MaxAccu requires a shorter time for location cloaking (see the lines in Figure 14a), it has a worse total latency than MinCost. The main reason is because MinCost has a higher outer query blocking rate (see Figure 14e) and, hence, more queries are answered locally. When the query interval is 8 s, MaxAccu performs better than MinCost because in this case the location cloaking time for MinCost is dramatically increased, which outweighs the benefit obtained from a higher outer blocking rate.

We show the latency breakdown for those remote queries for MaxAccu and MinCost in Figures 14b and 14c respectively. In both approaches, the network transmission delay is the most dominating factor while the refinement delay is negligible. As the query interval becomes longer, the complexity of the linear program increases and, hence, the cloaking time is lengthened. The cloaking time of MinCost is generally longer than that of MaxAccu because the liner program with MinCost as the objective requires more time to solve.

As shown in Figure 14e, the outer query blocking rate for MaxAccu is 0. Hence, its query results are 100% accurate. In contrast, MinCost has an outer query blocking rate of about 10%. For those blocked queries, approximate results are obtained based on the last result supersets. Figure 14d shows that the average relative error (measured by the ratio of the distance of an approximate kNN result to the actual kNN distance) is small for blocked queries. In the worst case, the relative error is no more than 130%.

Finally, we plot in Figure 14f the wireless communications in terms of the number of uplink and downlink messages transmitted. Note that without iPDA, the communications are 2,000 messages for 1,000 queries. From Figure 14f, MaxAccu and MinCost require less wireless communications in most cases since they can reuse the last result supersets to evaluate current queries. MaxAccu incurs a higher cost since fewer queries (including both inner and outer queries) are blocked as observed in Figure 14e.

In summary, both of the MaxAccu and MinCost approaches well serve their respective design objectives. MaxAccu achieves a higher query accuracy while MinCost gets a shorter query latency and less wireless transmissions.

7 Conclusions

This paper has presented a systematic study on privacy-preserving location-based query processing in mobile environments. A client-based framework iPDA based on location cloaking has been proposed. We have studied the optimal representation of cloak regions that results in minimum-size result supersets. We have also developed an optimal location cloaking technique that resists mobility analysis attacks and maximizes cloaking quality. Efficient processing algorithms for region-based queries have been developed to return exact result supersets. We have conducted extensive experiments to demonstrate the effectiveness of the iPDA framework.

References

[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. *PODS*, 2001.

[2] S. Berchtold, C. Böhm, D. A. Keim, F. Krebs, and H.-P. Kriegel. On optimizing nearest neighbor queries in high-dimensional data spaces. *ICDT* 2001.

[3] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1), 2003.

[4] R. Cheng, D. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *TKDE*, 2004.

[5] F. Emekçi, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy preserving query processing using third parties. *ICDE*, 2006.

[6] Geopriv Working Group. [Online] http://www.ietf.org/html.charters/geopriv-charter.html, 2005.

[7] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[8] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. *ACM MobiSys*, 2003.

[9] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. *ICDCS*, 2005.

[10] A. Guttman. R-trees: A dynamic index structure for spatial searching. *SIMGOD*, 1984.

[11] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases. *TODS*, 24(2):265–318, 1999.

[12] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. *ACM MobiSys*, 2004.

[13] H. Hu and D. Lee. Range nearest neighbor queries. *TKDE*, 18(1):78–91, 2006.

[14] H. Hu, J. Xu, W. S. Wong, B. Zheng, D. L. Lee, and W.-C. Lee. Proactive caching for spatial queries in mobile environments. *ICDE*, 2005.

[15] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. *Intl. Conf. on Pervasive Services (ICPS)*, 2005.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ICDE*, 2006.

[17] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query procesing for location services without compromising privacy. *VLDB*, 2006.

[18] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. *SIGMOD*, 1995.

[19] The R-tree Portal. [Online] http://www.rtreeportal.org/.

[20] Y. Tao, D. Papadias, and Q. Shen. Continuous nearest neighbor search. *VLDB*, 2002.

[21] B. Schilit, J. Hong, and M. Gruteser. Wireless location privacy protection. *IEEE Computer*, Dec. 2003.

[22] X. Xiao and Y. Tao. Personalized privacy preservation. *SIGMOD*, 2006.

[23] J. Xu, B. Zheng, W.-C. Lee, and D. L. Lee. Energy-efficient index for querying location-dependent data in mobile broadcast environments. *ICDE*, 2003.

[24] M. Youssef, V. Atluri, and N. R. Adam. Preserving mobile customer privacy: An access control system for moving objects and custom profiles. *MDM*, 2005.

Wireless LAN Positioning : Studies on Asymmetrical Signal Strength

Wilson M. Yeung

Abstract

Most of the wireless LAN positioning systems use the Received Signal Strength (RSS) tuple to estimate the current location of user. The RSS tuple can be obtained at access point or mobile client. However, most of the wireless LAN positioning systems use the RSS tuple collected at either access point or mobile client, but not use both of them. In this paper, we propose a new wireless LAN positioning approach which use both RSS tuples, and therefore enhanced the performance. In our experiment, the proposed approach are at least 40% better over the current approaches.

1 Introduction

Wireless LAN positioning has became a hot topic recently. In order to achieve higher precision, many positioning algorithms has been proposed by different research labs. Once we get a suitable positioning algorithm, location-aware services can be provided through the mobile devices, e.g. laptop computer, Personal Digital Assistant (PDA), mobile phone.

Most of the research of wireless LAN positioning use Received Signal Strength (RSS) as location fingerprints, or to calculate the probability of different locations. In fact, the RSS can be collected at Access Point (AP) or at the mobile client(mobile station , MS). The former one means the access points receive the signal from the mobile client while the later one means the mobile client obtains RSS from different access points. Some previous research collect the RSS samples in the former way, while the other research performed in the later way. Besides that, the previous research has used the same wireless network interface, i.e. the wireless LAN card, as both access points and the mobile client, and therefore the transmission power of both access points and mobile client are the same. Thus the RSS collected at both access points and mobile client are more or less the same. In practical environment or application, however, the mobile client is seldom to use the same wireless card as the access points. That means access points and mobile client maybe has different transmission power, and hence the RSS collected at both side maybe different. In other words, the signal strength of access points and mobile client can be asymmetrical in practical environment.

In this paper, we are going to investigate the characteristic of asymmetrical signal strength, and to propose a positioning algorithm which can achieve a higher precision by combining the result of location estimation based on the RSS collected at AP side and client side.

The rest of the paper is organized as follows. In Section 2, we review the related work in wireless LAN positioning. In Section 3, we discuss our observation. We present the proposed algorithm in Section 4. Section 5 contains describe the experiment setup. Experiment result is presented in Section 6. Finally, we conclude the paper and describe the future work in Section 7.

2 Related Works

There are serval positioning algorithms has been proposed in literature, and these algorithms are usually based on either fingerprint approach [1] [2] [7] [9] or probability approach [10] [11] [8]. The RADAR system, by Bahl et al. [1] [2], is a well known fingerprint based wireless LAN positioning system. They used Pentium-based PCs as access points and a laptop computer as mobile client, and both the access points and mobile client used the same wireless adapter. This system works by the mobile client broadcasts packets and the access points record the signal strength they received. By combining the RSS from all the access points at the same moment, a RSS tuple is then formed. Finally, a location fingerprint can be obtained by taking average on the RSS tuples obtained at some known location. Despite RADAR, some previous research also proposed other fingerprint-based positioning algorithm. [7] [9] The basic idea of these algorithms are similar to RADAR, i.e. use the RSS tuples to train the location fingerprints. However, these research used a PDA as a mobile client and measure the signal strength. In other words, they use mobile client to record the RSS from the nearby access points at some known locations, and therefore RSS tuples are then obtained.

The basic principle of probability-based algorithm or system is obtaining RSS distribution from nearby access points at the sampling locations in the offline phase and calculating the probability of every sampling locations based

on the RSS collected in online phase. The location with the highest probability is then reported as current location. The HOURS system, by Youssef et al. [11] [10], is another well known wireless positioning system, which is based on probability approach. They obtain the RSS distribution by using the mobile client to record the signal strength received from the access points at the sampling locations. In fact, the probability-based positioning system by Xiang et al. also used the mobile client to perform the signal measurement.

Besides the academic research, there are some commercial products [5] [4] of wireless LAN positioning have been released. The Ekahua client [5] collects the RSS tuple at the mobile client side. The "Cisco Wireless Location Appliance" [4] seems can obtain the RSS tuple by receiving the signal from a mobile client in the connected access points. Among all these related work, it seems most of the positioning system collects the RSS sample either in the AP sideor the mobile client side. However, to the best of our knowledge, no research has studied the possibility of using RSS tuple collected at both the AP side and the mobile client side to enhance the performance of positioning.

3 Observation

3.1 Practical Environment

In the practical environment, the network services provider needs to install some access points in a specified site, in order to provide the WiFi services to their client. That means the provider can use any kind of access points and set the transmission power of these access points. Unlike the research environment, there is no restriction on choosing and default mobile device. In other words, the services provider most probably doesn't know what kind of devices using their service. In fact, different brand of wireless infrastructure or network interface card (NIC) may use different brand of wireless chipset, and therefore the transmission power of these devices is probably difference. Once the transmission power of the access points and mobile client are difference, the signal strength received on both side should be difference, i.e. asymmetrical signal strength.

In order to confirm the asymmetry of the signal strength, we conducted a preparatory experiment. Figure 1 shows the floor plan of the test site we used, and Table 1 lists the averaged RSS obtained at the specified locations. RSS_1 means the averaged RSS measured by Access Point 1 (i.e. the radio signal is transmitted by the mobile client), while RSS_2 is the averaged RSS measured by the mobile client (i.e. the radio signal is transmitted by the Access Point 1). In this preparatory experiment, the wireless NIC of AP and mobile client we used have different wireless chipsets and transmission power.(The details of the test site and the devices we used will be shown in Section 5.), and obviously the mobile client has lower transmission power. Generally, AP should has larger transmission power than the mobile client, and that actually means we can generate two radio maps, one for radio signal from the AP and another one for the radio signal from the mobile client.

Location	$RSS_1(dBm)$	$RSS_2(dBm)$
1	-53.5833	-60.4167
2	-42.7833	-45.7000
3	-50.2583	-53.9333
4	-53.3333	-57.8583
5	-55.1333	-59.7667
6	-62.1750	-64.9083
7	-66.4083	-68.9167
8	-70.5750	-74.7583

Table 1. Comparison of Averaged Received Signal Strength

3.2 Comparison

As mentioned earlier, we can perform positioning with the RSS tuples collected at access points or mobile client. We name the former one as "AP-approach" and the later one as "MS-approach". The main advantage of AP-approach over the MS-approach is we can obtain the RSS tuples without install any software on the mobile client. That means we can obtain the RSS tuple of any mobile client which is connected to the network. Once a mobile device is connected, we can estimate the current location of that device, and then report the estimation result to the user through some common protocol, e.g. HTTP. In other words, the AP-approach can estimate the user position silently, and the mobile device perhaps uses a embedded tool, say, a browser, to obtain the current position. Unfortunately, the performance of a positioning system based on AP-approach is sensitive to the transmission power of the mobile client. For example, two mobile devices, with different transmission power, can obtain a different result even both devices at the same location. It is because the radio map is device dependant under AP-approach. Once the service providers want to provide positioning service to a new device, they need to obtain a new radio map by that new device, i.e. the calibration cost is high.

Unlike the AP-approach, some RSS capture software needs to be installed on the mobile client, in order to collect the RSS tuple at the mobile client. In fact, the main advantage of MS-approach over AP-approach is easy to implement. In MS-approach, we can obtain the RSS tuple at any moment by calling some wireless NIC or OS API. In

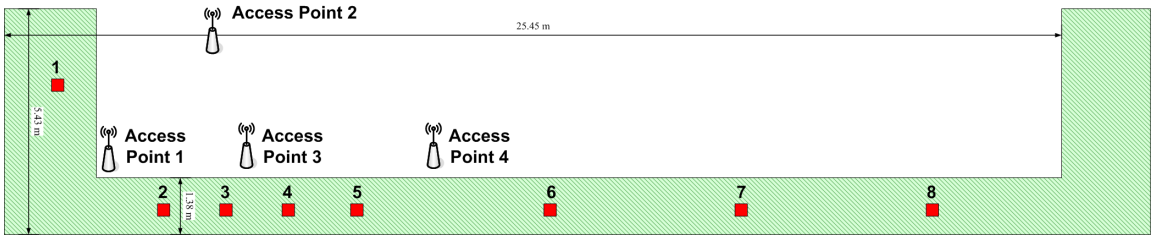


Figure 1. Floor plan 1

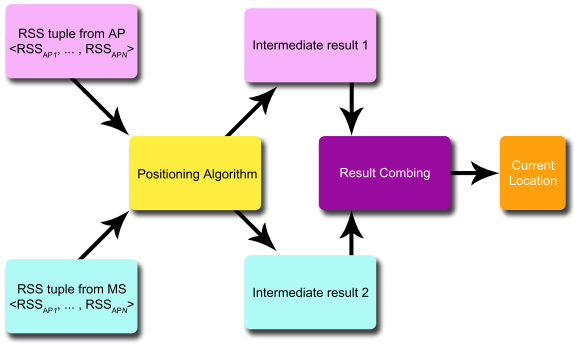


Figure 2. Proposed approach

AP-approach, however, we need to synchronize all installed access points before to obtain a RSS tuple, which increased the difficulty of implementation. Besides that, the performance of MS-approach is more stable, because the radio map is built by the radio signal from access points. Although there are many different mobile devices, the radio signal strength, from the fixed access points , received by these devices is more or less the same, and hence the impact on the performance is lowered.

4 Proposed Approach

Based on the characteristic of asymmetrical signal strength, we introduce a new approach on wireless LAN positioning. At any moment, we can obtain two RSS tuples,one is obtained by AP-approach while another one is obtained by MS-approach. Our proposed approach is combining the estimation result of using these two RSS tuple.

Figure 2 shows the idea of our proposed approach. Each RSS tuple, obtained at either AP or MS, is used to calculate the intermediate result by a positioning algorithm. Afterwards, we get two set of results. Then we combine use these two set of results to calculate the score or probability for each candidate location, and hence to get the result. In our experiment, we implemented fingerprint algorithm [9] as our positioning algorithm. There are many ways to combine the result, for example, averaging , summation, etc.. And in our implementation , we combine the results by this equation :

$$\hat{D_n} = D_{AP,n} \times D_{MS,n} \quad (1)$$

where $\hat{D_n}$ is the score of location n, $D_{AP,n}$ and $D_{MS,n}$ are the smallest Euclidean distance(in signal space) [1] [2] [7] [9] of location n from the intermediate result of AP and MS, respectively. The location with the smallest score is then reported as the result.

5 Experiment Setup

In this section, we describe the experiment setup. Our test site is the corridor area (the green area in Figure 1) of the seventh floor of a nine-storey building. We installed four access points in the test site. (See the figure 1) We use an open source wireless router Linksys WRT54GL [3] as our access point. Rather than using the original OS bundled in the router, we installed OpenWRT [6], which is a open source Linux-based OS, in our AP. To obtain RSS tuple at the access point, we developed a tool which run on the Linux-based AP. Our tool can drive the APs run in monitor mode, which can receive all packet over network they built, and obtain the RSS of the received packet, by calling the wireless chipset driver API, and send the RSS record to our database server. We also used HP iPAQ 4150 Pocket PC, which has a 802.11b wireless LAN card built-in, as our mobile client. To obtain the RSS tuple at mobile client, we developed a Pocket tool which can obtain and save the RSS tuple.

In the test site, we defined 8 sampling locations (the red square in Figure 1). For each location, we obtained sample RSS tuples in each of four directions(say, North , East , South , West). We obtain 30 samples, at both AP and MS, in each direction, and therefore we obtained 960 samples in both AP and MS approach. In order to evaluate the performance of our proposed approach, we trained the fingerprint algorithm with 80% of samples and use the remained 20% samples as out testing samples. The samples collect at AP and MS were trained to the fingerprint algorithm separately,

	AP-approach	MS-approach	Proposed approach
Average error distance	0.815 m	0.552 m	0.306 m
Standard deviation	1.485 m	1.169 m	0.746 m
80^{th} percentile	1.545 m	1.50 m	0.0 m
90^{th} percentile	3.045 m	1.58 m	1.545 m

Table 2. Performance Evaluation

so that we can evaluate the performance of 1) AP-approach , 2) MS-approach , and 3) our proposed approach.

6 Results

The performance among different approaches are summarized in Table 2. The average error distance of AP-approach and MS-approach are 0.815m and 0.552m respectively. The proposed approach yielded 0.306m under average error distance, and which is 66% and 44% better than the result of AP-approach and MS-approach respectively. Besides, the 90^{th} percentile of the proposed approach is 1.545m, while the AP-approach and MS-approach yielded 3.045m and 1.58m. In other words, the error distance of the proposed approach within 1.545m 90% of time. Obviously, our proposed approach enhanced the performance of positioning.

7 Conclusion and Future Work

We proposed a new wireless LAN positioning approach which uses both the RSS collected at access points and mobile client. To test our proposed appraoch, we conducted a experiment which used four access points and one mobile client to obtain the sample data. From the experiment, we can see the proposed approach perform better than the original two approaches.

In order to confirm the stability of our proposed approach, we are going to increase the scale of testing. We will test the proposed approach in a larger indoor environment, and define more sampling locations. Besides, we will try to implement and evaluate different positioning algorithms, and therefore to find out the suitable algorithms for the proposed approach.

References

[1] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the RADAR user location and tracking system. Technical report, Microsoft Corporation, February 2000.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] Cisco Systems, Inc. Linksys WRT54GL V1.1 Wireless-G Broadband Router, http://www.linksys.com/, 2006.

[4] Cisco Systems, Inc. Wi-fi based real-time location tracking: Solutions and technology. 2006.

[5] Ekahau, Inc. , http://www.ekahau.com.

[6] OpenWRT. White Russian RC6 http://openwrt.org/, 2006.

[7] W. H. Wong, J. K. Ng, and W. M. Yeung. Wireless lan positioning with mobile devices in a library environment. In *Proceedings of ICDCS-MDC 2005 Workshop, Columbus, Ohio, USA.*, pages 633–636, June 2005.

[8] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X.Gao. A wireless lan-based indoor positioning technology. *IBM Journal of Research and Development*, Vol. 48, 2004.

[9] W. M. Yeung and J. K. Ng. An enhanced wireless lan positioning algorithm based on the fingerprint approach. In *Proceedings of IEEE TENCON 2006, Hong Kong, China*, 2006.

[10] M. Youssef, A. Agrawala, and U. Shankar. Wlan location determination via clustering and probability distributions, March 2003.

[11] M. A. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh. A probabilistic clustering-based indoor location determination system. Technical Report UMIACS-TR 2002-30 and CS-TR 4350, Department of Computer Science and UMIACS, University of Maryland, March 2002.