PROCEEDINGS

The HKBU 8th Computer Science Postgraduate Research Symposium

August 29, 2008

PG Day 2008

Department of Computer Science
Hong Kong Baptist University

The 8th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

<table>
<tr><th colspan="3">August 29 Friday, 2008</th></tr>
<tr><th>Time</th><th colspan="2">Sessions</th></tr>
<tr><td>09:00-09:20</td><td colspan="2">On-site Registration</td></tr>
<tr><td>09:20-09:30</td><td colspan="2">Welcome: Prof. Jiming LIU, Head of Computer Science Department (RRS 905)</td></tr>
<tr><td rowspan="2">09:30-12:00</td><td>Session A: (Chair: Yu LI) (T909)
Database & Networking</td><td>Session B: (Chair: Victor Cheng) (RRS 905)
Information System & Data mining</td></tr>
<tr><td>• Lazy-update B+ Tree: A New Indexing Method for Flash Devices
Saitung ON
• DigestJoin: A New Join Method for Flash-based Storage Media
Yu LI
• A Survey on Mechanism Design in Distributed Systems
Xiaowei CHEN
• Decentralized Multihop Localization for Wireless Sensor Networks
Chan Tang WONG</td><td>• Collaborative Semantic Indexing of Multimedia Data Objects
Wing Sze CHAN
• Automatic image content annotation and indexing
Chun Fan WONG
• Mining of Ownership for online forum participants
Tianjie ZHAN
• Individualized Reaction Movements with Environments for Virtual Humans
Yuesheng HE
• Empirical Mode Decomposition Applied in Face Recognition
Dan ZHANG</td></tr>
<tr><td>12:00-14:00</td><td colspan="2">Tea Break</td></tr>
<tr><td rowspan="2">14:00-16:00</td><td colspan="2">Session C: (Chair: Jian LI) (RRS 905)
Pattern Recognition</td></tr>
<tr><td colspan="2">• A Novel Motion Based Lip Feature Extraction for Lip-reading
Meng LI
• Joint feature selection for Local Learning Based Clustering
Hong ZENG
• BRF Kernel based discretization for face template security
Yicheng FENG
• Boosting EigenActions: A new algorithm for Human Action Categorization
Chang LIU</td></tr>
<tr><td>17:00</td><td colspan="2">Best Paper & Best Presentation Awards Announcement via Email</td></tr>
</table>

TABLE OF CONTENTS

Session A: Database & Networking

Session B: Information System & Data mining

Session C: Pattern Recognition

Lazy-update B+ Tree: A New Indexing Method for Flash Devices

Sai Tung ON

Abstract

With the rapid increasing capacity of flash memory, flash-aware indexing techniques are highly desired for flash devices. The unique features especially the asymmetric read/write cost of flash memory severely deteriorate the performance of traditional B+ algorithm. In this paper, we propose a new indexing method, called Lazy-update *B+ tree, to make efficient use of ram resource to overcome the limitations of flash memory. The basic idea is to defer the time of committing update requests by buffering them in a segment of memory. They are later committed in groups so that each write operation can be amortized by a bunch of update requests. We formulate the victim selection problem and develop two heuristic-based commit policies. Simulation results show that the proposed method, along with a well-designed commit policy, greatly improves the update performance of the traditional B+ tree, while preserving its efficient query performance.*

1. Introduction

Flash memory has been widely adopted as the data storage media for a wide spectrum of computing devices. Compared with traditional magnetic hard disk, flash memory possesses advantages in various aspects: faster data access, lighter weight, smaller dimensions, better shock resistance, lower power consumption and less noise. These unique features make flash memory the best data storage media for embedded systems and portable handheld devices such as PDA, Smartphones and digital cameras. With recent technology breakthroughs in both capacity and reliability, flash-based devices are capable to handle large, complex, and more data-centric tasks. Therefore, we anticipate more and more DBMS applications will be running on these devices.

As the state-of-the-art database implementation techniques, especially those indexing structures and algorithms, were developed for magnetic hard disk, DBMS applications may not fully achieve their best attainable performance on flash-based devices. Flash memory exhibits a number of characteristics which might have a significant impact on the efficiency of traditional database techniques. First of all, in-place update on flash memory is inefficient due to the erase-before-write constraint. Updating a data item must be preceded by a time-consuming erase operation on a large block of flash memory (called erase block). Second, flash memory has asymmetric read/write cost. The page write cost is more expensive than read, while the block erase requirement makes write cost even more higher. Table 1 [1] shows the read/write/erase speed of a Samsung flash memory chip. We can observe that the ratio of write speed to read speed is 1:2.5, while the ratio of erase speed to read speed is about 1:18.7. Similar observations can be drawn on energy aspect [2, 3]. Third, each block can bare a limited number of erase cycles (typically, 10,000 ~ 100,000 times). A block will be worn out when this number is exceeded. The above features of flash memory point to the same conclusion, i.e., an efficient flash-aware database algorithm should incur as fewer writes as possible, even at the price of introducing more reads or computational cost, as long as the overall performance enhances.

Table 1: Access Time: Magnetic Disk vs. Flash Memory

Media	Access time		
	Read	Write	Erase
Magnetic Disk†	12.7 ms (2 KB)	13.7 ms (2 KB)	N/A
Flash Memory‡	80 µs (2 KB)	200 µs (2 KB)	1.5 ms (128KB)

Magnetic Disk†: Seagate Barracuda 7200.7 ST380011A
Flash Memory‡: Samsung K9WAG08U1A 16 Gbits

B+ tree is the most widely-used index structure because of its high scalability and practicality. It is a balanced tree in which every path from the root to a leaf has the same length. Although B+ tree can enable optimal query performance, maintaining its structure usually requires intensively fine-grained updates over B+ tree nodes. Due to the aforementioned characteristics of flash memory, traditional B+ tree algorithm will definitely encounter severe performance degradation if the B+ tree is implemented upon flash memory, especially when the workload is update-intensive.

In this paper, we propose a new indexing method, called *lazy-update* B+ tree, for flash-based devices. By allocating a segment of main memory (called *lazy-update pool*) to buffer update requests (in the form of key insert/ key delete), *lazy-update* B+ tree greatly reduce the maintenance cost of B+ tree structure while preserving its high query performance. The basic idea is as follows. When an update request arrives, instead of committing it to the B+ tree immediately, it will be temporarily stored

in the *lazy-update pool*. When the pool is filled, guided by a commit policy, a group of update requests will be committed to the B+ tree to reclaim space for buffering future requests. As update requests in the same group are updating the same B+ tree node, each write cost is amortized by several update requests and thus some write operations can be saved. The price of adopting *lazy-update* B+ tree is due to the extra cost of looking up the lazy-update pool when processing queries. As long as a good trade-off between the saving by gathering updates and the aforementioned extra cost is achieved, the overall performance can be improved.

An efficient commit policy is important for *lazy-update* B+ tree method. It has a great impact on how many small updates can be gathered. In the ideal case, the commit policy can always select those groups which do not have any further update requests to gather to commit. As a result, the number of write operations can be minimized. However, this is unlikely to achieve in practice as the entire sequence of future update requests is unknown in advance. A careful designed commit policy is desired to maximize the number of gathered update requests thus reduce write operations.

The rest of the paper is organized as follows. In Section 2, we review the related work on B+ tree algorithms and flash-based data management. Section 3 presents our key ideas and gives an overview of the *lazy-update* B+ tree method. In section 4, we define the victim selection problem and propose two practical solutions. In section 5, we show the performance evaluation results. Section 6 concludes the paper.

2. Related Work

With the rapid development of flash memory technology, data management on flash-based media has received much attention from research community in recent years. To enable a quick deployment of flash-memory technology, early works attempted to hide the unique characteristics of flash memory. They focused on issues in simulating traditional magnetic disk with flash memory chips. Kawaguchi et al. [4] proposed a software module called flash translation layer (FTL) to transparently access flash memory thus conventional disk-based algorithms and access methods can work as usual. To overcome the erase-before-write constraint, out-of-place-update scheme was adopted and various garbage collection mechanisms [4, 5, 6] were proposed to reclaim invalidated space. To lengthen the life span of flash memory, wear-leveling algorithms that attempt to evenly distribute writes/erases across the pages inside flash chips were developed in [7, 8, 9]. Besides these fundamental achievements, recent work shifted to exploit the characteristics of flash memory to enhance the performance of file systems and DBMSs. In view of the poor write speed on flash memory, the log structure is adopted to reduce the number of write requests. On this direction, some flash-aware log-based file systems like YAFFS [10] and JFFS [11] are proposed. Lee and Moon [1] presented a novel design of data logging called in-page logging (IPL) to further improve the logging performance in DBMSs on flash-based media. Kim and Ahn [12] proposed to use the in-device write buffer to improve the random write performance of flash storage. Lee et al. [13] conducted a case study to investigate how the performance of database applications is affected when the magnetic disk is replaced by flash-based media.

B+ tree is a well-known index structure for speeding up data accesses in both file systems and database systems. Extensive research efforts have been put into optimizing B+ tree algorithms. To overcome the asymmetric read/write speed and the erase-before-write limitation on flash-based media, some flash-aware B+/B tree algorithms are developed. Wu et al. [2] introduced BFTL, an optimized B tree layer for flash memory. In BFTL, all changes are written on log pages and therefore expensive update cost for each node is avoided. As a side effect, an in-memory Node Translation Table (NTT) is required to maintain the list of log pages for each node. Observing that the log-based indexing scheme is not suitable for read-intensive workload on some flash devices, Nath and Kansal [3] developed FlashDB, which uses a self-tuning B+ tree that dynamically adapts its storage structure to the workloads and storage devices. Although these indexing methods successfully reduce the update cost, its query performance is degraded because multiple log pages might be accessed when searching a single tree node. Furthermore, as the size of NTT is proportional to the number of tree nodes, the footprint of NTT would consume a large amount of memory.

3. Overview of Lazy-update B+ Tree

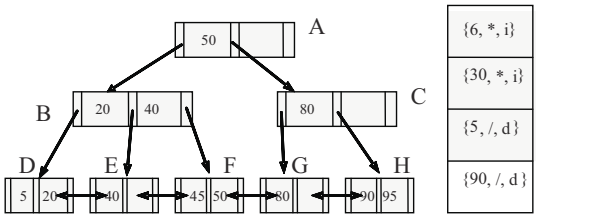
To make efficient use of memory resource, we propose to divide the main memory into two parts: one for caching corresponding pages of accessed B+ tree nodes as usual (known as *page cache*) and another for buffering update requests (called *lazy-update pool*). Each update request is represented in the form of {*key*, *recptr*, *type*} where *key* is the value of the key to be inserted/deleted, *recptr* is the pointer of the inserted record (null for deletion) and *type* indicates the action type (i.e., ‘i’ stands for insert and ‘d’ for delete). A request to modify a key is represented by an insert-type request and a delete-type request.

Whenever an update request arrives, instead of committing it to the B+ tree immediately, it will be temporarily stored in the lazy-update pool. Inside the pool, we use the cancel-out policy to eliminate redundant update requests. In detail, every two in-pool update requests which have the same key but opposite action

types are matched pairs and thus should be eliminated from the pool. Update requests are organized in groups. Each set of requests which are updating the same node forms a *group*. When the pool is insufficient to accommodate the incoming update request, guided by a commit policy, a group of requests are selected as victims and committed to the B+ tree to reclaim space. For queries, in addition to searching over the B+ tree by traditional algorithm, an extra searching of the lazy-update pool is required. Algorithm 1 gives an overview of the *lazy-update* B+ tree method.

```
When a request R arrives,
if R is an update request then
   if lazy-update pool is filled then
      Use a commit policy to select a group of requests as victims
      Commit victims to the B+ tree in bulk
   Buffer R in the lazy-update pool
   Use cancel-out policy to eliminate redundant requests
else                    // R is a query request
   Searching over lazy-update pool to get query result Q1
   Apply traditional algorithm on B+ tree to get query result Q2
   Merge Q1 and Q2 to get the final query result
```

Algorithm 1: Overview of Lazy-update B+ tree



(a) A B+ Tree (order = 3) (b) Lazy-Update Pool (size = 4)

Figure 1: Lazy-update B+ Tree Example

The proposed method offers opportunities to gather small updates which are applied on the same B+ tree node to reduce the number of write operations. Let we use the following example to exemplify how it is attainable. For simplicity, without lost of generality, we assume each B+ tree node is exactly held in a page of flash memory in this paper. Consider the B+ tree in Figure 2(a), where four update requests are issued, i.e., insert keys 6, 30, and then delete keys 5, 90. The letters A~H are the labels of the corresponding pages where the B+ tree nodes reside. Under the traditional method, seven pages (i.e., D, E, B, A, E, D, H) will be updated sequentially. Under the proposed method, the update requests will be stored in the lazy-update pool first (see Figure 2(b)). Then after the pool is filled, these requests will be propagated to the B+ tree in groups. Therefore, {5, /, d} and {6, *, i} will be committed to page D together, and {30, *, i} and {90, /, d} will be committed to page E and H, respectively. As a result, only three pages (i.e., D, E, H) will be updated – a saving of 57% compared to the traditional method.

How to select victims when the lazy-update pool is filled is a critical issue for the *lazy-update* B+ tree method. It decides how many small updates can be gathered. Let's reuse the example in Figure 1 to illustrate it. Suppose that besides those four update requests (i.e., insert keys 6, 30, and then delete keys 5, 90), we have another four update requests, they are, delete keys 40, 95 and then insert keys 99, 100. As the lazy-update pool can hold only four requests, this request sequence should be committed in several batches. Consider a commit policy which always selects all in-pool requests as victims. As a result, these requests are handled in two batches, i.e., {6, *, i},{30, *, i},{5, /, d} and {90, /, d} in a batch, while {40, /, d},{95, /, d},{99, *, i} and {100, *, i} in another batch. As described previously, pages D, E and H will be updated when processing the former batch. For latter batch, {40, /, d} is committed to page E, while {95, /, d}, {99, *, i} and {100, *, i} are committed to page H together, Therefore, totally five pages (i.e., D, E, H, E, H) needs to be updated. If we choose to process them in three batches: {6, *, i} and {5, /, d} first, {30, *, i} and {40, /, d} second and finally {90, /, d}, {95, /, d}, {99, *, i} and {100, *, i}. Then the first batch is committed to page D, the second to page E and the last to page H. That is to say, only three pages {i.e., D, E, H} need to be updated. Hence, the commit policy has a great impact on the performance of the *lazy-update* B+ tree method. We will develop several strategies to guide the victim selection in next section.

4. Commit Policies for Lazy-update B+ Tree

An efficient commit policy is important for *lazy-update* B+ tree as it can gather more update requests applying on the same B+ tree nodes thus reduce the number of write operations. In this section, we first formulate this commit problem and then we propose two strategies to address this problem. We also present the structures and pruning techniques which assist in identifying the victims.

4.1. The Victim Selection Problem

Within the proposed method, when the *lazy-update* pool is filled, a group of update requests are selected to be committed to the B+ tree to reclaim space. For a request sequence, the victim selection problem is to schedule an optimal request committing sequence to finish all requests with minimum cost. In detail, we formulate it as follows.

Definition1: Given a request sequence $S = \sigma_1\sigma_2\cdots\sigma_m$ each of which represents a key insertion/deletion on B+ tree. Consider a *lazy-update* pool which can hold up to N update requests. Let P_i be the set of update requests residing in the pool when the ith request in S comes. Let V_i be the victim group selected to reclaim space when

the *i*th request in S comes. Initially, the pool is empty, thus $P_1 = \phi$ and $V_1 = \phi$. For all $P_i, i = 2, \cdots, m, m+1$,

$$P_i = \begin{cases} P_{i-1} \cup \sigma_{i-1},\ V_i = \phi, & if\ |P_{i-1}| < N \\ P_{i-1} \cup \sigma_i - V_i,\ \ V_i \subseteq P_{i-1}, & if\ |P_{i-1}| = N \end{cases} \quad (1)$$

The Victim Selection Problem is to find a sequence $V = V_1 V_2 \cdots V_m V_{m+1}$ which satisfies (1) and minimizes the following cost function

$$F(S) = \sum_{i=1}^{m+1} \cos t(V_i) + \cos t(P_{m+1}) \quad (2)$$

where $\cos t(V_i)$ and $\cos t(P_{m+1})$ are the costs of committing update requests in V_i and P_{m+1}, respectively.

In this paper, we focus on the online case when the selection of victims is on the basis of knowledge of the past update requests. It is without doubt that an optimal commit policy is unattainable in such case. Therefore, we turn to heuristic-based solutions.

4.2. Biggest Size Policy

In order to increase the hit occurrence, it is more profitable to evict one large group than to evict a bunch of small ones to reclaim the same amount of space. This motivates us to propose *biggest size policy*. This strategy is simple and is easy to implement. Update requests firstly form their groups, then among all groups, the one which has the largest size and is the least recently hit is selected as victim. In order to trace the hit information for each group, we associate each update request with an integer to record its order in the entire request sequence and maintain the order of the newest request for each group.

4.3. Cost-based Policy

While the biggest size policy aims to maximize the hit ratio, the objective of the cost-based policy is to minimize the price resulting from evicting the victim group. Before describing the proposed policy, we first derive the price of evicting the victim group.

Intuitively, a group would gradually expand as long as it stays in the pool to receive new requests. In other words, keeping a group is profitable as it can gather more update requests thus the update cost can be amortized by more requests. A gain function is defined to quantify the aforementioned profit for each group g:

$$gain(g) = \cos t(R) + \cos t(R') - \cos t(R \cup R') \quad , \quad (3)$$

where R is the set of update requests residing in g, R' is the set of new update requests during a future period T, the sum of the first two items is the write cost of committing R and R' separately, and the last item is the write cost of committing them in bulk.

In the following, we will discuss how to compute the gain value. To facilitate our analysis, besides the notations given above, we further define the following notations.

- N : the set of leaf nodes to be updated if R is committed.
- n_k : the k_{th} leaf node in N.
- $range_k$: the range of n_k.
- $f(k; r; t)$: the possibility of having k new update requests whose key values are in the range r during the period t.

For each $range_k$, if $\exists r' \in R', r'.key \in range_k$, then, as both of $\cos t(R)$ and $\cos t(R')$ include a write on the leaf node n_k, we can save one page write if R and R' are committed together. Otherwise, there is no cost saving. Therefore, the cost saving on n_k can be calculated by $(1 - f(0; range_k; T)) \cdot 1$ *page write*. By adding up the cost savings on each leaf nodes, we can get the value of $gain(g)$. That is to say, we have the following formula:

$$gain(g) \approx \sum_k ((1 - f(0; range_k; T)) \cdot 1 page\ write) \quad , (4)$$

where the value of the left item is approximately equal to that of the right item because we ignore some cost savings which rarely happen (e.g., all requests in R are canceled out due to the matched pairs in R', which results in more cost savings).

As can be observed in (4), in order to calculate the gain value for a group, we must first identify a suitable period T and the possibility function. For simplicity, we use the following settings: T = ∞. As the period is set to be infinite, $f(0; r; t) \equiv 0$ holds in any case. Therefore, formula (4) can be simplified as:

$$gain(g) \approx \sum_k 1 page\ write \, . \quad (5)$$

Thus, the gain value solely depends on how many leaf nodes are updated if R is committed.

Apparently, the price of evicting a group g is equal to its gain value. The amount of space reclaimed by evicting g is proportional to its size. In order to minimize the total price of evicting groups, the heuristic solution is to evict the group with the minimum heuristic value given by $H(g) = \frac{gain(g)}{g.size}$ whenever the space is insufficient. The idea behind is to spend as lower price as possible for each unit of reclaimed space. Similar to the biggest size victim policy, when there are several groups having the smallest heuristic value, the one which is the least recently hit is selected as victim. We call this policy as *cost-based policy*.

It is easy to see that cost-based policy reduces to biggest size policy when no rebalancing operations are involved when committing victims from the pool, because under this circumstance each group has equal evicting price.

The space overhead of cost-based policy is due to the need to store the node size and hit information of each group for victim selection. However, we will later see that such overhead is worthy as the performance is greatly improved by adopting this policy.

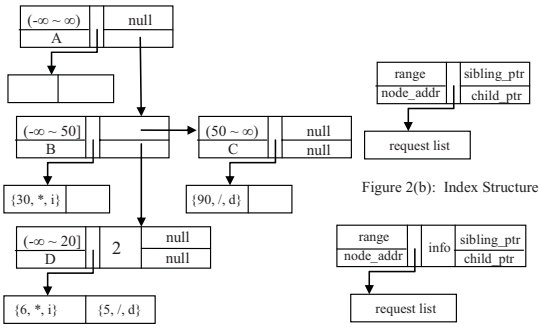


Figure 2(a): An Orthogonal List Example

Figure 2(b): Index Structure

Figure 2(c): Group Structure

4.5. Implementation Issues

4.5.1 Data Structure for Efficient Pool Management

The update requests in the lazy-update pool should be well-organized to support efficient searching and fast identifying victims. We propose to organize them with an orthogonal list to address the above objectives. Figure 2(a) illustrates how the requests in Figure 1 are organized by the proposed orthogonal list. As shown in the figure, the requests are clustered and distributed hierarchically, and hence searching and traversing over the pool can be very efficient.

Each block corresponds to a node in the B+ tree. The one corresponding to the root is called root block. A block b is called child block of b' if its corresponding node is the child of that of b'. Furthermore, block b and b' are called sibling blocks if their corresponding nodes are siblings. There are two types of blocks in the proposed orthogonal list: *index* and *group*.

- *Index*: an index corresponds to a non-leaf node. It has several components: range, request_list, child_ptr, node_addr and sibling_ptr, where range and node_addr are the value range and page address of its corresponding node, respectively, request_list is the array of update requests stored in this block, child_ptr represents a pointer to the head of the chain of its child blocks, and sibling_ptr represents a pointer to its sibling block. The sibling blocks are kept sorted by their ranges. An update request is stored in an index if and only if its key is in the range of this index but not in the range of any other block. Figure 2(b) shows the structure of *index*.
- *Group*: a group corresponds to a leaf node. In addition to the components in *index*, group has an extra component called info to store the information for victim selection. Figure 2(c) shows its structure.

To insert an update request into the orthogonal list, the proper block is first located by searching from the root block to the bottom. Then, the request is directly added into the request_list of that block. We detail it below:

```
Parameter: the update request q, the root block root
block = root, temp = root;
while true do
    while temp ≠ null and q.key not in temp.range do
        temp = temp.sibling_ptr;
    if temp = null then
        break;
    else
        block = temp;
        temp = block.child_ptr;
if there exists a matched pair p of q in block.request_list then
    use the cancel-out policy to eliminate p from the pool;
else
    insert q into block.request_list;
```

Algorithm 2: Insertion Algorithm

Initially, there is only a root block in the orthogonal list. All update requests are stored within this block. When it is time to select victims, these update requests are moved to form groups for victim selection. Meanwhile, their ancestor blocks, together with these groups, are created and added into the orthogonal list. Similarly, when searching the B+ tree for query results, the accessed nodes are used to assist in creating new blocks and thus requests can be clustered in a more fine-grained level. We detail them in Algorithm 3 and 4, respectively.

When some groups are selected as victims to be committed into the B+ tree, correspondingly, they are removed from the orthogonal list. Furthermore, when an index doesn't contain any update requests or child groups, it should be removed as well.

By adopting the orthogonal list to manage the update requests, the extra cost of searching over the lazy-update pool when processing queries is minimized. Moreover, the cost of accessing a node can be amortized by many requests. Therefore, the number of read operations is expected to be reduced too.

4.5.2 Performance Enhancement

We propose a pruning technique to minimize the read operations during the victim selection. Originally, when the pool is insufficient to hold new requests, update requests residing in upper blocks are moved into lower level to form groups for victim selection (see Algorithm

3). Next, the biggest size policy/cost-based policies are adopted to select victims. We propose to merge these two phases together, while leaving along those update requests/groups which are impossible to be selected as victims. We detail it in Algorithm 5 (based on the cost-based policy). The algorithm applying biggest size policy is similar, and the only difference is to change the pruning condition from H(v)*bk.size $\geqslant$ 1 to v.size $\leqslant$ bk.size.

```
Parameter: the root block root
push root into stack;
bk = null;
while stack not empty do
    bk = POP(stack);
    get the chain of its child groups chain by bk.child_ptr;
    if bk is not a group and bk.request_list not empty then
        get the corresponding node node by bk.node_addr;
        divide bk.request_list into r[1], …, r[m] by node's entries
        for i = 1 to m
            create block chd[i] with r[i];
            inset it into chain and keep the chain sorted;
        update the bk.child_ptr if needed;
    for each child in chain
        PUSH(child);
```

Algorithm 3: Group Forming for Victim Selection

```
Input: the query request q, the root block root
Output: query result R
Follow the method in Algorithm 2 to locate the proper block bk using
parameter q.key and root;
temp = null;
for each uq in bk.request_list
    if uq.key = q.key then
        temp = uq; break;
if temp ≠ null    then          //no need to query on B+ tree
    R = temp.recptr;
else
    retrieve the corresponding node startnode by bk.node_ptr;
    search from startnode to leaf on B+ tree to get result R;
    let nodes be the set of nodes accessed in the above searching;
    for each node in nodes
        get the block b which corresponds to node;
        //see algorithm 3 line 7~13
        try to redistribute b.request_list in fine-grained level;
return R
```

Algorithm 4: Handling Query Requests (lookup query)

Besides the pruning technique targeted at reducing read operations, we propose to make efficient use of the page cache to further decrease the number of write operations. As the page cache is used for caching corresponding pages of accessed B+ tree nodes, thus it is profitable to commit those requests whose corresponding leaf nodes are cached and dirty (i.e., the leaf node is updated previously but not yet written back to flash memory). This idea is easy to implement: when a new update request arrives, we verify if it is updating some leaf node which is dirty and cached in the page cache. If so, this request is committed immediately. Otherwise, we buffer it in the lazy-update pool.

```
Input: the root block root
Output: the victim group G
push root into stack;
bk = null,  G = null;              // bk is a block
while stack not empty do
    bk = POP(stack);
    get the chain of its child groups chain by bk.child_ptr;
    if bk is not a group and bk.request_list not empty then
        Let size = the number of requests in bk.request_list;
        if  G = null or  H(G)*size ≥1 then
            get the corresponding node n by bk.node_addr;
            //see algorithm 3 line 7~13
            try to redistribute bk.request_list in fine-grained level;
    else
        if  bk is a group then
            if G = null or H(G)*bk.size ≥ 1  then
                if bk.node_size is unknown  then
                    get the corresponding node l by bk.node_addr;
                    bk.node_size = l.node_size;
                if G = null or H(v) > H(bk)  then
                    G = bk;
                else
                    if H(G) = H(bk) and G.order < bk.order then
                        G = bk;
    for each child in chain
        PUSH(child);
return G
```

Algorithm 5: Victim Selection with Pruning

5. Performance Evaluation

5.1. Simulation Setup and Performance Metrics

We conducted the simulation study on a desktop computer running Windows XP SP2 with an Intel Quad 2.4GHz CPU and 4GB memory. We implemented a FTL module to emulate a 2GB flash memory whose block size and page size are 128 KB and 2 KB respectively by the same amount of main memory. The FTL adopted the page mapping strategy and the greedy garbage collection policy proposed in [4]. The testbed is implemented using Java (JDK 1.5).

We implemented the *lazy-update* B+ tree method and the traditional B+ tree method upon the FTL to make comparison of their performance. In detail, the algorithms under evaluation include: traditional B+ tree (Standard), *Lazy-update* with biggest size policy (Lazy(Biggest)) and *Lazy-update* with cost-based policy (Lazy(Costbased)). In order to verify the effectiveness of the proposed commit policies, we also implemented *Lazy-update* methods with LRU policy (called Lazy(LRU)) and FIFO policy (called Lazy(FIFO)) for comparison.

In our evaluation, we constructed a dataset by using the data from DBLP [14]. We downloaded the XML records from the DBLP website[1] and extracted the data on the author attribute. We converted the name of each author into an integer as the primary key. After that, we

[1] http://dblp.uni-trier.de/xml/

constructed a dataset containing 610,907 distinct authors and built the B+ tree index on their names. For fair comparison, the max amount of main memory space dedicated for each algorithm is equivalent (1% of the index size). The traditional cache replacement policy LRU was applied to the management of the page cache. Both the size of key entry in each node and the size of update request are 12 bytes (8 bytes for key storage while other 4 bytes for storing the associated page address). The order of B+ tree is set to 171 so that each node can exactly fit in a page. The buffer pool of those lazy-update B+ tree algorithms is configured to 50% of the entire assigned memory by default. We summarized the parameter settings in Table 2.

Table 2: Simulation Parameter Settings

Parameter	Setting
Page size/Block size	2 KB/128KB
Key entry /Update request size	12 bytes
Index size	610,907 key entries
B+ tree order	171
Memory size	1% of index size
Buffer Pool size	50% memory space by default

At the beginning of all experiments, the authors which appear in DBLP before year 2007 were selected to build the B+ tree index (540936 entries). Then, each algorithm was tested by running the following workloads:

W-Query: contains 80% queries, 20% updates to be the representative of query-intensive workloads.

W-Update: contains 20% queries, 80% updates to be the representative of update-intensive workloads.

In order to the performance with different delete/insert ratio, we further categorized the above workloads into W-Query(Insert-only),W-Update(Insert-only), W-Query(Mix) and W-Update(Mix). In the first two workloads, updates requests are inserting authors appear after year 2007 into the tree index. In the latter workload, 60% of updates are insert type, while 40% are delete type.

The FTL recorded the performance metrics including the numbers of pages read, pages written, and the computational cost when running the algorithms. We also calculated the overall I/O cost per request by using the write/read speed in Table [1]. The number of block erases is omitted as they seldom happen during our simulation.

5.2. Overall Evaluation

Table 3 ~ 6 show the performance of all algorithms under different workloads. We can observe that *LazyUpdate* methods greatly outperform traditional algorithm: the required write operations of LazyUpdate methods was only half of the traditional method even when the workload is query-intensive. As the accessing cost when updating a tree node is also amortized by a group of requests, the number of pages read of LazyUpdate methods was less than traditional method as well. For the computation cost, as the proposed LazyUpdate methods adopt an orthogonal list to support efficient searching and fast identifying victims, the extra cost of searching over the lazy-update pool when processing queries became a trivial factor.

Among different commit policies, the proposed biggest size policy and cost-based policy outperforms conventional replacement policies (i.e., LRU and FIFO) on I/O aspect, while they had a little bit higher computational cost. The cost-based policy incurred fewer write operations and achieved best overall performance, whereas the biggest size policy required less read operations and computational cost.

Table 3: Performance under W-Query (Insert-Only)

Algorithm	Read	Write	CPU(us)	IO (us)
Standard	601971	98419	27.43	194.04
Lazy(LRU)	538029	62906	25.27	159.10
Lazy(FIFO)	528938	61805	24.71	156.39
Lazy(Biggest)	496541	54231	25.30	144.64
Lazy(Costbased)	497187	48116	25.75	141.29

Table 4: Performance under W-Query (Mix)

Algorithm	Read	Write	CPU(us)	IO (us)
Standard	1706914	258988	27.08	188.53
Lazy(LRU)	1522422	163903	25.16	154.72
Lazy(FIFO)	1495231	162925	24.15	152.35
Lazy(Biggest)	1386600	128896	23.34	136.84
Lazy(Costbased)	1388555	122448	25.99	135.70

Table 5: Performance under W-Update (Insert-Only)

Algorithm	Read	Write	CPU(us)	IO (us)
Standard	165397	97740	101.46	374.67
Lazy(LRU)	112254	61974	87.81	244.32
Lazy(FIFO)	97977	60326	85.91	227.50
Lazy(Biggest)	111571	53784	92.27	224.97
Lazy(Costbased)	113868	47937	93.68	213.71

Table 6: Performance under W-Update (Mix)

Algorithm	Read	Write	CPU(us)	IO (us)
Standard	464322	257376	101.53	354.60
Lazy(LRU)	318688	161821	90.65	231.51
Lazy(FIFO)	275403	159978	85.15	216.18
Lazy(Biggest)	305402	129072	86.00	201.05
Lazy(Costbased)	308465	121435	90.80	195.92

5.3. Effect of Buffer Pool Size

In this part of the experiment, we evaluated the performance of LazyUpdate methods under different buffer pool size. Figure 3 and 4 show the results while the buffer pool ratio is varied from 0.1 to 0.9. As we can see, when the ratio is increased, the number of pages read/written and I/O cost decrease. This can be explained as follows. The bigger is the buffer pool, the more update requests can be buffered. As a result, the chance for the

update requests to gather increases and hence each update cost can be amortized by more requests.

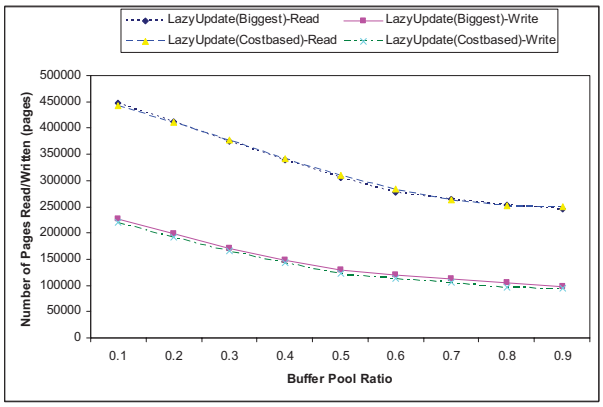


Figure 3: Number of Pages Written/Read Under Different Buffer Pool Size

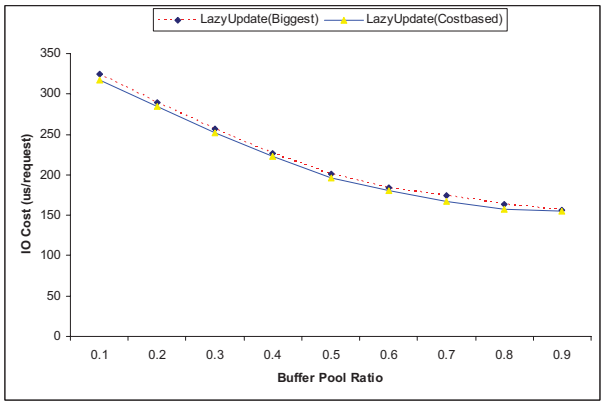


Figure 4: IO Cost Under Different Buffer Pool Size

6. Conclusions

In this paper, we have discussed the challenges of maintaining B+ tree structure upon flash memory. To overcome the asymmetric read/write limitation, we have proposed a new indexing method, called *lazy-update* B+ tree to make efficient use of the ram resource to reduce the number of write operations. We have identified the critical issue of victim selection, and proposed two commit policies to address it. Simulation results show that the lazy-*update* B+ tree method greatly improve the update performance of the traditional B+ tree algorithm, while preserving its efficient query performance.

References

[1] Sang-Won Lee, Bongki Moon, Design of Flash based DBMS: An In-page Logging Approach, Proceedings of the ACM SIGMOD, pp. 55-66, Beijing, China, 2007.

[2] Chin-Hsien Wu, Tei-Wei Kuo, Li-Ping Chang: An efficient B-tree layer implementation for flash-memory storage systems. ACM Transactions on Embedded Computing Systems, vol.6, no. 3, July 2007.

[3] S. Nath and A. Kansal, "FlashDB: a dynamic self-tuning database for NAND flash". Tech. Rep. MSR-TR-2006-168, Microsoft Corporation, 2006.

[4] A. Kawaguchi et al. A Flash-Memory Based File System. In Proc. of the 1995 Winter USENIX Conference, pages 155--164, 1995.

[5] Li-Pin Chang and Tei-Wei Kuo, Real-time Garbage Collection for Flash-Memory Storage System in Embedded Systems, ACM Transactions on Embedded Computing Systems, Vol 3, No. 4, 2004.

[6] Chiang M L, Paul C H, Chang R C, "Manage flash memory in personal communicate devices," In Proceedings of IEEE International Symposium on Consumer Electronics, 1997:177--182.

[7] Han-joon Kim , Sang-goo Lee, A New Flash Memory Management for Flash Storage System, 23rd International Computer Software and Applications Conference, p.284, October 19-26, 1999.

[8] Li-Pin Chang, Tei-Wei Kuo, An efficient management scheme for large-scale flash-memory storage systems, Proceedings of the 2004 ACM symposium on Applied computing, pages 862-868, March 14-17, 2004, Nicosia, Cyprus.

[9] Chang, Y., Hsieh, J., and Kuo, T., Endurance enhancement of flash-memory storage systems: an efficient static wear leveling design. In Proceedings of the 44th Annual Conference on Design Automation, pages 212-217, San Diego, California, June, 2007.

[10] Aleph One Ltd, Embedded Debian, "Yaffs: A NAND-Flash Filesystem" http://www.aleph1.co.ukyaffs/, 2002.

[11] D. Woodhouse. JFFS: The Journalling Flash File System. In Proc. of the Ottawa Linux Symposium, 2001.

[12] Hyojun Kim,Seongjun Ahn, BPLRU: a buffer management scheme for improving random writes in flash storage, Proceedings of the 6th USENIX Conference on File and Storage Technologies, p.1-14, February 26-29, 2008, San Jose, California.

[13] B. Moon, C. Park, S. W. Lee. A Case for Flash Memory SSD in Enterprise Database Applications. Pages 1075-1086, SIGMOD 2008.

[14] BLP Bibliography database.
http://www.informatik.uni-trier.de/~ley/db/

DigestJoin: A New Join Method for Flash-based Storage Media

Yu Li

Abstract

Flash is emerging to be the next generation storage media for DBMS applications. Compared to traditional magnetic disk, a distinguished feature of flash-based storage media is the fast random read. This is against conventional wisdom to avoid random I/Os in query processing. Based on this observation, in this paper, we propose a new join method, called DigestJoin, to exploit fast random read operations for flash-based storage media. The basic idea is to reduce the cost of the join operator at the expense of tuple reloading through a two-phase join. While the tuple reloading cost is critical to the performance of DigestJoin, we show the hardness of this problem by a graph formulation and develop three heuristic-based strategies. Experimental results based on TPC-H datasets show that DigestJoin together with a carefully designed tuple reloading strategy substantially outperforms the traditional join method.

1 Introduction

With recent technology breakthroughs in both capacity and reliability, flash is emerging to be the next generation storage media [6, 10, 13]. While flash memory cards (e.g., Secure Digital and Compact Flash Card) have already been widely used in portable handheld devices such as PDA and Smartphones, Solid State Drive (SSD) shows great promise to replace magnetic hard disk for desktop PCs and file servers. Flash-based storage is superior to its magnetic counterpart in various aspects: faster data access, better shock resistance, lower power consumption, lighter weight, smaller dimensions, and less noise. As such, it is expected that flash will become the main storage media for DBMS applications in the near future. Moreover, since flash-based storage like SSD usually provides the same set of APIs as magnetic disk, the transition from magnetic to flash-based storage will be transparent and smooth for most DBMS applications.

It is well known that seeking is the dominant cost in disk page accesses and that random I/O is more expensive than sequential one for traditional magnetic disk. In comparison to magnetic disk, flash-based storage is not made of any mechanical component. To read/write a data page, the traditional, expensive "seeking" operation is not needed. Yet some overhead still exists before each I/O operation, which is usually caused by the encapsulated logic for such purposes as wear leveling and internal caching [2, 4]. Nevertheless, such overhead is much smaller than the mechanical seeking for magnetic disk. As reported in [7], the overhead before read/write a page on SSD is much cheaper than the seeking cost of magnetic disk, and is comparable to the cost paid for transferring data. In particular, random read of SSD thus become as fast as sequential I/O operations.

However, the state-of-the-art database implementation techniques, especially those query processing algorithms [9], were developed for magnetic disk. They tend to avoid random I/Os and make use of sequential I/Os as much as possible, in the hope of that the expensive seeking cost can be amortized over a series of sequential I/O operations. Such strategies, however, cannot fully exploit the fast random read of flash-based storage media. As relational join is particularly I/O intensive, alternate methods that utilize more random reads than traditional join methods are likely to run significantly faster on flash-based storage media and are therefore highly desirable.

In this paper, we propose a new join method, called *DigestJoin*, for flash-based storage media. Instead of performing the join directly over the original tables, the basic idea of *DigestJoin* is to do a two-phase join. In the first phase, it extracts only minimal information necessary for the join operator, denoted by digest tables, from the original tables. Each entry in the digest table, consisting of a join key and a tuple id (*tid*), corresponds to a data tuple in the original table. A traditional join algorithm is then applied over the digest tables to generate digest join results. Since the size of a digest table could be much smaller than the original table, the join cost can be significantly reduced. Next in the second phase, based on the digest join results, the algorithm reloads the full tuples from the original tables to produce the final join results. Intuitively, this method aims to cut down the cost of the join operator by reducing the sizes of input tables, at the cost of introducing more random reads in the second phase of tuple reloading.

How to efficiently reload the full tuples based on the digest join results is a critical issue for the *DigestJoin* method.

Consider a join on two tables. A digest join result contains a join key and the *tid*s of the two tuples with the same join key value. Based on *tid*s, we can reload the full tuple from the original table, and the entire page containing the tuple should be fetched. Many tuples to be reloaded may be stored on the same page. In the ideal case, each page should be loaded at most once. However, this is difficult to achieve in practice due to memory constraint. As the digest join results are usually not clustered with respect to page address, a careful page loading schedule is desired to minimize the number of page accesses.

Our main contributions made in this paper are summarized as follows:

- We study the possibility of utilizing the fast random read to speed up traditional join algorithms on top of flash-based storage media. To the best of our knowledge, this is the first study to investigate new join methods for flash-based storage media.
- We propose a two-phase *DigestJoin* method that exploits the fast random read of flash-based storage media. The method is generic and can be integrated with any classic join algorithm such as sort-merge and hash joins.
- We formulate the page loading problem by a graph model and show the hardness of this problem. We then develop three heuristic-based solutions.
- We evaluate *DigestJoin* in comparison with the traditional join method in an experimental DBMS implemented on a real SSD device. By comparing their performances with TPC-H benchmark datasets, we show that *DigestJoin* significantly outperforms the traditional method.

The rest of the paper is organized as follows. In Section 2, we give an overview of the *DigestJoin* method and identify its most critical issue of page loading. Section 3 models the page loading problem and develops three loading strategies. Section 4 presents the results of performance evaluation. In Section 5, we review the related work on join processing and flash-based data management. Finally, Section 6 concludes the paper.

2 Overview of DigestJoin

Join is one of the most important operators in RDBMS, and has been extensively studied in the literature. In this paper, we focus on non-index-based joins. Nested-loop, sort-merge, and hash joins are the most well-known non-index-based join algorithms. When joining large tables, one or more tables to be joined may be too large to fit in the usable main memory. In that case, the tables can be loaded into the main memory iteratively (for nested-loop join) or partitioned with the aid of external disk storage (for sort-merge and hash joins). No matter which join algorithm is used, the objective is to incur as few disk IOs as possible for the join process.

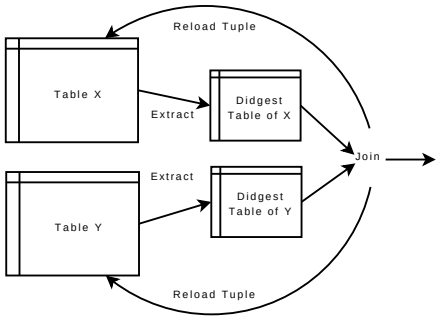


Figure 1. Overview of DigestJoin

Consider the following natural join on two TPC-H tables, where `CUSTOMER` and `ORDERS` are joined through the key `C_CUSTKEY` :

```
SELECT *
FROM CUSTOMER, ORDERS
WHERE CUSTOMER.C_CUSTKEY=ORDERS.C_CUSTKEY;
```

Traditionally, the sort-merge join algorithm first sorts the original tables (using external sorting if the memory is insufficient), followed by a merge through alternative scans of the sorted tables. We can observe that given a certain amount of memory, the join cost directly depends on the sizes of the input tables to be joined. The smaller is the input size, the less is the join cost. Motivated by this observation, the proposed *DigestJoin* method is divided into two phases: *digest join phase* and *tuple reloading phase* (see Figure 1).

Denote table `CUSTOMER` as $X = \{x_1, x_2, \ldots, x_m\}$($x_1$ denotes the column `C_CUSTKEY`) and table `ORDER` as $Y = \{y_1, y_2, \ldots, y_n\}$($y_1$ denotes the column `C_CUSTKEY`). In the digest join phase, the two tables are first abstracted into two digest tables, i.e., $X' = \{x_1, tid_x\}$ and $Y' = \{y_1, tid_y\}$, respectively, where x_1 and y_1 are the join keys, and tid_x and tid_y are the tuple ids that are sufficient to physically locate the corresponding tuples. Then, we apply some traditional join algorithm (e.g., nested-loop, sort-merge, or hash join) over the digest tables to generate the digest join results in the form of $\{x_1, tid_x, tid_y\}$. We expect that the digest tables are much smaller than the original tables. Hence, the cost of joining the digest tables can be significantly cheaper than joining the original tables. However, the digest join results $\{x_1, tid_x, tid_y\}$ only tell us about what tuples will be joined by providing their tids. To output the final join results, in the second phase of tuple

reloading, we have to reload the corresponding tuples from the original tables. Algorithm 1 gives an overview of the *DigestJoin* method.

Step 1: Scan the join tables and construct the digest tables $\{x_1, tid_x\}$ and $\{y_1, tid_y\}$
Step 2: Apply traditional algorithm to join the digest tables
Step 3: Reload full tuples to produce the final join results based on digest join results $\{x_1, tid_x, tid_y\}$

Algorithm 1: Framework of DigestJoin

How to efficiently reload tuples is a critical issue for the *DigestJoin* method. Let's use a simple example to illustrate it. Suppose that we have a list of digest join results: $< x_1, tid_{x_1}, tid_{y_1} >$, $< x_2, tid_{x_2}, tid_{y_2} >$, $< x_3, tid_{x_3}, tid_{y_3} >$, and $< x_4, tid_{x_4}, tid_{y_4} >$, and that tid_{x_1} and tid_{x_3} are stored on page A, tid_{x_2} and tid_{x_4} are on page B, tid_{y_1} and tid_{y_3} are on page C, tid_{y_2} and tid_{y_4} are on page D. If we have sufficient memory space, we may fetch all those four pages A through D and keep them in the memory to facilitate the final join. However, in practice the memory space is limited, and therefore we need to carefully schedule the page loading to minimize the read cost. Suppose that in the worst case the memory space can hold two pages only. If we perform the final join in the order of x_1, x_2, x_3, and x_4, we need to fetch pages A and C for x_1, B and D for x_2, then A and C again for x_3, and finally B and D again for x_4. Here, duplicate page accesses are incurred and in fact each page is loaded twice. Alternatively, if we swap the order of x_2 and x_3 for the final join, each page can be loaded only once. As can be seen, the page loading schedule significantly affects the IO cost and, hence, it is a key to the success of *DigestJoin*. We shall develop several efficient strategies for page loading in the next section.

Before we proceed to the next section, we use the running TPC-H example to exemplify how much *DigestJoin* can improve over the traditional join algorithm. According to the TPC-H definition, the tuple size for a digest table is only 6% of that for the `CUSTOMER` table, and 9% of that for the `ORDERS` table. Assume that there are $10,000$ and $5,000$ pages for `CUSTOMER` and `ORDERS`, respectively. Consider the traditional sort-merge algorithm. Given a memory size of 20 pages, we need 4 and 3 passes to sort these two tables, and the total sorting cost is $4 \times 2 \times 10,000 + 3 \times 2 \times 5,000 = 110,000$ I/Os. The next merge procedure will alternatively scan both of the sorted tables, leading to a total join cost of $110,000 + 10,000 + 5,000 = 125,000$ I/Os. Now if we employ *DigestJoin*, then the tables input to the sort-merge join operator are the digest tables extracted from `CUSTOMER` and `ORDERS`, which are of size 600 and 450 pages, respectively. Thus, the join cost will be reduced to $3 \times 2 \times 600 + 3 \times 2 \times 450 + 600 + 450 = 7,350$ I/Os. As long as the tuple reloading phase introduces fewer than 117,650 page I/Os, *DigestJoin* would outperform the traditional sort-merge join.

We remark that the proposed *DigestJoin* method, while appealing to flash-based storage media, may not be favorable to magnetic disk. This is because the tuples involved in the join results are likely to be scattered over the disk pages. Thus, the tuple reloading will incur quite a number of random read operations, which are extremely costly for disk-based storage media and may even exceed the join cost saved. Fortunately, thanks to the fast random read, flash-based storage media has a better chance to gain from the *DigestJoin* method and we will later demonstrate that with a careful page loading schedule, the tuple reloading cost is far below the join cost saved.

3 Page Loading Strategies for DigestJoin

3.1 Page Loading Problem

In *DigestJoin*, the first phase obtains a list of digest join results, based on which the pages containing the full tuples are fetched to generate final join results. The page loading problem is to schedule an optimal page loading sequence to finish all page joins with minimum number of page accesses.

To further understand the page loading problem, we adopt a graph model to formulate it in a mathematical way. The graph model is used to represent the join relationship between the pages. A join graph is defined as an undirected bipartite graph $G =< V_1 \cup V_2, E >$, where V_1 and V_2 denote the set of pages from the two original tables, respectively, and $E \subseteq V_1 \times V_2$ denotes the set of page-pairs to be joined. For each $(v_a, v_b) \in E$, there must exist some tuple on page v_a which has join result with some tuple on page v_b. The join graph can be used to dynamically model the remaining pages to be loaded and joined. An edge (v_a, v_b) is removed from E if pages v_a and v_b have been loaded into the memory and joined. A vertex v is removed from G once its degree becomes zero (i.e., no pages to be joined with).

An example of join graph is shown in Figure 2, where vertices 1 and 2 represent the pages from one table, while vertices a, b and c represent the pages from the other table. An edge $(1, a)$ means there are tuples on pages 1 and a to be joined in the final results. Similarly, edges $(1, b)$, $(1, c)$, $(2, a)$ and $(2, c)$ represent the join relationship between other pairs of pages.

Based on the join graph, a page loading sequence is equivalent to an order of removing all edges of the graph. As mentioned previously, an edge can be removed if and only if the corresponding pages are loaded into the memory and joined. Therefore, an optimal page loading sequence is an order of loading pages into the memory to remove all

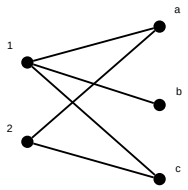


Figure 2. Example of Join Graph

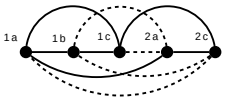


Figure 3. Corresponding Graph of the Join Graph in Figure 2

edges in the join graph with minimum number of page accesses. If there is sufficient memory space to hold all related pages, apparently only one-time access for each page is needed and, thus, any order of page loading sequence would be optimal. However, when the memory space is limited, page swaps will be resulted and the problem becomes very complicated.

To ease the analysis on the hardness of this problem, we transform the join graph into another equivalent graph model. Specifically, for a join graph $G =< V_1 \cup V_2, E >$, we construct a corresponding weighted complete graph $G' =< V', E' >$ by the following method: for each edge $(v_a, v_b) \in E$, we create a new vertex v' in G', that is to say, each edge in the join graph is transformed to a vertex in the new graph. Therefore, v' in the new graph represents a join operation: loading the corresponding pages v_a and v_b into the memory to join. We assign the weight of each edge as the page swapping cost between two join operations implicated by its vertexes. Figure 3 shows the transformed graph of the join graph in Figure 2. In the graph, we name each vertex by its corresponding edges in the join graph, e.g., vertex $1a$ corresponds to the edge $(1, a)$ in Figure 2. As for the edge weight, if we assume the memory can hold two pages only, it is 1 for edge $(1a, 1b)$ (solid line) as one page has to be swapped out, and it is 2 for edge $(1b, 2a)$ (dashed line) as two pages have to be swapped out.

With the new graph model, a page loading sequence is transformed to a tour of all vertices on the graph, and its page access cost is transformed to be the sum of the weights of all edges in the tour. Thus, the page loading problem is to find a tour in the complete graph with minimum cost, which is well-known as the classic Hamiltonian path problem. In the literature, Merrett et al. [8] has studied a special case of the page loading problem by constraining the main memory to hold exactly two pages. They proved that the constrained problem can be reduced from a special case of the Hamiltonian path problem and showed that the problem of determining whether there exists a solution with $n-1$ page swaps is *NP-complete* (assume there are totally n pages in the join graph).

However, in general cases when the memory space can hold more than two pages, the problem becomes even more complex as the weight of each edge will dynamically change. Since the special case of the page loading problem is already *NP-complete*, its general version should be at least *NP-complete*. To the best of our knowledge, no approximation algorithm exists to address the page loading problem. In the rest of this section, we turn to heuristic solutions and propose three strategies for the page loading problem.

3.2 Naive Loading Strategy

One intuitive idea to tackle the page loading problem is to load pages online whenever there is a digest join result, it loads the corresponding pages to retrieve the tuples and produce the join result. This is called naive loading strategy. For each digest join, only two page buffers are required to load the corresponding tuples, and the remaining available memory can be used as page cache. The traditional cache replacement policy LRU can be applied to the management of the page cache. We detail this strategy in Algorithm 2.

Input : Sorted digest table dt_1, sorted digest table dt_2
Output: Join results

Allocate buffer p_1 for dt_1, p_2 for dt_2
Allocate rest buffers as *PageCache* with LRU policy
while *joining* dt_1 *and* dt_2 *with* p_1 *and* p_2 **do**
  if *there is a join result* $< join\_key, tid_1, tid_2 >$ **then**
    $page_1$ = the page id containing tid_1
    $page_2$ = the page id containing tid_2
    if $page_1$ *in PageCache* **then**
      Extract tuple with tid_1 to t_1
    else
      Load $page_1$ into *PageCache*, extract tuple with tid_1 to t_1
    if $page_2$ *in PageCache* **then**
      Extract tuple with tid_2 to t_2
    else
      Load $page_2$ into *PageCache*, extract tuple with tid_2 to t_2
    Join t_1 and t_2 and output

Algorithm 2: Naive Page Loading

3.3 Page-based Loading Strategy

Although the page loading problem has been shown to be at least *NP-Complete*, there exist some special cases when the problem can be easily solved. The first case is when the memory space is sufficient to hold all pages having join results, as discussed in the previous subsection. Another case

is when the digest join results are clustered with respect to page address and the full tuples having join results are clustered with respect to the join key. In this case, directly applying the naive loading strategy would result in an optimal page loading sequence which ensures each page is loaded at most once. This motivates us to propose the *page-based loading strategy*.

Within this strategy, we build two kinds of temporary tables to assist in page loading. The first one is called *reloading instruction table*, which archives digest join results. After this table is created and filled with all digest join results, we sort and cluster its digest join results based on their page addresses[1]. Thus, loading tuples based on clustered digest results efficiently avoids duplicate page loading requests. However, tuples fetched according to their page addresses are generally not clustered on the join key. Hence, we have another temporary table called *join candidate table* to store the tuples fetched according to the reloading instruction table. Sort-merge join or hash join algorithm can be then applied on this table for producing final join results. This strategy successfully schedules an optimal page loading sequence to finish all tuple joins, at the cost of introducing some extra I/O cost for maintaining those two temporary tables. We summarize it in Algorithm 3.

Input : Sorted digest table dt_1, sorted digest table dt_2
Output: Join results

while *joining* dt_1 *and* dt_2 **do**
  if *there is a join result* $< join\_key, tid_1, tid_2 >$ **then**
    Output tid_1, tid_2 to reloading instruction tables rt_1 and rt_2
Sort rt_1 and rt_2 based on *page id*
while *scanning* rt_1 *and* rt_2 **do**
  Fetch clustered tuples according to tids Output the tuples to join candidate tables jct_1 and jct_2
Perform sort-merge join or hash join on jct_1 and jct_2

Algorithm 3: Page-based Loading

3.4 Graph-based Loading Strategy

Instead of archiving the digest join results in temporary tables, an alternative method is to archive it with the join graph, the one that we use to model the page loading problem. If the memory can hold all digest join results in the form of a join graph, we may find good heuristics to travel all edges to do the page loading and joining. In the literature, there are two heuristics in that direction [1, 11]. Their studies focused on page fetch scheduling for index-based joins. The basic idea behind these heuristics is intuitive – each time it selects a subgraph which at least contains all edges of one vertex but requires the fewest non-resident pages to join. Here, a non-resident page is a page that does not reside in the memory. In other words, it is suggested to check all subgraphs in the current join graph: identifying whether it contains all edges of any vertex and counting how many vertices (i.e., pages) are not in the memory. Then the subgraph satisfying the above conditions is selected to join. As iterating all subgraphs of the join graph makes the computational cost prohibitively high, an approximation method is to select the vertex with the fewest non-resident neighbors, together with its neighbors in the join graph (called segment) [1].

However, it is impractical to directly apply those heuristics in our scenario. One reason is due to the constrained memory. As our research focuses on joining large tables, the memory space is not likely to hold the whole join graph. Furthermore, in order to avoid producing duplicate join results, we have to attach tids to the edges in the join graph in implementation. Consider a case when only page ids are attached. Suppose we have an edge indicating a join between page a of table X and page b of table Y in the join graph. When the memory space for storing the join graph runs out, we conduct the join on pages a and b to reclaim space by removing the corresponding vertices and edge from the join graph. Next time when another digest join result requiring page a to join with page b comes, because there is no data recording what results have already been produced, duplicate join results might be generated. Apparently, attaching tids to the edges can address this issue. However, it makes the join graph to grow even faster and aggravates the shortage of memory.

Another reason is that those existing heuristics only provide a rough bound on the required size of page cache. In particular, it only guarantees that the maximum size of the required page cache will not exceed $min\{sizeof(X), sizeof(Y)\} + 1$ when joining tables X and Y. This bound is generally not very useful, because if the memory is large enough to hold one table in the memory, we can just load that table into the memory and use the remaining memory space to load the other table to do join. In this case, the simple nested-loops join algorithm is surely to achieve optimal I/O performance.

In view of the above issues, we propose the *graph-based loading strategy* which aims at achieving acceptable performance even when the memory is highly limited. We divide the memory space into two parts: one for storing the join graph (i.e., join graph storage) and the other for caching loaded pages (i.e., page cache). We dynamically manage the memory space for the join graph storage and the page cache. When a join result from the digest table comes, if there is space, we directly add it into the join graph. Otherwise, we adjust the space of join graph based on its *required storage size* (RSS) and *required cache size* (RCS). The RSS of a join graph is equal to how many pages of space are required to hold this join graph. We organize the join graph

[1]The page addresses can be inferred from the logical tuple ids.

```
Input  : Sorted digest table dt1, sorted digest table dt2
Output: Join results
Allocate page p1 for dt1, p2 for dt2
Allocate rest pages for PageCache and JoinGraph, and set
JoinGraph initial size to be one page
while joining dt1 and dt2 with p1 and p2 do
    if there is a join result < join_key, tid1, tid2 > then
        Try join tuples with tid1 & tid2 only with PageCache
        if join succeeds then
            continue to next digest join result
        Try add < join_key, tid1, tid2 > to JoinGraph
        if fail then
            /* insufficient space for JoinGraph */
            rcs = RequiredCacheSize(JoinGraph)
            if rcs < current cache size then
                Increase JoinGraph size by one page
                Decrease PageCache size by one page
                Add < join_key, tid1, tid2 > to JoinGraph
            else
                seg = SelectSegment(JoinGraph)
                LoadAndJoin(seg)
                Update in-cache flags of JoinGraph
                Remove seg from JoinGraph
                Add < join_key, tid1, tid2 > to JoinGraph
                rss = RequiredStoreSize(JoinGraph)
                if rss > current JoinGraph size then
                    Decrease JoinGraph size by one page
                    Increase PageCache size by one page
```

Algorithm 4: Graph-based Loading

in an adjacent list data structure, and the RSS is larger than the size of the adjacent list, as extra space is needed to store the tids for the edges. The RCS of a join graph is the minimum cache size we should maintain for loading and joining any segment of this join graph. Initially, the join graph only takes up one page and the page cache spans the rest of the memory. Throughout the join process, we dynamically adjust the sizes of join graph and page cache. Specifically, when we fail to insert a join result from the digest table to the join graph:

- If RCS $<$ the current page cache size, we enlarge the memory space for join graph storage (correspondingly, shrink the space of the page cache by removing some cached pages) to insert that result.
- Otherwise, we try to select a segment of the join graph to load and join using the page cache. After that, we insert the digest join result, and check RSS to see whether we need to enlarge the space for the page cache (correspondingly, shrink the memory space of the join graph).

We summarize the graph-based loading strategy in Algorithm 4.

In Algorithm 4, there are several important functions. We discuss them one by one below.

- *RequiredStoreSize()*: This function computes the storage requirement of the current join graph measured in pages, because in DBMS only page-oriented memory management is supported.
- *SelectSegment()*: This function selects a segment from the join graph. We follow the same approximation in [1]. Because there may exist edges having both vertices (i.e., pages) in cache due to the selected segment, we actually select more edges than that approximation to join as many pages as possible.
- *LoadAndJoin()*: This function accepts a segment of the join graph, loads corresponding pages from original tables and joins the tuples indicated by tids. When the cache is fully occupied, we have to swap some pages out. In particular, we have to decide the order to load pages in the selected segment and select the victim pages in the cache to be swapped out. In our implementation, our policies are as follows: 1) We select to load the page in the segment with maximum resident degree, which is defined as the number of its in-cache neighbors in the join graph. 2) We select the page in the cache with minimum non-resident degree as victim. The non-resident degree of a page is defined as the number of its not-in-cache neighbors in the join graph. Note that the resident degree and non-resident degree are updated accordingly after each page loading or swapping.
- *RequiredCacheSize()*: As we follow the same approximation of [1] in *SelectSegment()* function, intuitively we may assign the value of RCS as the max degree of vertex in the current join graph plus one. However, as we tend to select more edges than the approximation, it is possible that k times of that value makes more sense, where k is a constant whose value needs to be decided in practice.

4 Performance Evaluation

We implemented *DigestJoin* with all three page loading strategies in an experimental database system built on top of a 16GB Mtron SSD[2]. The experimental database system is designed to enable easy evaluation on the performance of different join algorithms. Its raw storage manager maintains a bunch of pre-allocated continuous space on the SSD and provides a page-oriented read/write interface. The system also maintains some statistics information, such as system parameters (e.g., page size, memory size, etc.), table summaries (e.g., # tuples, # pages, etc.), as well as join statistics (e.g., join selectivity).

[2]Mtron SSD Mobi 3000 serials, 16GB, MSD-SATA3025

Tables are organized on page-oriented space of the SSD. Each page in our evaluation is 4K bytes. The data tuples are stored on the pages following a row-based storage scheme. Since we are concerned with non-index-based joins, we do not build any index for each table. To save space, we store data tuples in variable-size format, so that the actual ratio of the size of the digest table to that of the original table may not be as high as we indicated in Section 2. In our evaluation the average tuples per page is 25 and the ratio of tuple size to digest entry size is 10. Finally, the data tuples are imported into the storage in a random order. This is to simulate a general-case join where the join key values could be in any arbitrary order in the original tables.

In our evaluation, the test dataset is taken from TPC-H benchmark. In particular, we use two `CUSTOMER` tables (each is 32MB in 8192 pages) of TPC-H and do the natural join on them through the key `C_CUSTKEY`. This will produce a 100% selectivity. In our implementation, we install a filter function before we are about to get a join result. The filter function will flip a coin and decide whether to drop the result or not. By controlling the probability in flipping the coin, we can simulate different selectivity settings.

We use the traditional sort-merge join algorithm as a representative to compare with the *DigestJoin* method, while expecting that similar performance results can be observed for nested-loop and hash joins. In detail, the algorithms under evaluation include: traditional sort-merge join (Basic), *DigestJoin* with naive page loading strategy (Digest(Naive)), *DigestJoin* with page-based reloading strategy (Digest(Page)), and *DigestJoin* with graph-based reloading strategy (Digest(Graph)). We run all the experiments on a desktop PC equipped with a Core 2 Quad Q6600 CPU and 4GB main memory.

We choose the proper configuration of parameter k for the graph-based page loading strategy based on a set of experiments. Actually we vary k from 0.5 to 4. As we observed, when the parameter k is increased from 1 to 4, the I/O cost also increases. Because the larger is the value k, the smaller memory space is dedicated to the join graph, and as a result, the join graph may not grow large enough and thus degrade the performance of page loading. On the other hand, the value k should not be set smaller than 1. When it is set at 0.5, the memory space for caching loaded pages is too small, which leads to frequent page swaps, and many pages are repeatedly reloaded. Therefore, we set k at 1 for the graph-based page loading strategy in the rest evaluation of *DigestJoin* method.

We evaluate different join methods and page loading strategies, and investigate how the join selectivity would affect their performance. We define the *join selectivity* as the percentage of tuples having join results(e.g., given a 1000-page table, if 100 pages of tuples are involved in the final join results, then the selectivity is 10%). It indicates how many join results are going to be produced, so directly affects the tuple reloading cost. Figures 4 and 5 show the performance comparison with join selectivity changing from 0 to 0.1 and from 0.1 to 0.5, respectively. The results do not count the I/O cost to output final join results, since usually they are not output to the secondary storage and this cost is the same for all join algorithms. As such, the result of Basic remains the same over different selectivity settings. It is used as the baseline in the performance analysis.

As can be seen from Figure 4, Digest(Naive) outperforms Basic only when the selectivity is very low (i.e., < 0.08). When the selectivity is higher than 0.08, as the naive page loading strategy brings too many duplicate page accesses, it becomes worse than Basic. Similarly, Digest(Page) has the best performance among all join algorithms under a low selectivity, but its performance degrades when the selectivity increases. When the selectivity is higher than 0.3 (see Figure 5), it performs worse than Basic. Overall, Digest(Graph) gets the best performance and outperforms Basic by 15%-64% for all cases tested. As predicted, because of the increased page loading cost, its performance improvement becomes smaller when the selectivity increases.

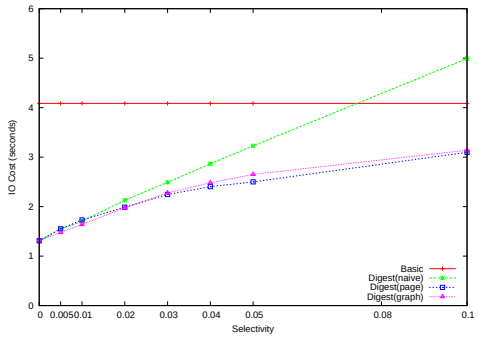


Figure 4. DigestJoin vs. Traditional Sort-Merge Join under Low Selectivities (Uniform Join Distribution)

5 Related Work

Data management on flash-based storage media has attracted a lot of research interest in recent years. Because of the unique characteristics of flash memory, early work focused on issues in simulating traditional hard disk with flash memory chips, such as [3, 2, 4]. On top of these fundamental achievements, recent work started to exploit the characteristics of flash memory to enhance the performance of DBMS applications. In view of the asymmetric read/write speed and the erase-before-write limitation, Wu et al. [13] proposed log-based indexing scheme for flash

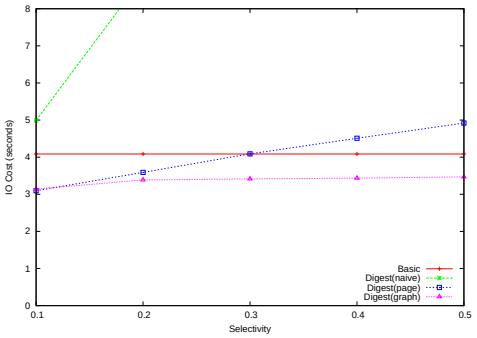


Figure 5. DigestJoin vs. Traditional Sort-Merge Join under High Selectivities (Uniform Join Distribution)

memory. Observing that the log-based indexing scheme is not suitable for read-intensive workload on some flash devices, Nath and Kansal [10] developed an adaptive indexing method that adapts to the workload and storage device. Lee and Moon [6] presented a novel design of data logging called in-page logging (IPL). More recently, Shah et al. [12] presented a new storage strategy for fast scanning and joining on flash drives. However, all the prior work did not take advantage of the fast random access speed of the flash storage. To the best of our knowledge, our work is the first attempt to utilize this characteristic to speed up traditional join algorithms on top of flash-based storage media.

Join is one of the important query operators in relational DBMS. Extensive research efforts have been put to the optimization of join processing [9]. The join algorithms can be classified as non-index-based (the focus of this paper) and index-based. For non-index-based joins, both of the external sorting (in sort-merge join) and hashing (in hash join) requires a considerable number of I/O operations on the secondary storage, which is expensive random I/Os on magnetic disk, so they try to involve only sequential read/write operations. As flash has a more faster random read performance, in this paper we explore the opportunity to further improve these non-indexed-based join algorithms for flash-based storage media.

For index-based join algorithms, by accessing the indexes of the join attribute, a list of tuple pairs that participate in the join is first composed. Then, the tuples themselves are fetched and physically joined. Such techniques face a problem of how to generate an optimal page fetching schedule which fulfills all page join requests with minimum number of page accesses. Merrett et al. [8] studied this problem and proved that it is *NP-complete*. After that a number of heuristics have been developed, presented in [5], [11] and [1]. In this paper, we address a similar page loading issue for the proposed *DigestJoin* method and develop several efficient memory-conscious page loading strategies.

6 Conclusion

In this paper, we have proposed a new join method called *DigestJoin* by exploiting the fast random read of flash-based storage media. *DigestJoin* is a generic join method as its implementation invokes traditional join algorithms. For the *DigestJoin* method, we have identified a critical issue of page loading, which is shown to be at least *NP-complete* by a graph formulation. Three heuristic-based strategies, namely naive, page-based, and graph-based loading, have been developed. We have also implemented the *DigestJoin* method with the three page loading strategies in an experimental database system on top of a real Mtron SSD. The evaluation results show that the *DigestJoin* method generally improves the traditional join algorithm under various system parameter settings.

References

[1] C. Y. Chan and B. C. Ooi. Efficient scheduling of page access in index-based join processing. *IEEE Trans. on Knowl. and Data Eng.*, 9(6):1005–1011, 1997.

[2] L. Chang. On efficient wear leveling for large-scale flash-memory storage systems. In *SAC'07*, pages 1126–1130, 2007.

[3] L. Chang, T. Kuo, and S. Lo. Real-time garbage collection for flash-memory storage systems of real-time embedded systems. *Trans. on Embedded Computing Sys.*, 3(4):837–863, 2004.

[4] Y. Chang, J. Hsieh, and T. Kuo. Endurance enhancement of flash-memory storage systems: an efficient static wear leveling design. In *DAC'07*, pages 212–217, 2007.

[5] F. Fotouhi and S. Pramanik. Optimal secondary storage access sequence for performing relational join. *IEEE Trans. on Knowl. and Data Eng.*, 1(3):318–328, 1989.

[6] S. Lee and B. Moon. Design of flash-based dbms: an in-page logging approach. In *SIGMOD'07*, pages 55–66, 2007.

[7] S.-W. Lee, B. Moon, C. Park, J.-M. Kim, and S.-W. Kim. A case for flash memory ssd in enterprise database applications. In *SIGMOD'08*, pages 1075–1086, 2008.

[8] T. H. Merrett, Y. Kambayashi, and H. Yasuura. Scheduling of page-fetches in join operations. In *VLDB'81*, pages 488–498, 1981.

[9] P. Mishra and M. H. Eich. Join processing in relational databases. *ACM Comput. Surv.*, 24(1):63–113, 1992.

[10] S. Nath and A. Kansal. Flashdb: Dynamic self-tuning database for nand flash. Technical Report MSR-TR-2006-168, Microsoft Research, 2006.

[11] E. R. Omiecinski. Heuristics for join processing using non-clustered indexes. *IEEE Trans. Softw. Eng.*, 15(1):18–25, 1989.

[12] M. A. Shah, S. Harizopoulos, J. L. Wiener, and G. Graefe. Fast scans and joins using flash drives. In *DaMoN'08*, pages 17–24, 2008.

[13] C. Wu, T. Kuo, and L. P. Chang. An efficient b-tree layer implementation for flash-memory storage systems. *Trans. on Embedded Computing Sys.*, 6(3):19, 2007.

A Survey on Mechanism Design in Distributed Systems

Xiaowei CHEN

Abstract

Mechanism design is the sub-field of microeconomics and game theory that considers how to implement good system-wide solutions to problems that involve multiple self-interested agents, each with private information about their preferences. It is widely used in many important applications in distributed systems. Distributed Algorithmic Mechanism Design (DAMD) combines theoretical computer science's traditional focus on computational tractability with its more recent interest in incentive compatibility and distributed computing. The Internet's decentralized nature, in which distributed computation and autonomous agents prevail, makes DAMD a very natural approach for many Internet problems. This paper first outlines the basics of MD, AMD, DAMD and then reviews previous DAMD results on multicast cost sharing and interdomain routing. Then it discusses several possible DAMD applications in distributed systems, especially task and resource allocation problems. The remainder of the paper poses some general open problems.

1. Introduction

With the advent of the Internet, we are seeing the design and deployment of large-scale distributed systems. Relevant examples are grid computing, semantic web, pervasive computing, e-commerce, mobile computing and peer-to-peer (P2P) systems. The deployed systems include large-scale parallel computing projects such as SETI@home [1] and factoring [2], Internet services such as file sharing and web caching, and systems to support the basic network functionality, such as routing and congestion control. Usually, the primary focus of research on these distributed systems is the development of good distributed algorithms, i.e., algorithms with low computational complexity and communication requirements.

For these complex systems, agent-based approaches, which emphasize autonomous actions and flexible interactions, are natural computational models. When designing such multiagent systems (MASs), it requires that incentive compatibility and computational tractability are jointly addressed. Mechanism design theory became relevant for a wide variety of applications only after Hurwicz (1972) introduced the key notion of incentive-compatibility, which allows the analysis to incorporate the incentives of self-interested participants. In particular, it enables a rigorous analysis of economies where agents are self-interested and have relevant private information.

Most of these distributed programs run on computers at many different locations on the Internet, owned and operated by self-interested parties. These selfish agents involved in the system cannot be relied on to blindly follow any prescribed algorithm. Whenever possible, they might strategize for their own gain. Thus, mechanism design (MD), a large area of research in economics, aims to structure incentives so as to induce the desired behavior of selfish agents.

Informally, a mechanism is a system with the following form: Each agent can select a strategy from an allowed range of strategies. The system then processes all the agents' selected strategies and outputs an outcome as well as monetary payments to (or receipts from) each agent. Economic mechanism design focuses on issues of incentive and strategy and largely ignores computational considerations. The algorithmic mechanism design (AMD) approach of Nisan and Ronen [3] combines the two considerations. Feigenbaum, Papadimitriou, and Shenker [4] extended the Nisan-Ronen framework to include distributed computation. They initiated the distributed algorithmic mechanism design (DAMD) approach to designing systems for the Internet, which combines the incentive-compatibility considerations of mechanism design with the distributed-computing objective of designing systems with modest computation and communication requirements.

The Internet is an arena in which incentive compatibility, distributed computation, and computational complexityare all highly relevant. Thus, we believe that DAMD, with its simultaneous attention to these issues, will be important for understanding our distributed systems' future. This paper is intended to provide a basic overview of DAMD and to identify several promising areas for future research. We start, in Section 2, by providing some necessary background on mechanism design (MD), algorithmic and otherwise. In Section 3, we review some previous DAMD results on multicast cost sharing and interdomain routing. The next section is devoted to exploring the current promising applications of DAMD. Last, we pose some open problems, some very specific and others quite general, concerning the foundations and applications of DAMD. Additional material about the AMD and DAMD research agendas can be found in [5, 6, 7].

2. MD to AMD to DAMD

2.1 Mechanism Design

In essence, game theory is the study of what happens when independent agents act selfishly. Mechanism design is the branch of game theory which reconciles private interests with social goals. It asks how one can design systems so that agents' selfish behavior results in the desired system-wide goals. The "mechanisms" in this field are output specifications and payments to agents that incent them to behave in ways that lead to the desired system-wide result.

Figure 1 depicts a simple mechanism setting: There are n agents; each agent i has some private information, called her private type t_i. This type is drawn from a set T of possible types; the set T is known to all, but t_i is known only to agent i. Agent i chooses strategy a_i to maximize her welfare. For example, the agents could be bidders at an auction; in this case, the private type of an agent is the amount it is willing to pay for the item being auctioned. We use t to denote the vector $(t_1; t_2; \ldots ; t_n)$.

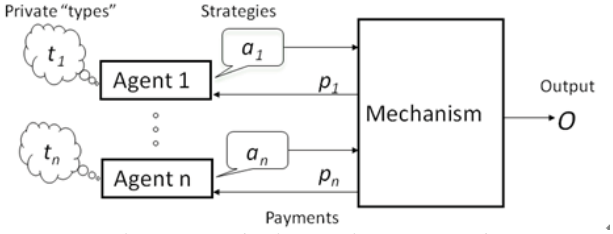


Figure 1. Mechanism Design Framework

The function of a mechanism in this setting is to solve a decision or allocation problem that affects all the agents. There is a set O of possible decisions (or allocations), and the mechanism must pick some decision $o \in O$.

More formally, each of the n autonomous strategic agents has a utility function u_i: $O \rightarrow R$, where $u_i \in U$, that expresses its preferences over these outcomes. The desired system-wide goals are specified by social choice function (SCF) F: $U^n \rightarrow O$ that maps each particular instantiation of agents (who are completely described by their utility functions) into a particular outcome. The problem is that these utilities are known only to the agents, not to the system designer or to any other central administrative entity; thus, one cannot just implement the desired outcome by fiat.

Intuitively, a mechanism design problem has two components: the usual algorithmic output specification and descriptions of what the participating agents want, formally given as utility functions over the set of possible outputs. The goal in traditional MD is to design a system in which rational agents interact in a way that leads to equilibriums with desired systemwide properties. These properties are encapsulated in the SCF, which defines a desired outcome for each possible set of agent types. SCF describes the properties that the designer would like MAS outcomes to possess. Given this, the designer's problem is to provide incentives so that agents choose to act in a way that implements a particular SCF. Here are some typical desired properties (desiderata) of SCFs:

1. Pareto optimality: Implementing an outcomethat is not Pareto-dominated by any other outcome, so no other outcomes make one agent better-off while making other agents worse-off.
2. Maximized social welfare: Implementing an outcome that maximizes the total utility across agents. This is often called the efficient outcome.
3. Maximized utility to a particular agent: Maximizing the expected utility to a single agent, typically the center, across all possible mechanisms. A common setting is a revenue-maximizing auction, in which the goal is to design a mechanism that maximizes the auctioneer's revenue.
4. Budget balance: The total payment that agents make equals exactly zero (a strict budget balance), so money isn't injected into or removed from a system. Or, the total payment is nonnegative (a weak budget balance), so the mechanism doesn't run at a loss. We can also consider an *ex ante* budget balance, in which the mechanism is balanced on average, and an *ex post* budget balance, in which the mechanism is balanced at all times, for all instances.
5. Individual rationality: The SCF gives each agent nonnegative utility in equilibrium. We can consider interim individual rationality, in which an agent has nonnegative utility in expectation given its own type, and ex post individual rationality, in which an agent always has nonnegative utility.

Budget balance is especially important in systems that must be self-sustaining and require no external benefactor to input money or central authority to collect payments. Yet, budget balance often conflicts with other desiderata, such as efficiency. Individual rationality is important when you can't force agents to participate in a mechanism.

The key concpets of mechanism are: Revelation Principle, Incentive-Compatibility and Implementation Theory. The central mechanism design question is that which social goals can be achieved for given solution concept. While mechanism design involves esoteric game-theoretic issues, here we will avoid them as much as possible and only present mechanism design content relevant to distributed algorithmic mechanism design.

2.1.1 The Revelation Principle

Definition 1: Dominant Strategy

A mechanism is an implementation of dominant strategies if for each agent a_i and each type t_i there exists a strategy $s_i \in S_i$, termed as dominant strategy, such that for all possible strategies of the other agents a_{-i} (i.e. $\{A/a_i\}$), s_i maximizes agent a_i's utility.

Definition 2: Truthful Implementation

A mechanism is truthful if for each agent a_i and all its t_i, $S_i = T_i$, i.e., its dominant strategy is to report its real type.

Definition 3: Strongly Truthful Implementation

A mechanism is a strongly truthful implementation if truth-telling is the only dominant strategy for agents.

2.1.2 Incentive-Compatibility Mechanism

Given solution concepts and SCF desiderata, the central question in MD is, which set of desiderata can be implemented in a MAS's game-theoretic equilibrium, given that the agents are assumed to be self-interested? In this context, the revelation principle is a key concept when it comes to generating impossibility and possibility results.

The revelation principle states that any mechanism can be transformed into an incentive-compatibility (IC), direct-revelation mechanism (DRM). In this context, "direct" means the agents' strategy space is restricted to reporting their types and "incentive compatible" means the equilibrium strategy for agents is truth-telling. In the special case of a DRM in which truth revelation is a dominant strategy, we say the mechanism is strategyproof.

The revelation principle is important in MD for two reasons:

1. Theoretical: It allows a focus on IC DRMs for the development of impossibility and possibility results.
2. Practical: The properties that an IC DRM can satisfy can provide a normative guide for the outcome and payments that a realized implementation must compute. This mechanism need not itself be a DRM and can have better computational properties than the original mechanism.

Formally, an SCF is strategyproof if $u_i\ (F(u)) \geqslant u_i\ (F(u|^i v))$, for all i and all $v \in U$, where we use the notation $(u|^i v)_i = v$ and $(u|^i v)_j = u_j$, for all $j \neq i$. If F is strategyproof, then no agent has an incentive to lie, and the desired social goals can be achieved byasking agents to reveal their utility functions. Mechanisms in which agents are asked to directly reveal their utility functions are call direct mechanisms.

An SCF is group-strategyproof if the following holds for all S, u, and u' (where $S=\{i \mid u_i \neq u_i'\}$ is the defecting group): Either $u_i(F(u)) = u_i(F(u'))$, $\forall i \in S$, or $\exists i \in S$ for which $u_i(F(u')) < u_i(F(u))$. That is, if any agent in the group benefits from the group's colluding and lying to the mechanism, then at least one agent in the group suffers.

An important class of problems are those in which the utilities are quasilinear, and the outcome space O factors into a set of system states $\tilde{O}$ and a set of payment states $P \subseteq R^n$ that represent a vector of payoffs (or charges). For such problems, there is a class of strategyproof mechanisms, called Vickrey-Clarke-Groves (VCG) mechanisms [8, 9, 10], that result in the system state that optimizes $\sum_i v_i(\tilde{o})$.

Direct strategyproof mechanisms provide a conceptually simple, if not always ideal, way to achieve strategyproof SCFs. However, there are many cases in which the desired result, i.e., the desired social choice function F, is not strategyproof. To describe how to realize such non-strategyproof SCFs, we use indirect mechanisms. Traditional game theory often uses Nash-equilibrium solution concept, i.e., selfish playis assumed to result in strategy vectors in which no agent can unilaterallyincrease his utility. Other solution concepts include rationalizable strategies (agents use strategies that are best responses to rational beliefs about the other agents' strategyc hoices [11, 12]), evolutionarily stable strategies (agents imitate the successful strategies used byothers in previous rounds of the game [13]), and dominant strategies (agents onlyc hoose strategies that, regardless of how other agents play, never result in lower payoffs than any other strategy). To date, most of the AMD and DAMD literature uses the dominant-strategy solution concept.

2.1.3 Implementation Theory

The revelation principle is extremely useful. However, it does not address the issue of multiple equilibria. That is, although an optimal outcome may be achieved in one equilibrium, other, sub-optimal, equilibria may also exist. There is, then, the danger that the participants might end up playing such a sub-optimal equilibrium. Can a mechanism be designed so that all its equilibria are optimal? The first general solution to this problem was given by Eric Maskin. The resulting theory, known as implementation theory, is a key part of modern mechanism design.

In the 1970s, the formulation of the so-called revelation principle and the development of implementation theory led to great advances in the theory of mechanism design. It is important to note that, although the mechanism is chosen by the system designer, the solution concept is supposed to reflect reality. The solution concept thus depends greatly on the context (e.g., is it a repeated game or a single-shot game, do agents collude, do they know about the other agents, do they know about the other agents' strategic choices, etc.). Because the Internet is somewhat different from traditional game-theoretic contexts, the traditional solution concepts maynot be sufficient.

2.2 Algorithmic Mechanism Design

The game-theory literature on mechanism design does not consider computational and communication

complexity, and many of the existence proofs rely on extremely impractical mechanisms. For the mechanism-design approach to have any practical relevance for Internet computation, one must focus on scalable algorithms. That is, the function must be computable with reasonable computational and communication resources.

Nisan and Ronen [3] initiated the study of AMD by adding computational tractabilityto the set of concerns that must be addressed in the design of incentive-compatible mechanisms. Succinctly stated, Nisan and Ronen's contribution to the mechanism-design framework is the notion of a (centralized) polynomial-time mechanism. They also provide strategyproof, polynomial-time VCG mechanisms for some concrete problems, including lowest-cost paths and task allocation.

The original paper of Nisan and Ronen [3] sparked a large body of research on algorithmic aspects of mechanism design. In particular, there is growing interest in incentive compatibility in both distributed and centralized computation in the theoretical computer science community (e.g., [14, 4, 15, 16, 17, 18]) and in the "distributed agents" part of the AI community (e.g., [19, 20, 21, 22, 23, 24]).

2.3 Distributed Algorithmic Mechanism Design

The centralized computational model of [3] is not adequate for the study of Internet computation, where not only are the agents distributed, but so are the resources (e.g., link bandwidth and cache storage) and the computational nodes. One of the main motivations for algorithmic mechanism design is the study of Internet mechanisms. Internet-based mechanisms involve distributed algorithms and any measure of their computational feasibility must reflect their distributed nature.

In the Internet, the agents are often dispersed across the network; thus, the input and output of the mechanism must occur at dispersed locations. One way to compute the mechanism is to send all the input strategies to a single location, compute the output and payments in a centralized manner, and then send the required information back to the agent locations. However, this approach may require prohibitively high communication and it may cause congestion near the centralized server. This led Feigenbaum, Papadimitriou, and Shenker to consider distributed computational models in their paper on multicast costsharing mechanisms [4]; this started the study of distributed algorithmic mechanism design. Feigenbaum et al. pointed out that for a mechanism to be feasible in an Internet setting, it must be computable by a distributed algorithm with low computational complexity and modest communication requirements. More specifically, the distributed algorithm should ideally have the following properties:

1. The local computations require polynomial-time.
2. Low communication complexity; the total number of messages sent is ideally $O(s)$.
3. Each message is reasonably small, e.g., polylog(s).
4. No single link is congested, i.e., the maximum number of messages on a link is $O(1)$.

Here, s denotes the input size of a given instance of the problem. They introduced the term network complexity to cover these four aspects of a distributed algorithm. A mechanism is said to have "good network complexity" if it satisfies all these properties.

3. Applied Scenarios in DAMD

The DAMD approach is relevant to several problems of practical importance. For instance, as we discuss in Section 4, problems of web caching, peer-to-peer systems, overlay networks, and task and resource allocation involve distributed computing by many selfish agents. In this section, we simply introduce two specific scenarios in which DAMD has been applied: multicast cost sharing, which exercises the notion of absolute network complexity, and interdomain routing, which exercises the notion of BGP compatibility.

3.1 Multicast Cost Sharing

In the standard unicast model of Internet transmission, each packet is sent to a single destination. Although unicast service has great utility and widespread applicability, it cannot efficiently transmit popular content, such as movies or concerts, to a large number of receivers; the source would have to transmit a separate copy of the content to each receiver independently. The multicast model of Internet transmission relieves this problem by setting up a shared delivery tree spanning all the receivers; packets sent down this tree are replicated at branch points so that no more than one copy of each packet traverses each link. Multicast thus greatly reduces the transmission costs involved in reaching large user populations.

The large-scale, high-bandwidth multicast transmissions required for movies and other potential sources of revenue are likely to incur substantial transmission costs. The costs when using the unicast transmission model are separable in that the total cost of the transmission is merely the sum of the costs of transmission to each receiver. Multicast's use of a shared delivery tree greatly reduces the overall transmission costs, but, because the total cost is now a submodular and nonlinear function of the set of receivers, it is not clear how to share the costs among the receivers. A series of papers has addressed the problem of cost sharing for Internet multicast transmissions. In the first paper on the

topic, Herzog et al. [25] considered axiomatic and implementation aspects of the problem. Subsequently, Moulin and Shenker [26] studied the problem from a purely economic point of view. Several papers [4, 27, 15, 28] adopt the distributed algorithmic mechanism design approach, which augments a game-theoretic perspective with distributed computational concerns.

Although the multicast cost-sharing problem has been quite useful in establishing the basic conceptual foundations of DAMD, it is neither realistically formulated nor of pressing importance. Interdomain routing, our next example, is both more realistic and more important.

3.2 Interdomain Routing

The Internet is comprised of many separate administrative domains or Autonomous Systems (ASs). Routing between these domains, e.g., interdomain routing, is currently handled by the Border Gateway Protocol (BGP). There has been much research on routing in general and BGP in particular, but most of it takes a traditional protocol-design approach. Feigenbaum, Papadimitriou, and Shenker [29] focused on DAMD issues inherent in interdomain routing.

The basic incentive problem involves transit traffic, i.e., traffic neither originating from nor destined to the AS that is currently carrying the packets. For the overall efficiency of the network, packets should travel along shortest or, more generally, lowest-cost paths (LCPs). These optimal paths would typically, in general networks, cut across several ASs. However, carrying transit traffic is a burden that ASs would prefer to avoid. The basic problem is simple: Overall system efficiency is maximized when ASs accept transit traffic, but individual domains are happiest when they carry no transit traffic at all.

In the model of Feigenbaum et al. [29], which is an extension of an earlier (centralized) LCP-mechanism model proposed by Nisan and Ronen [3] and studied further by Hershberger and Suri [16], each AS incurs a per-packet cost for carrying traffic, where the cost represents the additional load imposed on the internal AS network bythis traffic. Furthermore, the model also assumes that, to compensate for these incurred costs, each AS is paid a price for carrying transit traffic. The goal is to maximize network efficiency by routing packets along the LCPs. Standard routing protocols (such as BGP) can compute LCPs given a set of AS costs. However, under many pricing schemes, an AS would be better off lying about its costs; such lying would cause traffic to take non-optimal routes and thereby interfere with overall network efficiency.

To prevent this, one needs the pricing scheme to be strategyproof, so that ASs have no incentive to lie about their costs. The pricing scheme should also have the reasonable property that ASs that carry no transit traffic at all receive no payment. It is shown in [29] that there is only one strategyproof pricing scheme with this property; it is a member of the VCG family. Moreover, a BGP-compatible distributed algorithm is given that computes these prices. This algorithm requires only minor and straightforward modifications of the BGP computational model given by Griffin and Wilfong [30]. Specifically, the algorithm in [29] requires a small constant-factor increase in both the table sizes and the message sizes of BGP, but it does not require any new messages or any new infrastructural or computational capability; in particular, all messages are still sent between neighbors in the AS graph. Similarly, the local computation done by a node in each stage (i.e., between receiving an updated table from a neighbor and, if necessary, sending an update to each of its neighbors) is the same order of magnitude as the BGP local-computation time.

The results on multicast cost sharing and interdomain routing represent the two most successful applications to date of DAMD to practical network problems. In section 4, we return to specific Internet-based problems in which DAMD maybe applicable.

4. DAMD in Distributed Systems

In this section, we turn our attention from general and foundational issues in DAMD to specific mechanism-design challenges now faced by Internet researchers. We select main four possible application areas to discuss: web caching, peer-to-peer file sharing, overlaynet works, and distributed task and resource allocation. We do not attempt to provide complete references to the vast literatures on each of these subjects.

4.1 Web Caching

Web caches are an important tool for enhancing the performance of web access; they are used to eliminate hot spots in the network and to reduce access latencies. A web-caching architecture provides the framework for a set of collaborating caches to interact with each other and serve a client community. A wide variety of caching architectures have been proposed; their common intent is to achieve overall system efficiency, and their common assumption is that caches are obedient. This assumption maybe valid when the entire caching infrastructure belongs to the same administrative entity, but, when the caching infrastructure spans administrative boundaries, the caches might deviate from the protocols to maximize their individual welfare.

There are two fundamental incentive issues in caching. First, when considering a single cache, the utility of the requesting clients would be maximized by caching the

frequently requested pages that provide the highest user utility. However, the only way that a cache can learn about these valuations is from the clients themselves. One needs strategyproof mechanisms to elicit truthful valuations from clients.

Second, caches have limited resources (bandwidth and storage) and incur a cost for storing a page or serving a client request. When there are several caches collaborating, but they are managed by separate economic entities, there is a question of how to design mechanisms to distribute the caching load. If the caches are not reimbursed for the cost of serving pages, they have an incentive to manipulate other caches into serving the pages and therebya void bearing the operating costs. If the caches receive payments for serving pages, they might "compete" with each other to serve the highest-value pages, therebyduplicating each other's content, leaving some important pages uncached, and yielding sub-optimal performance. Thus, the system needs to provide incentives designed to have the caches report their true operating costs.

Develop distributed algorithmic mechanisms for caching in which clients are induced to reveal their true preferences, and caches are induced to implement the optimal resource allocation. These caching mechanisms should be built as scalable extensions to existing distributed caching protocols and provide provable performance guarantees. Some aspects of this problem are addressed in, e.g., [31].

4.2 Peer-to-Peer File Sharing

Peer-to-peer (P2P) file-sharing networks, e.g., Gnutella, KaZaA, and Freenet, are the *ne plus ultra* of autonomous distributed systems; each machine belongs to a different user, and there is no central administrative authority. Thus, incentive issues are likely to be very important to the future of P2P technology; this provides a great opportunity for DAMD research.

P2P file-sharing represents a shift from purely commercial content-distribution systems (e.g., content providers paying CDNs to distribute their content) to a "gift economy," in which individual users offer up their resources - content, access bandwidth, storage, and CPU - for the greater good. However, initial studies show that there is a serious "free rider problem" [32]. It maybe that, without some systematic way of providing incentives for users to share their files, these P2P systems will become increasingly centralized, with only a few commerciallysupp orted nodes offering to share files.

One initial attempt to provide incentives to participants is the MojoNation P2P system, which awards users "mojo" for offering resources [33]. "Mojo" can be used to gain priority access when the system is overloaded.

While the original P2P systems operate as gift economies, MojoNation can be thought of as a barter economy, in which mojo is exchanged in return for services, but no money is transferred. The exchange of money would allow for a much wider range of possible economic mechanisms. The question is whether this extension to monetary exchanges would result in superior performance.

Some models of P2P systems consider a collection of identical nodes. Although this is theoretically appealing, it is contradicted by preliminary measurement studies suggesting that there is a very wide range of node capabilities (in bandwidth, CPU, and disk) in P2P systems [34]. Wide heterogeneities maylead to significantly increased efficiency in P2P systems; that is, the highly capable nodes can act as semi-centralized repositories.

Most of models address the extent to which users share files with other P2P users. But in fully decentralized P2P systems, nodes function both as caches of shared files and as routers for queries destined for other nodes. In that sense, the P2P-incentive issues are a union of the issues in routing and caching. However, it isn't clear how to model the incentive aspects of P2P routing, and this is also an open question worthy of study.

4.3 Application-layer overlay networks

Many distributed systems form an application-layer network out of their constituent nodes. For instance, in many P2P file-sharing systems, each node has a set of neighbors to whom it forwards queries. There have also been many proposals for application-layer networks to perform unicast routing (e.g., [35]) and multicast routing (e.g., [36]). These overlay networks are formed by algorithms that assume nodes are obedient and ignore their own incentives. Clearly, obedience maynot be in a node's best interest. In the case of P2P file sharing, nodes would want to be close to others who share lots of files (so that their own queries could be answered quickly) but far away from others that generate lots of queries (to minimize the time theysp end processing them). In routing, a node would want to minimize the maximal distance to other nodes but would also not want to carry much traffic, and so it would prefer not to be on many link layer protocols (LCPs). This poses the question of what kinds of networks would result if users were selfish and chose their neighbors accordingly.

We expect the answer to depend on the particular system, e.g., selfishly constructed overlayfile-sharing networks will likely be different from selfishly constructed overlay routing networks.

One of the purposes of these overlaynet works is to choose routes that improve end-to-end latencyand availability. The overlaynet work can be seen as a "selfish" entity, picking LCPs for its traffic without concern for the overall network's performance. What happens if we have many different overlay networks? It may be that the

advantages of overlay networks are undermined by the result of competition among them.

4.4 Distributed Task and Resource Allocation

Instead of P2P file sharing, in which users share their storage capacity, or application-layer routing, in which users share their communication capacity, many users can participate in a CPU-intensive task to share their CPU capacity.

Distributed task and resource allocation have been fundamental research topics in distributed computer science [37, 38, 39]. Multiple agents need to work together due to an inherent distribution of resources such as knowledge, capability, information, and expertise among the agents. Agents are often unable to accomplish their own tasks alone, or they might be able to accomplish tasks better when working with others [40]. Traditionally, the designers of distributed task and resource allocation algorithms and protocols have made an implicit assumption that the participating agents will act as instructed - except, perhaps, for faulty or malicious ones [41, 42]. The main concerns of designing distributed allocation algorithms are algorithmic complexity and communication load (network complexity). The following section describes some example scenarios in real applications in the distributed task and resource allocation mechanisms.

4.4.1 Outsourcing/Virtual Teamwork

As information and communication technologies overcome the constraints of time and distance, it becomes a necessary to create virtual organizations that consists of a temporary network of independent companies linked by IT infrastructure to share skills, costs, and access to one another's markets. One of the most important advantages of a virtual organization is executing synthetic tasks by forming temporary teams composed of experts from different fields and independent organizations (which are most likely geographically dispersed) through the Internet.

One typical example is to construct temporary offshore software development teams to accomplish multiple projects [43]. The problem is how to construct the most efficient offshore teams from various outsourcing service vendors to finish these projects as soon as possible. One of the major challenges of this problem is that the actual capabilities of software engineers from different outsourcing service vendors are private information that cannot be accessed directly from outside. The project manager has to construct virtual software development teams and allocate tasks based on the reported capabilities.

Obviously, each vendor is self-interested in the sense that its goal is to maximize its own profit. Therefore, an incentive-compatible allocation mechanism is required to induce outsourcing service vendors to be willing to report the true capability of their software engineers.

Figure 2 illustrates the synthetic task allocation problem. A set of projects needs to be done. Each of these projects requires cooperation among agents from different groups with different expertise. Within each group, different agents require different amounts of time to finish identical tasks.

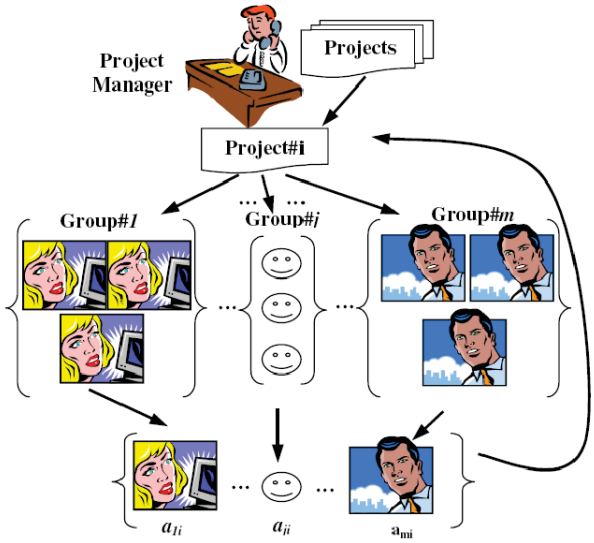


Figure 2. Synthetic Task Allocation

Indeed, the synthetic task allocation problem is essential to conducting efficient task allocation in a virtual organization. The Internet is becoming a large-scale virtual organization that holds a tremendous amount of information and resources with different owners. Little is known about how to run this organization efficiently. There does not exist a well-understood organizational structure that can model this system. In this system, allocating tasks to appropriate computational resources is analogous to allocating synthetic tasks to offshore teams. Computational tasks might need different resources from different resource owners on the Internet. Also, computational resources are most likely heterogeneous. Resources of the same type may have different capabilities (e.g. CPUs with different speeds). Incentive-compatible allocation mechanisms are required to induce resource owners to reveal the true capabilities of their computational resources.

4.4.2 Resource Sharing in Computational Grids and Peer-to-Peer Systems

Many scientific and commercial computational applications require increasingly powerful computational resources to satisfy computational performance requirements. Large amounts of computational resources connected via the Internet are idle most of the time.

Utilizing these idle resources that are owned by different organizations or individuals and are geographically dispersed to satisfy intensive computational power requirements from numerous scientific and commercial applications has become a major goal of distributed computer science. The emergent areas of grid computing and peer-to-peer computing are aimed at overcoming this challenge [44, 45, 46, 47, 48].

The recent developments in grid and peer-to-peer computing have positioned them as promising next-generation computing platforms. They enable the creation of virtual enterprises for sharing computational resources distributed across the world. Both of these two research areas in distributed computing are aimed at addressing the problem of organizing large-scale computational societies for resource sharing within virtual communities where resources may not be controlled by any single organization.

The participating agents are inherently self-interested in computational grids and peer-to-peer systems. Providing incentives for those self-interested agents to participate in a computational grid or peer-to-peer system is a key to making these computational systems feasible. A basic assumption is that agents in a computational grid or a computational peerto-peer system have peak workloads at different times so that they can utilize others' resources at idle times. That necessitates distributed load balancing [49, 50] among self-interested agents. Agents share their resources with their partners. The question is how to establish such partnerships?

In current computational grids, community standards are represented via explicit policies [51, 44]. Resources owned by various administrative organizations are shared under locally defined policies that specify what is shared, who is allowed to share, and under what conditions. Normally, a small number of sites are connected in collaborations engaged in complex scientific applications. As system scale increases, grid developers are now facing problems relating to autonomic configuration and management. How to automatically adjust system level policy to be adaptive to system updates (both hardware and software) and user requirement changes remains a major challenge in grid computing. As [51] pointed out: "Over all, scalable autonomic management remains a goal, not an accomplishment, for Grid computing."

The explicit grid policy implemented in the existing computational grids can be viewed as agreements achieved through negotiation among participating organizations.

4.4.3 Combinatorial Trade in Electronic Markets

In electronic markets, the distance between producers, wholesalers, distributors, retailers, and consumers has practically disappeared [52]. There are many more choices faced by all parties involved in electronic combinatorial trade than in a traditional trade system. The relationship between suppliers and customers is under-going revolutionary change.

Buyers vary a great deal in the quantity of goods they want to purchase, in customer service requirements, in income, in time constraints and in many other dimensions. Different purchasing goals can cause widely varying production and transaction costs. Suppliers have their own "buyer selection" strategies to achieve better profitability. Quickly differentiating the supplier's marketing strategy based on the difference of purchasing goals among various buyers plays a key role in improving the sellers' competitive capabilities in electronic markets [53, 54, 55].

In traditional markets, it is impractical for buyers to build such purchasing strategies because of the expense of access to product information. However, in the age of electronic commerce, buyers can access product information easily and inexpensively. Small buyers with little or zero bargaining power in traditional markets now can build collaborative purchasing strategies to minimize their cost in electronic markets. A well-known example of building such a purchasing strategy for buyers is to form buyer coalitions (e.g., Buyer Club) to enlarge the total quantity of goods purchased in each transaction [56]. Buyers can obtain lower prices without buying more than their real need. If the buyers are heterogeneous in the sense that they need to buy different goods in a combinatorial market, the mechanism is the so-called combinatorial coalition formation.

Buyer coalition formation is a distributed combinatorial optimization problem, which is a highly non-trivial problem that needs to be solved by considering incentive compatibility and computational tractability jointly.

5. DAMD Related Methodologies and Models

Many researchers in distributed artificial intelligence have reviewed related game-theoretic aspects in terms of incentive compatible issues in multiagent systems. This section reviews existing theoretic models that jointly address incentive compatibility and computational tractability and the methodologies, especially for distributed task and resource allocation

5.1 Algorithmic Mechanism Design

A mechanism design problem has two components: the algorithmic output specification and descriptions of what kind of benefits the participating agents can obtain. These components are given as utility functions over the set of possible outputs. Feigenbaum et al. [4] extended this model to distributed algorithmic mechanism design (DAMD), in which the same goals of incentive compatibility and computational tractability are presented. In addition, the agents, the relevant information, and the

computational model are all inherently distributed. Network complexity also needs to be considered.

He [57] studied a synthetic task allocation problem by formalizing this problem as the algorithmic mechanism design optimization problem [5]. Each synthetic task needs to be accomplished through the cooperation among agents who belong to different groups that are self-interested and have different specialties. If the capabilities are known, this problem can be solved as a makespan problem. But with selfinterested agents, our goal is to design a payment mechanism that gives agents incentive to tell the truth and form optimal teams automatically. The problem is extremely hard in the sense that there are $O(n^m)$ possible teams (n is the size of a group, and m is the number of groups.) and $k>>n$ tasks needs to be executed. Indeed, even the individual task allocation problem is NP-hard [58]. Therefore, the synthetic task allocation problem needs to jointly address incentive compatibility and computational tractability.

For self-interested agents, He [57] has developed two incentive-compatible mechanisms for this problem. The MinTeamwork is an n-approximation mechanism and a strongly truthful implementation for monotonic teamwork. By changing the valuation function and having a more restrictive assumption, the MinCompletion mechanism is a truthful implementation with 2-approximation for strongly monotonic teamwork. He has shown that incentive-compatible mechanism design is applicable for synthetic task allocation problems in virtual organizations.

5.2 Coalition Formation

Sandholm [59] probably did the most complete survey of literature related to the theory of coalition formation among self-interested agents in his dissertation. He points out that coalition formation includes three activities. The first is coalition structure generation, that is, formation of coalitions by the agents such that agents within each coalition coordinate their activities. Mathematically, it means partitioning a given set of agents into disjoint coalitions.

Game theory provides many solution concepts for evaluating the stability of a coalition under the assumption that agents involved in coalition formation have perfect rationality (i.e., algorithms can find the optimal solution with zero computational cost), which is not realistic in the real world. Sandholm and Lesser [60] extended coalition formation in game theory to a normative theory of coalitions in combinatorial domains based on a domain classification for bounded rational agents.

Game theorists did not provide actual methods of forming coalitions in real applications. Researchers in distributed artificial intelligence have put a great deal of effort into developing feasible algorithms [61] for all three activities of coalition formation among self-interested agents. Most of these works take centralized approaches by formalizing coalition formation as a set of optimization problems. Generally, a group leader is chosen for organizing the coalition formation process and is in charge of payoff division.

It is unclear how to select such an unselfish leader who is fair and acts in each member's and the group's best interests. The computational intractability of the centralized approaches also makes these algorithms only applicable for a small number of agents.

There are only a few works on coalition formation that adopt distributed approaches. Shehory and Kraus [62] developed distributed any-time algorithms of forming coalitions for cooperative agents for task allocation problems. They proposed two additional distributed algorithms for coalition formation among self-interested agents in non-super-additive games [63]. A merging process of coalition configurations from all agents is required. Voting is suggested to be one of the possible decision-making methods. Lerman and Shehory [64] developed a distributed buyer coalition formation mechanism for a large-scaled electronic market, where a buyer coalition may form when buyers encounter other buyers or existing coalitions randomly.

5.3 Negotiation among Self-interested Agents

Negotiation has been studied in many different disciplines such as politics, economics, business and public relations. Rosenschein and Zlotkin [65] pointed out that the world functions through interacting agents. Each person pursues his own goals through encounters with other people or machines. The process of negotiation takes place in both formal and informal contexts. It is part of our daily life.

The importance of studying negotiation is straightforward for designing decentralized mechanisms for distributed task and resource allocation problems [66]. Negotiations are initialized when agents need to make agreements on how to allocate a shared resource, how to do distributed load balancing, how to exchange resources etc.

There are two main research issues on negotiation in multiagent systems. The first is how to develop practical negotiation strategies [67], i.e., what kind of strategies that agents should use to maximize their own good during a negotiation process. The second is to develop protocols that allow automated negotiation agents to negotiate with each other. Game theory tools (e.g., Nash equilibrium, dominant strategy etc.) are used to evaluate negotiation strategies and protocols.

Negotiation has been studied in distributed artificial intelligence both in distributed problem solving (DPS) where agents are cooperative and in Multiagent Systems (MAS) where agents are self-interested. Negotiation is

used in DPS for solving conflicts, distributed planning and distributed search.

Sycara [68] developed a model of negotiation that combines case-based reasoning and optimization of multi-attribute utilities. Zeng and Sycara embedded learning into negotiation. Agents can learn from previous encounters about their opponents' negotiation strategies so that they choose corresponding strategies to influence their opponents or to obtain a better deal. Sierra et al. [66] presented a model of negotiation for autonomous agents, which is distilled from intuitions about good behavioral practice in human negotiation. Sandholm and Lesser [69] explored issues such as levels of commitment that arise in automated contract among self-interested agents whose rationality is bounded by computational complexity.

5.4 Distributed Task and Resource Allocation in Multiagent Systems

Distributed task and resource allocation is a central theme of distributed computer science. Here we interested in cooperative task allocation and resource sharing problems in systems that are established through interactions among multiple self-interested agents that are developed by different designers and belong to different owners. This type of allocation problems has a variety of applications in grid and peer-to-peer computing, electronic commerce and virtual organizations.

5.4.1 Contract Net and Levels of Commitment

The most influential distributed task allocation mechanism in distributed artificial intelligence is the Contract Net protocol [70], which can be used for both cooperative agents and self-interested agents. The basic idea is that a task manager auctions a group of tasks, agents bid on these tasks based on their local marginal cost calculations. The original Contract Net does not take computational tractability into consideration, even though the marginal cost calculation for combinatorial problems are most likely intractable. Sandholm and Lesser [69] extended the Contract Net protocol to allow it to work among self-interested computationally limited agents. Agents can reallocate tasks to each other for dynamically constructed charges. As a result, a more profitable global task allocation is reached than the initial one, while not executing a centralized task allocation algorithm.

5.4.2 Methods for Task Allocation via Agent Coalition Formation

As mentioned before, Shahory and Kraus [39] developed distributed anytime algorithms for forming coalitions among cooperative agents for task allocation problems. They considered situations where it is necessary to execute a task by a group of agents because it is more efficient or a single agent is not able to perform the task. The objective of this work is to improve the efficiency by allocating tasks to cooperative agent coalitions, which are formed through distributed algorithms. There is not an explicit negotiation protocol among agents. Each agent calculates the costs of coalitions it involves by itself and joins the coalition with the lowest cost for a certain task. The procedure is executed iteratively until there are no more tasks or no existing coalition is beneficial.

5.4.3 Auction and Market Based Resource Allocation Mechanisms

Auctions and markets represent two ends of a spectrum of market formulations [71]. On the market end, an attempt is made to satisfy all bidders and sellers at a given price. At the auction end, one bidder and seller is satisfied at a given price.

Market based resource allocation mechanisms are decentralized and no direct communication is needed. The balance between supply and demand decides the actual resource allocation. When the supply and demand for a certain resource reaches equilibrium, the price becomes stable. How long it will take to reach equilibrium is normally unpredictable. Hence, price setting is a big obstacle for developing a market based resource allocation mechanism [71].

An auction [72] is the simplest resource allocation mechanism for selfinterested agents in terms of its implementation. Auction-based distributed resource allocation mechanisms have been successful in many real distributed allocation applications. The most positive result about auction is that the second sealed price auction belongs to VCG family and is a truthful implementation. Also, an auction has no problem with price setting.

These two types of allocation mechanisms are basically monetary approaches [45]. When computational resource allocation among self-interested agents is considered, agents do not explicitly buy others' resources but use them when they are idle. It is very hard for agents to decide whether they should buy the time slots of using a resource or buy the resource itself.

6. Open DAMD Problems

The central mission of theoretical computer science (TCS) is to determine which problems are easy and which are hard in relevant computational models. In the Turing-machine model of centralized computation, the (crude) distinction is between polynomialtime solvable problems and those that are NP-hard. One of the major goals of this study of DAMD foundations is to develop the tools needed to classify relevant problems as easy or hard "to compute incentive-compatiblyon the Internet"

and to find more natural examples of both hard and easy DAMD problems.

Informally, a DAMD problem can be considered "easy" if it can be solved in a manner that is both incentive-compatible and computationallytractable. The technical definitions of incentive compatibility and computational tractability will depend on the particular problem under consideration.

The welfare-maximizing multicast cost sharing is easy when strategyproofness is the incentive-compatibility requirement, and low absolute network complexity is the computational-tractability requirement. The first open problem is to determine how general this result is. The marginal-cost mechanism is the only strategy proof and efficient mechanism that satisfies Non Positive Transfer (NPT) and Voluntary Participation (VP). If we remove the NPT and VP requirements, then we have the entire family of VCG mechanisms at our disposal. How many of these have reasonable network complexity?

Open Problem 1. Fully characterize the set of easy welfare-maximizing multicast cost sharing problems.

Of course, we are interested in far more than just multicast cost sharing, and one of the central DAMD challenges is the search for additional examples.

Open Problem 2. Design good distributed algorithmic mechanisms to show that natural problems of interest are easy.

While easy problems are a field's "successes," hard problems often lead to a deeper understanding of an approach's fundamental limitations. Thus, we are interested in how to define hardness in the DAMD context. Superficially, a problem is hard if it cannot be solved in a manner that satisfies both the incentive-compatibility and the computational tractability requirements. There will be many problems for which this cannot be done; NP-hard problems, for example, cannot be solved in a computationally tractable manner (unless P=NP), and there are no efficient, strategyproof, and budget-balanced solutions to general cost-sharing problems.

Canonical hard problems will help us understand the fundamental nature of hardness in DAMD, as opposed to hardness that results solely from computational issues or solely from incentive issues.

Open Problem 3. Define the computational models and computational resources needed to formalize "network complexity," both absolute and relative, and other relevant measures of DAMD complexity. Develop the appropriate notions of "reduction" to show that certain problems are hard or complete for the relevant complexity classes.

Open Problem 4. Thoroughly investigate the interplay between strategic models and computational models in DAMD. In particular, develop realistic strategic models for a variety of DAMD problems, including problems in which one or more of the players are adversarial or faulty. If possible, develop general techniques for converting distributed algorithmic mechanisms in which some of the parties must be assumed to be obedient into ones in which all parties are realistically strategically modeled..

7. Summary

This survey paper reviews the mechanism design (esp. DAMD) in distributed systems and gives some possible Internet applications with DAMD. The Internet's decentralized nature, in which distributed computation and autonomous agents prevail, makes DAMD a very natural approach for many Internet problems.

References

[1] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and DanWerthimer. Seti@home: An experiment in public-resource computing. Communications of the ACM, 45(11):56-61, November 2002.

[2] Arjen Lenstra, Mark S. Manasse. Factoring by electronic mail. In Advances in Cryptology, Eurocrypt '89, LNCS volume 434, pages 355-371. Springer-Verlag, Berlin, 1990.

[3] Noam Nisan, Amir Ronen. Algorithmic mechanism design. Games and Economic Behavior, 35:166-196, 2001.

[4] Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. Journal of Computer and System Sciences, 63:21-41, 2001.

[5] N. Nisan, Algorithms for Selfish Agents: Mechanism Design for Distributed Computation, In Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, volume 1563, Springer, Berlin, pages 1-17, 1999.

[6] N. Nisan, Rationalityas a Paradigm for Internet Computing, Invited Talk at the U.C. Berkeley Workshop on Theory of Computation & the Sciences, http://www.cs.huji.il/~noam/rationality.ppt

[7] C. Papadimitriou, Algorithms, Games, and the Internet, in Proceedings of the 33rd Symposiumon Theory of Computing, ACM Press, New York, pages. 749-753, 2001.

[8] E. Clarke, Multipart pricing of public goods, Public Choice 11, pages. 17-33, 1971

[9] T. Groves, Incentives in teams, Econometrica 41, pages. 617-663, 1973

[10] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, Journal of Finance 16, pages 8-37, 1961

[11] B. Bernheim, Rationalizable Strategic Behavior, Econometrica 52, pages 1007-1028, 1984

[12] D. Pearce, Rationalizable Strategic Behavior an d the Problem of Perfection, Econometrica 52, pages 1029-1050, 1984
[13] J. Smith, Evolution and the Theory of Games, Cambridge UniversityPress, Cambridge, 1982.
[14] Aaron Archer and Eva Tardos. Frugal path mechanisms. In Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02), pages 991-999. ACM Press/SIAM, New York, January 2002.
[15] Amos Fiat, Andrew V. Goldberg, Jason D. Hart line, and Anna R. Karlin. Competitive generaliz edauctions. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02), pages 72-81. ACM Press, New York, May 2002.
[16] John Hershberger and Subhash Suri. Vickrey prices and shortest paths: What is an edge worth? In Proceedings of the 42nd annual ACM Symposium on the Foundations of Computer Science (FOCS '01), pages 129-140. ACM Press, New York, 2001.
[17] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00), pages 242-252. ACM Press, New York, 2000.
[18] Tim Roughgarden and Eva Tardos. How bad is selfish routing? Journal of the ACM, 49:236-259, 2002.
[19] Dov Monderer and Moshe Tennenholtz. Distributed games: From mechanisms to protocols. In Proceedings of the 6th National Conference on Artificial Intelligence and of the 11th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI '00), pages 32-37. AAAI/MIT Press, Menlo Park, CA, July 1999.
[20] David C. Parkes. iBundle: An efficient ascending price bundle auction. In Proceedings of the ACM Conference on Electronic Commerce (EC '99), pages 148-157. ACM Press, New York, November 1999.
[21] David C. Parkes and Lyle H. Ungar. Iterative combinatorial auctions: Theory and practice. In Proceedings of the 7th Conference on Artificial Intelligence and of the 12th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI '00), pages 74-81. AAAI/MIT Press, Menlo Park, CA, July 2000.
[22] Tuomas Sandholm. Distributed rational decision making. In G. Weiss, editor, Multiagent systems: A Modern Introduction to Distributed Artificial Intelligence, pages 201-258. MIT Press, Cambridge, MA, 1999.
[23] MichaelWellman. A market-oriented programming environment and its applications to distributed multicommodity flow problems. Journal of AI Research, 1:1-23, 1993.
[24] M. Wellman, W. Walsh, P. Wurman, and J. Mackie-Mason. Auctions for decentralized scheduling. Games and Economic Behavior, 35:271-303, 2001.
[25] Shai Herzog, Scott Shenker, and Deborah Estrin. Sharing the "cost" of multicast trees: an axiomatic analysis. IEEE/ACM Transactions on Networking, 5(6):847-860, December 1997.
[26] Herve Moulin and Scott Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. Economic Theory, 18:511-533, 2001.
[27] Micah Adler and Dan Rubenstein. Pricing multicasting in more practical network models. In Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Mathematics (SODA '02), pages 981-990. ACM Press/SIAM, New York, January 2002.
[28] John Mitchell and Vanessa Teague, Private communication, 2002.
[29] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, A BGP-based Mechanism for Lowest-Cost Routing, in Proceedings of the 21st Symposium on Principles of Distributed Computing, ACM Press, New York, pages 173-182, 2002.
[30] T. Griffin and G. Wilfong, An analysis of BGP convergence properties, in Proceedings of SIGCOMM' 99, ACM Press, New York, pages 277-288, 1999.
[31] T. Kelly, Y-M. Chan, S. Jamin, and J. MacKie-Mason, Biased Replacement Policies for Web Caches: Differential Quality-of-Service and Aggregate User Value, in Proceedings of the 4th International Web Caching Workshop, pages 1-10, 1999.
[32] E. Adar and B. Huberman, Free Riding on Gnutella, in First Monday, http://www.firstmonday.dk/issues/issue510/adar/index.html, October 2000.
[33] J. McCoy, http://www.mojonation.net/
[34] S. Saroiu, P. Gummadi, and S. Gribble, A Measurement Studyof Peer-to-Peer File Sharing Systems, in Proceedings of Multimedia Computing and Networking, SPIE Press, Bellingham, pages 156-170, 2002.
[35] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, Resilient OverlayNet works, in Proceedings of the 18th Symposium on Operating System Principles, ACM Press, New York, pages 131-145, 2001.
[36] Y. Chu, S. Rao, and H. Zhang, A Case for End System Multicast, in Proceedings of ACM Sigmetrics' 00, ACM Press, New York, pages 1-12, 2000.
[37] Zweben, M. and M.S. Fox, Intelligent Scheduling. San Francisco, CA: Morgan Kaufmann Publishers, 1994.

[38] Clearwater, S.H, Market-Based Control: A Paradigm for Distributed Resource Allocation. Singapore: World Scientific, 1996.
[39] Kraus, S. and T. Plotkin. Algorithms of Distributed Task Allocation for Cooperative Agents. Theoretical Computer Science 242(1-2): 1-27, 2000.
[40] Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA: The MIT Press, 1999.
[41] Nisan, N. and A. Ronen, Algorithmic Mechanism Design. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing. Pages 129-140, Atlanta, GA, May 1-4, 1999.
[42] Ronen, A., Solving Optimization Problems among Selfish Agents. Ph.D Dissertation, Hebrew University, Jerusalem, 2000.
[43] Hatch P.J., Offshore 2005 Research Preliminary Findings and Conclusions. Vers.1.2.5. Ventoro Report. Portland, OR, Ventoro Corporation, 2005.
[44] Foster, I., N.R. Jennings, and C. Kesselman. Brain Meets Brawn: Why Grid and Agents Need Each Other. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems. Pages 8-15, New York, July 21-23, 2004.
[45] Buyya, R., Economic-Based Distributed Resource Management and Scheduling for Grid Computing. Ph.D. Dissertation, Monash University, Melbourne, Australia. 2002.
[46] Buyya, R., D. Abramson, J. Giddy, and H. Stockinger. Economics Paradigm for Resource Management and Scheduling in Grid Computing. Concurrency and Computation: Practice and Experience. 14: Grid Computing Environments Special Issue 13-14. 2002.
[47] Milojicic, D.S., V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technique Report, HPL-2002-57, HP Laboratories, Palo Alto, CA. 2002.
[48] Berman, F., G. Fox, and T. Hey. Grid Computing: Making the Global Infrastructure a Reality. Chichester, England: John Wiley & Sons. 2003.
[49] Lan, Z., V. Taylor and G. Bryan. A Novel Dynamic Load Balancing Scheme for Parallel Systems. Journal of Parallel and Distributed Computing 62(12):1763-1781. 2002.
[50] Shan, H., L. Oliker, and R. Biswas. Job Superscheduler Architecture and Performance in Computational Grid Environment. In Proceedings of Super Computing'03. Pages 44, Phoenix, AZ, November 15-21. 2003.
[51] Foster, I. and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In Proceedings of the Second International Workshop on Peer-to-Peer Systems. Pages 118-128, Berkeley, CA, February 20-21. 2003.
[52] Ye, Y. and Y. Tu. Dynamics of Coalition Formation in Combinatorial Trading. In Proceedings of International Joint Conference of Artificial Intelligence. Pages 625-632, Acapulco, Mexico, August 9-15. 2003.
[53] He, L. and T.R. Ioerger. An Efficient Heuristic Bundle Search Algorithm for Buyers in Electronic Markets. In Proceedings of the 2004 International Conference on Artificial Intelligence. Pages 729-735, Las Vegas, NV, June 23-26. 2004.
[54] He, L. and T.R. Ioerger. Combining Bundle Search with Buyer Coalition Formation in Electronic Markets. Electronic Commerce Research and Applications Journal 4(4): 329-344. 2005.
[55] He, L. and T.R. Ioerger. Forming Resource-Sharing Coalitions: A Distributed Resource Allocation Mechanism for Self-Interested Agents in Computational Grids. In Proceedings of the Twentieth Annual ACM Symposium on Applied Computing. Pages 84-91, Santa Fe, NM, March 13 - 17. 2005
[56] Li, C. and K. Sycara. Algorithm for Combinatorial Coalition Formation and Payoff Division in an Electronic Marketplace. In Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems. Pages 120-127, Bologna, Italy, July 15-19. 2002.
[57] Linli, He, Mechanism Design for Distributed Task and Resource Allocation among Self-interested Agents in Virtual Organization, PhD Dissertation, Texas A&M University, 2006.
[58] Coffman, E.G., L. Flatto, M.R. Garey and R.R. Weber. Minimizing Expected Makespans on Uniform Processor Systems. Advances in Applied Probability 19:177-201. 1987.
[59] Sandholm, T., Negotiation among Self-Interested Computationally Limited Agents. Ph.D. Dissertation, University of Massachusetts at Amherst. 1996.
[60] Sandholm, T. and V. Lesser. Coalitions among Computationally Bounded Agents. Special issue on Economic Principles of Multiagent Systems, Artificial Intelligence 94(1): 99-137. 1997.
[61] Caillou, P., S. Aknine, and S. Pinson. A Multi-Agent Method for Forming and Dynamic Restructuring of Pareto Optimal Coalitions. In Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems. Pages 1074-1081, Bologna, Italy, July 15-19. 2002.
[62] Shehory, O. and S. Kraus. Methods for Task Allocation via Agent Coalition Formation, Artificial Intelligence Journal 101 (1-2): 165-200. 1998.
[63] Shehory, O. and S. Kraus. Feasible Formation of Coalitions among Autonomous Agents in Non-Super-Additive Environments. Computational

Intelligence 15(3): 218-251. 1999.
[64] Lerman, K. and O. Shehory. Coalition Formation for Large Scale Electronic Markets. In Proceedings of the International Conference on Multi-Aent Systems. Pages 167-174, Boston, MA, July 10-12. 2000.
[65] Rosenschein, J. S. and G. Zlotkin. Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. Cambridge, MA: The MIT Press. 1994.
[66] Jennings, N., P. Faratin, A. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, Methods and Challenges. International Journal of Group Decision Negotiation 10(2): 199-215. 2001.
[67] Kraus, S., Strategic Negotiation in Multiagent Environments. San Francisco, CA: The MIT Press. 2001.
[68] Sycara, K.P., Persuasive Argumentation in Negotiation. Theory and Decision 28: 203-242. 1990.
[69] Sandholm, T. and V. Lesser. Leveled Commitment Contracting: A Backtracking Instrument for Multiagent Systems. AI Magazine 23 (3): 89-100. 2002.
[70] Davis, R. and R.G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. Artificial Intelligence 20 (1): 63-109. 1983
[71] Wolski, R., J. Brevik, J. Plank, and T. Bryan. Grid Resource Allocation and Control Using Computational Economies. In: Grid Computing: Making the Global Infrastructure a Reality, eds. F. Berman, G. Fox, and T. Hey, 747-772, Chichester, England: John Wiley and Sons. 2003.
[72] Krishna, V., Auction Theory. London, UK: Academic Press. 2002.

Collaborative Semantic Indexing of Multimedia Data Objects

Wing Sze CHAN

Abstract

While creation of multimedia data objects becomes easy and massive, the searching activity of those objects becomes an important issue. However, this kind of activity is far more challenging than the searching of text-based documents. We propose an approach that enables the discovery of multimedia objects through dynamic collective indexing and explicit/implicit user feedback analysis. Our method allows semantic concepts to be discovered and convergent to the index hierarchy by analyzing users' interactions with the search system. By exploiting the relevance feedback from user, the overall user satisfaction would be maximized. Through the growth and evolution of the index hierarchy, the semantic index can be dynamically constructed, validated, and built-up with respect to the user preference. Index convergence behaviour and modelling are also discussed.

1 Introduction

A huge amount of multimedia data objects, in various forms and formats, exist and grow explosively in our daily life. It is estimated that by 2010, over a billion digital images will be created each day [4]. The multimedia data object retrieval problem becomes important and necessary. Multimedia information search is far more difficult than searching text-based documents since the content of text-based documents can be extracted automatically while the content of multimedia objects cannot be automatically determined [9, 10].

Research in image retrieval has been divided into two main categories: "concept-based" image retrieval, and "content-based" image retrieval [1–3, 5, 8, 11, 12, 15, 17]. The former focuses on higher-level human perception using words to retrieve images (e.g. title, keywords, captions), while the latter focuses on the visual features of the image (e.g. size, colour, texture). In an effective "concept-based" multimedia retrieval system, efficient and meaningful indexing is necessary [6,7]. Due to current technological limitations, it is impossible to extract the semantic content of multimedia data objects automatically [14, 19]. Meanwhile, the discovery and insertion of new indexing terms are always costly and time-consuming. Therefore, novel indexing mechanisms are required to support their search and retrieval.

Community users are usually critical of a multimedia searching system. The relevance feedback which collected from users' interactions through a system can be used to adapt and evolve the system behaviour to fit their needs. Different communities of users may have different expected relevance to a search term, depending on time, geographic and cultural interests. The relevance feedback which provided by the user are essential. However, it is hard to convince users to rate the numerous multimedia data objects. Furthermore, the concern of user privacy issues is rising [13, 18], and therefore we should explore other means for discovering multimedia data objects. The history of user behaviour and the time spent on the query sessions may reflect user feedback implicitly, while the analysis of query sessions is beyond the scope of this study. By analyzing both of the explicit and implicit relevance feedback from user, the search system can be tuned based on user preferences.

Our proposed approach can be seen as a form of machine learning, while it is dissimilar from a typical supervised learning approach, since there is no separate training phase. By the continuous interactive processes between user and the system, user knowledge and judgement of the search terms to the data objects will evolve over the time. Thus, we will also study the index convergence behaviour and modelling.

2 The Indexing Approach & Hierarchical Evolution

Our approach concentrates on the indexing of semantic contents of multimedia objects and will exclude metadata from consideration, since indexing by metadata is relatively straightforward and less meaningful than semantic contents as perceived by humans [10]. This indexing approach enables user to search multimedia objects conceptually by the semantic visual features.

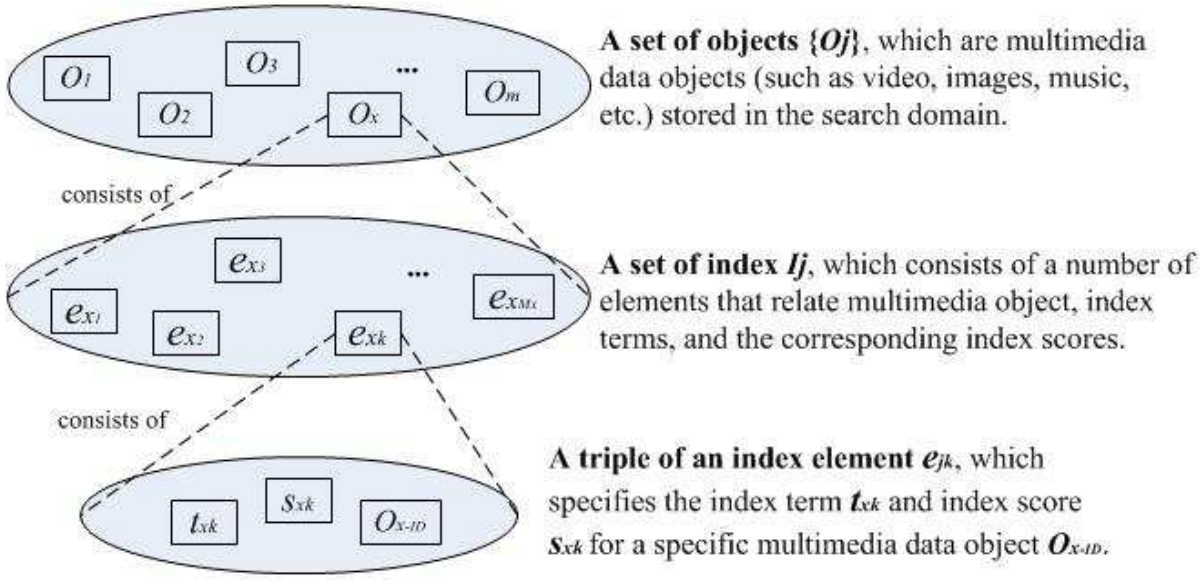


Figure 1. Composition of Multimedia Data Objects and Index Elements

2.1 Index Structure & Hierarchy

We consider a set of multimedia data objects $\{O_j\}$, such as images, video, or music, where their semantic characteristics and contents cannot be extracted automatically. Each O_j has an index set I_j, which consists of a number of elements $\{e_{j1}, e_{j2}, ..., e_{jM_j}\}$. Each index element e is a triple, which is composed of an index term ID t_{jk}, a corresponding index score s_{jk}, and an object ID O_{j-ID}. The index scores reflect the significance of an index term to the object; the higher the index score, the more important is the index term to the object. In other words, the lower the index score means the less important is the index term to the object. Fig. 1 shows a clear view of the decomposition structure of the multimedia data objects and index elements.

The index hierarchy refers to the collective index sets I of all the objects O_j in the database [10]. Fig. 2 shows that the index set I is partitioned into N levels L_1, L_2, ..., L_N by partitioning the score value s_{jk} with a set of parameters P_1, P_2, ..., P_N. For a given index term with score x, the index term will be placed in level L_i if $P_i \leq x < P_{i+1}$, where $i = 1, ..., N-1$. Otherwise, it would be placed in level N if $P_N \leq x$. In this index hierarchy, the higher the level, the more important it is.

2.2 Minimal Indexing & Index Growth

In order to be discovered by users, each multimedia object should be minimally indexed initially. When an object O_j is minimally indexed, it means that O_j has only a single index term T where T consists of a single word only. Through successive usage of the system, the index set of an object would be grow. Such that, an object which is minimally indexed with term $T = T_1$ may become indexed with multiple terms $T = T_1, T_2, ..., T_n$. Consider an object O_J, which is minimally indexed with an index term T_1. Consider a user enters an input query $Q(T_1, T_2)$, the system would return an answer vector V_{ans} which consists a set of objects that is indexed with T_1 or T_2. When the user select O_J in V_{ans}, T_2 would be added to O_J at the low level of the index hierarchy. If many queries that contain T_2 also select on O_J continuously, the index score of T_2 of O_J would be increase and promote to the high level of the index hierarchy. Consequently, the T_2 of O_J would be properly indexed. Meanwhile, the T_1 of O_J may drop to the lower level of the index hierarchy since it would be affected by the user relevance feedback.

2.3 Index Score Update Influenced by Relevance Feedback

The index scores are directly affected by the user relevance feedback, either positive or negative. By the continuous use of the system, our system can collect and analyze both explicit and implicit relevance feedback from users. When the system receives a positive feedback from user, the index score(s) that relate to the search terms of the query would be increased. Similarly, the index score(s) would be decreased when the a negative feedback is received. Those positive and negative feedbacks can be the

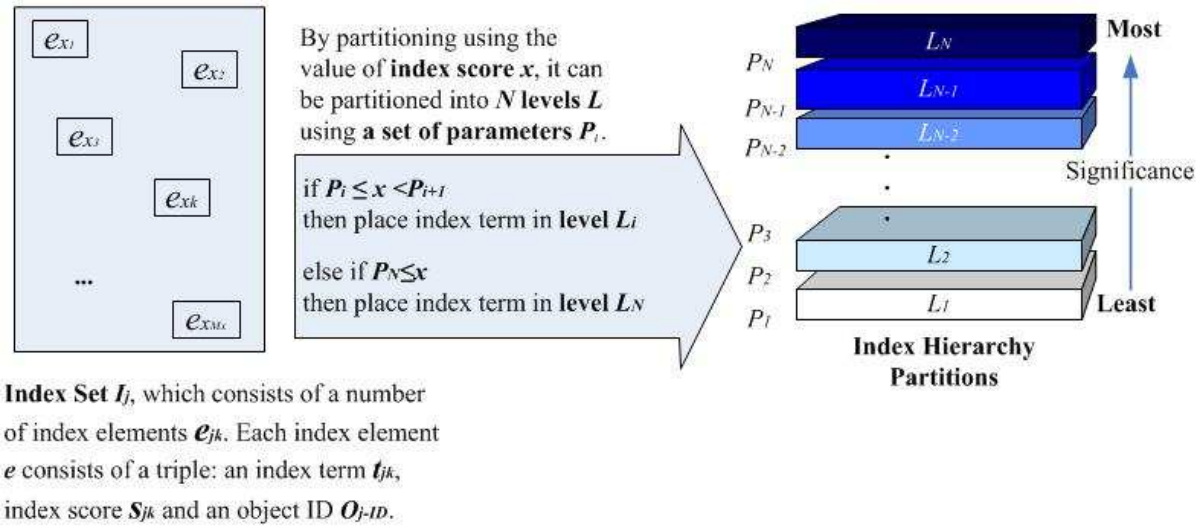


Figure 2. Index Hierarchy

relevance feedback that collected directly or indirectly from users. It would be discussed in Section 3.

Considering an example of a user input search query $Q(T_1, T_2)$ that consists of two search terms T_1 and T_2, suppose there are k multimedia objects $O_1, O_2, ..., O_k$ are returned in the answer vector V_{ans} disregarding the object rankings. When user provides a positive feedback or selects the desired object O_x in the answer vector, the index scores of T_1 and T_2 of O_x would be increased by a predefined value Δ_+. In opposite, when user provide a negative feedback on O_x, the index scores of T_1 and T_2 of O_x would be decreased by a predefined value Δ_-. Moreover, when a user do not select any object in the answer vector, the index scores of T_1 and T_2 for all objects in the answer vector $(O_1, O_2, ..., O_k)$ would also be decreased.

2.4 Object Ranking

In the information explosion era, search rankings has become significant [20]. Usually, the top-ranked objects should be more relevant to the search query. Every submitted query which consists of one or more search terms, is expected to return an answer vector $V_{ans} = [O_1, O_2, ...O_k]$, where k is the number of objects returned that is relevant to the search terms. Our object ranking approach relies on the index scores, since this score implies the relevance of an index term to an object. The higher the score, the more the index term is relevant to the object. There are two approaches for ranking the multimedia data objects returned in the query result lists.

2.4.1 Naïve Strategy

This strategy is to return the best k result objects ordered by their index scores, related to the search term(s) of the object, in descending order. By this strategy, the top ranked object O_1 in the answer vector V_{ans} is the most relevant to the search query, while O_k is the least. It is a typical strategy for building hot links, such as "top ten list of most clicked links" which can be found in many portals homepage. It implies that the probability that an object O_j being clicked would be directly proportional to the rank of object O_j. Since the "top ten" are more likely to be seen, therefore, those top ranked objects are more likely to be selected. Thus, the index scores of those objects have higher chance to be promoted. Such that there would be an initial bias, in the top tens links, which can easily lead to a local maxima problem. Consequently, some significant objects would be hard to show up or ranked very low in the result lists. In the worst case, some objects may never been shown to users. Such that, those objects may be nearly "hidden" or never receive positive feedback from users.

2.4.2 Randomized Strategy with Genetic Algorithms

This strategy provides variations in query results by using Genetic Algorithms (GA), and thus it is designed for overcoming the local maxima problem. It returns k result objects in the answer vector V_{ans} by random extractions from the index. The random extraction process involves a series of random selection among the object set O_j. It performs again and again until k distinct objects are selected to be shown. By the randomness of the GA, the best rated objects would have proportionally higher chances to appear in the

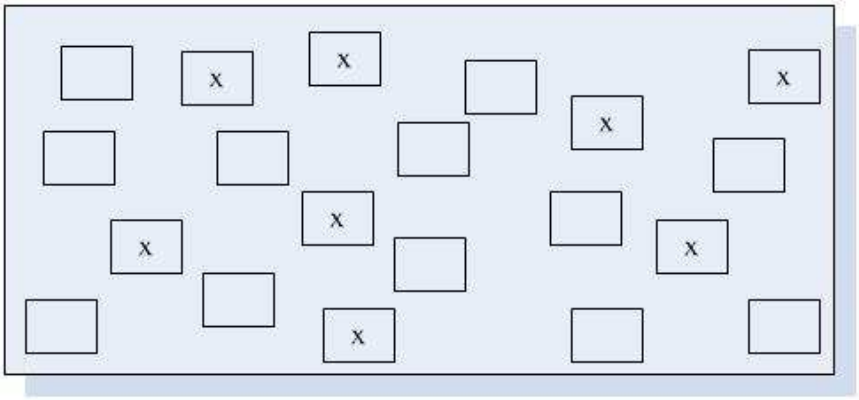


Figure 3. Potential Indexable Terms

answer vector; meanwhile, those "hidden" objects would have a non-zero probability of being promoted to appear in the answer vector. Therefore, those "hidden" objects would have a chance to be ranked in a higher ranking position and discovered eventually.

Although the problem of local maxima can be solved by our randomized strategy, it also introduces some noise which tend to lower the overall system performance. The system performance quality is degraded since the answer vector would consist of both good relevant objects and some irrelevant objects. However, those irrelevant objects are essential in the discovery of the "hidden" objects. Therefore, we adopt elitism, a technique from GA, to reduce the noise that induced by the randomized strategy. By adjusting the elite E (number of best objects in an answer vector), where $E \in [0, k]$, it guarantees the quality of the answer returned. There are two extreme cases when considering the value of E; $E = 0$ means there is no any elitism such that all objects in the answer vector are generated by the randomized strategy. On the other way around, $E = k$ means there is no noise object in the answer vector, such that it is equivalent to the mentioned naïve strategy.

3 User Relevance Feedback

Relevance feedback (RF) is a classical information retrieval (IR) technique where users relay their agreement with the system's evaluation of document relevance back to the system, which then uses this information to provide a revised list of search results [16]. It allows user to mark relevant (positive feedback) or irrelevant (negative feedback) to the object(s) of the result list by their relevance judgments. The user relevance feedback collected would be useful for refining the index scores, such that it helps tuning the index hierarchy to fit user preferences.

Our model collects both explicit and implicit relevance feedback from the user community. The explicit feedback refers to the relevance, indicating the relevance of the object retrieved for a query, is collected directly from the user judgements. Our model enables user to indicate relevance explicitly using a binary relevance system. Binary relevance feedback indicates that a multimedia data object is either relevant or irrelevant for a specific query. Once a user submit a query, our system will return a list of query results to the user. In order to maintain the spirit of web 2.0, collaborative users involvement, our system allows users to provide their relevance feedback for the multimedia objects of the query results. Their feedback can be either positive or negative.

Although the idea of exploiting user's feedback to rate relevance seems promising, it is not easy to convince a community of users to spend their time to explicitly rate objects. Therefore, our model also collect implicit relevance feedback from them. The implicit feedback is inferred from user behaviour and their history, such as noting which object(s) that user do and do not select for viewing, the duration of time spent on viewing an object. All these information, can be collected automatically, would reflect user's satisfaction and expectation of the query result. When user click on an object in the answer vector, we can infer that the selected object may relevant to the user query. Our system will treat it as a kind of positive feedback from the user implicitly. On the contrary, when user do not select any object in the answer vector, we can infer that user may think that the objects in the answer vector are irrelevant to their input query or they are not interested in those objects. Our system will treat it as a kind of negative feedback from the user implicitly.

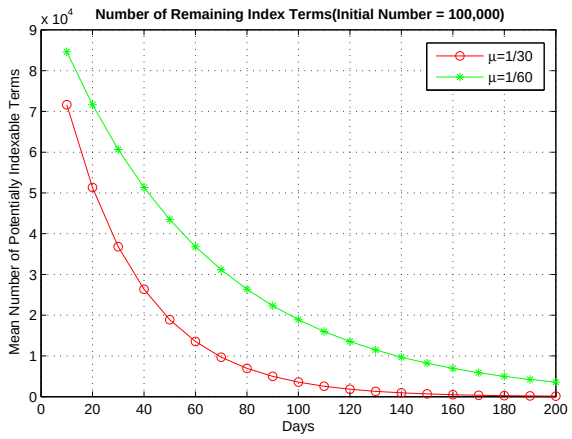


Figure 4. Number of Remaining Index Terms

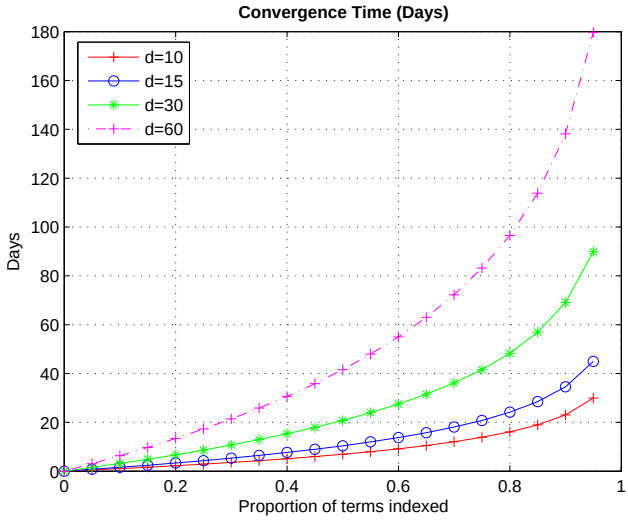


Figure 5. Convergence Time

4 Modelling Index Convergence Behaviour

Since the measurement of the relevance of an index term to an object is based on the related index score, the index scores of the system are expected to evolve to an ideal situation. We assume that each index score for an index term of a specific object would have a hidden ideal score value S_H. When the actual index score reaches S_H, this index can be considered as having convergent. By the continuous usage of the system, the indexes would be convergent.

4.1 Index Convergence Behaviour

In theory, indexes would evolve to an ideal status by the index continuous convergence processes. Consider there are J objects in the search space, each of the objects are indexed with m initial index terms, and M be the number of the maximal index terms.

Let N_t be the state of the system which signifies the number of terms remaining to be indexed. N_t is a random variable that changes over time. Let the process starts at $t = 0$. Thus initially, we have

$$N_0 = J(M - m). \tag{1}$$

As time goes on, N_t will gradually decrease. N_t will decrement by 1 whenever a potential indexable term is being indexed. We assume that the random indexing pattern for a given term follows a Poisson process with indexing rate μ, where in a small time interval Δh, a potential indexable term has a probability of $\approx \mu\Delta t$ of being actually indexed. The rate is dependent on the usage and indexing frequency of objects in the collection. Thus, over time, each potential indexable term is gradually being deleted as they are become indexed. Fig. 3 shows this situation where the empty boxes signify the potential indexable terms, while the ones with "X" signify the ones having been actually indexed in the course of this evolutionary process.

From the property of the Poisson distribution, the probability that a potential indexable term remaining unindexed at time t is $e^{-\mu t}$. Therefore, we obtain the following binomial distribution for N_t

$$Prob[N_t = k] = \binom{N_0}{k} (1 - e^{-\mu t})^{N_0 - k} e^{-\mu t k}, \tag{2}$$

which gives

$$E(N_t) = N_0 e^{-\mu t}, \tag{3}$$

$$Var(N_t) = N_0(1 - e^{-\mu t})e^{-\mu t}, \tag{4}$$

Adopting a time unit of days, $\frac{1}{\mu}$ can be taken as the average time elapsed to install the index term. For example, if $\mu = 0.1$, this means that the average time to install the index term is 10 days. Fig. 4 plots the number of remaining index terms over time for $N_0 = 100,000$, and $\mu = \frac{1}{30}, \frac{1}{60}$. We see that the number of indexable terms drops quickly at first, then do so slowly as time goes on. As $t \to \infty$, we see from equation 3 that the collection tends to be fully indexed with $E(N_t) \to 0$, irrespective of the initial number of potential indexable terms. Also, from equation 4, $Var(N_t) \to 0$ as $t \to \infty$, which indicates that the effect of stochastic fluctuation would be small; this implies that, over a long period of time, the process may be viewed as a deterministic one.

From equation 3, we can determine the time T_p, on average, when a certain proportion of p of the potential index-

able terms have been indexed; i.e. letting

$$p = \frac{(N_0 - N_0 e^{-\mu T_p})}{N_0}, \quad (5)$$

we obtain

$$T_p = \frac{1}{\mu} \ln(\frac{1}{1-p}). \quad (6)$$

Replacing μ by $d = \frac{1}{\mu}$ in the above gives

$$T_p = d \ln(\frac{1}{1-p}). \quad (7)$$

Fig. 5 shows the convergence behaviour for $d = 10, 15, 30, 60$. In this model, each potential index term behaves independently of other index terms. When there are many terms remaining to be indexed, the collective indexing rate tends to be high, and this collective rate will decline as fewer and fewer terms are available to be indexed; this is evident from Figure 3, where the curves rise much more steeply as $p \rightarrow 1$.We observe that in order to complete the indexing of 95% of the terms, it takes approximately three times the amount of time for indexing an individual term. Indeed taking $p = 0.95$, we have $\ln(\frac{1}{1-p}) = 2.99$.

5 Experiments on Index Convergence

We performed experiments to examine the possibility, systematic and repeatable of our methodology. In order to evaluate the effectiveness of our large scale indexing approach, we adopted a simulation approach based on the hidden ideal score values S_H. The goal of the series of experiments is to investigate the relationship between the initial S_H and the convergence of the index.

In the scoring system, 0 denotes the minimum score and 1 denotes the maximum score. Initially, Each index term of an object is associated with an static hidden score value S_H where $0 < S_H < 1$. The simulation model simulates the following processes: user queries submission, answer vector computation and user relevance feedback simulation on the basis of S_H. In the following series of tests, we performed 2,000 of objects in 5,000 queries of answer vector size 10 with different initial random S_H values. For the measurement of the system evaluation, we introduce relative answer relevance R to measure the single query optimality. R is the ratio between the total hidden score of a query answer and the best possible answer of length k; the aim of this measure is to express how optimal is the current query answer with respect to the objects currently in the database.

5.1 Experimental Results & Discussions

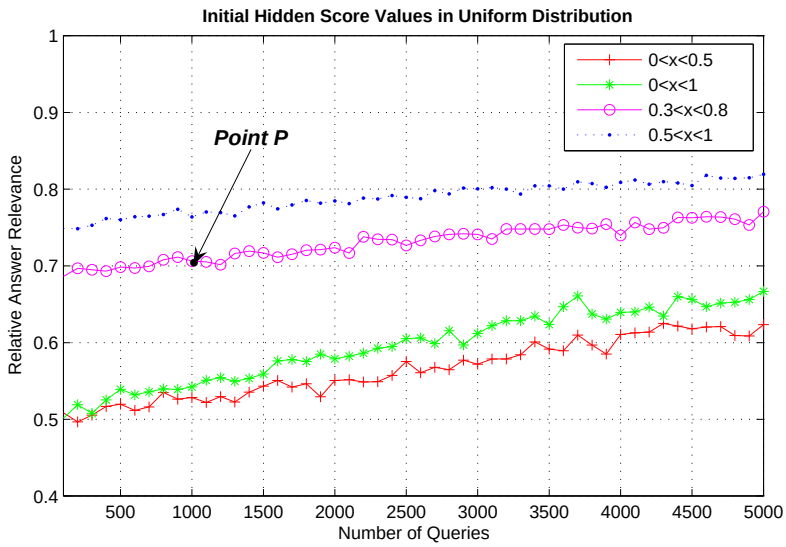
In the first series of runs, we assign initial S_H with random values x which follow uniform distribution in different ranges ($0 < x < 0.5$; $0 < x < 1$; $0.3 < x < 0.8$; $0.5 < x < 1$). Fig. 6 (a) shows the average R values for the preceding 100 queries. For example, to plot the point P in Fig. 6 (a), we sum the R-values obtained for the 901^{th} to the 1000^{th} query and then divide the sum by 100. From these run results, the values of R of these runs are increasing. In other words, all of the runs are convergent disregarding their initial S_H. In addition, the convergence behaviour of run with $0 < x < 0.5$ is similar to the one of run with $0 < x < 1$. However, the convergence rate of the run with $0 < S_H < 1$ is the fastest when since its curve gives the greatest slope.

Then, another series of runs were performed by assigning initial S_H with random values which follow normal distribution of different mean and standard deviation (s.d.). Fig. 6 (b) shows the average R values for every 100 queries with mean=0.5 and s.d.=0.5, 0.25, and 0.1 respectively, while Fig. 6 (c) shows the average R values for every 100 queries with mean=0.75 and s.d.=0.5, 0.25, and 0.1 respectively. In these runs that showing in Fig. 6 (b, c), all the values of R of these runs are increasing. Such that, both of these runs are convergent disregarding their initial S_H. Fig. 6 (b) shows that the runs of s.d.=0.25 and s.d.=0.5 are similar while these runs are having the same mean value (0.5), while the one with s.d.=0.5 attained the fastest converging rate by comparing with the runs with mean=0.5. Furthermore, the run with s.d.=0.5 in Fig. 6 (c) also attained the fastest converging rate by comparing with the runs with mean=0.75.

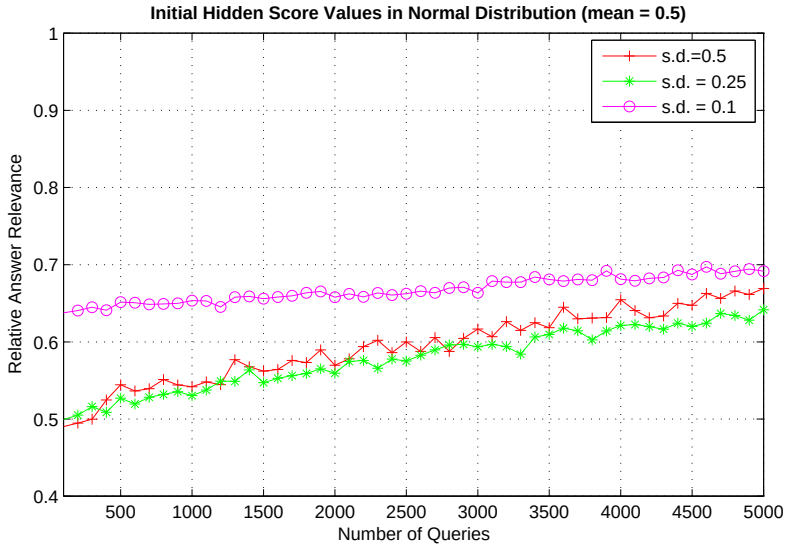
By comparing all of the runs, with different distribution of the initial S_H, we can conclude the following. All of the runs would be convergent irrespective of the initial value of S_H. When the S_H value of the run starts with a lower mean value, it attains a faster converging rate since more rooms are provided for the index to become convergent. Also, the run with uniform distribution of S_H ($0 < x < 1$) is nearly identical to the run with normal distribution of S_H (mean=0.5, s.d.=0.5) since the distribution of random S_H for these two runs are alike.

6 Conclusions & Future Works

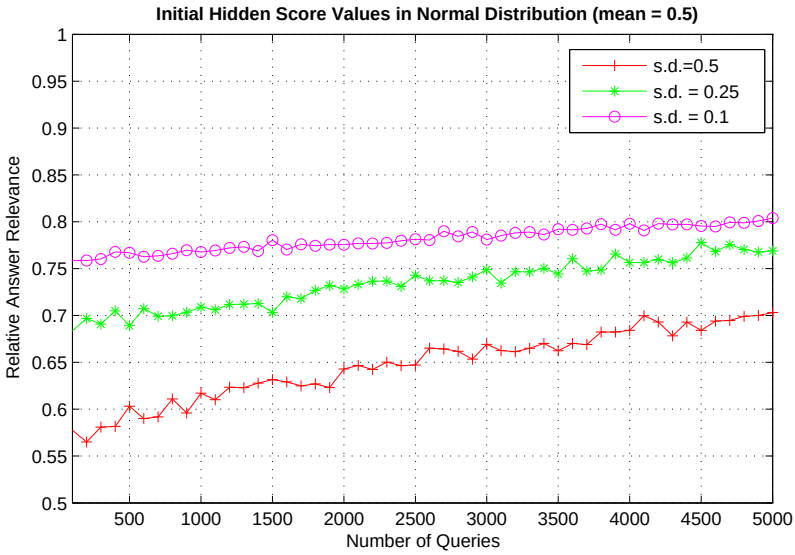
We presented a collaborative indexing approach for enabling multimedia retrieval within a huge amount of multimedia data objects. Our indexing approach helps to discover multimedia resources systematically by keeping track of the user query behaviour. By analyzing the search information, the user relevance feedback helps the index hierarchy to evolve towards to users' desired preferences. Thus, user satisfaction would be maximized. Our experimental result shows that the index would successfully converge after successive use. In the future, we will focus on examining the index convergence behaviour.



(a) Initial S_H Follows Uniform Distribution



(b) Initial S_H Follows Normal Distribution (mean=0.5)



(c) Initial S_H Follows Normal Distribution (mean=0.75)

Figure 6. Test with Different Initial S_H Values.

References

[1] I. A. Azzam, C. H. C. Leung, and J. F. Horwood. Implicit concept-based image indexing and retrieval. In Y.-P. P. Chen, editor, *Proceedings of the 10th International Multimedia Modeling Conference (MMM 2004), 5-7 January 2004, Brisbane, Australia*, page 354. IEEE Computer Society, 2004.

[2] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, 2002.

[3] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[4] J. F. Gantz, D. Reinesel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz. The expanding digital universe: a forecast of worldwide information growth through 2010. *IDC White Paper*, March 2007.

[5] T. Gevers and A. W. M. Smeulders. Image search engines - an overview. 2004.

[6] J. Go'mez and J. L. Vicedo. Next-generation multimedia database retrieval. *IEEE MultiMedia*, 14(3):106–107, 2007.

[7] G. Goth. Multimedia search: Ready or not? *IEEE Distributed Systems Online*, 5(7), 2004.

[8] R. Hawarth and H. Buxton. Conceptual-description from monitoring and watching image sequences. 18, 2000.

[9] C. Leung, J. Liu, W. S. Chan, and A. Milani. An architectural paradigm for collaborative semantic indexing of multimedia data objects. In *VISUAL '08: Proceedings of the 10th International Conference on Visual Information Systems*, Salerno, Italy, 2008. (To Appear).

[10] C. H. C. Leung and J. Liu. Multimedia data mining and searching through dynamic index evolution. In *VISUAL '07: Proceedings of the 9th International Conference on Visual Information Systems*, pages 298–309, Shanghai, China, 2007.

[11] H. Mu'ller, W. Mu'ller, D. M. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. 22(5), 2001.

[12] P. Over, C. H. C. Leung, H. H.-S. Ip, and M. Grubinger. Multimedia retrieval benchmarks. *IEEE MultiMedia*, 11(2):80–84, 2004.

[13] F. Saint-Jean, A. Johnson, D. Boneh, and J. Feigenbaum. Private web search. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 84–90, New York, NY, USA, 2007. ACM.

[14] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM.

[15] A. M. Tam and C. H. C. Leung. Structured natural-language descriptions for semantic content retrieval of visual materials. *J. Am. Soc. Inf. Sci. Technol.*, 52(11):930–937, 2001.

[16] V. Vinay, K. Wood, N. Milic-Frayling, and I. J. Cox. Comparing relevance feedback algorithms for web search. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1052–1053, New York, NY, USA, 2005. ACM.

[17] R. C. F. Wong and C. H. C. Leung. Automatic semantic annotation of real world web images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. (To Appear).

[18] Y. Xu, K. Wang, B. Zhang, and Z. Chen. Privacy-enhancing personalized web search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 591–600, New York, NY, USA, 2007. ACM.

[19] B. Yang and A. R. Hurson. Ad hoc image retrieval using hierarchical semantic-based index. In *AINA '05: Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 629–634, Washington, DC, USA, 2005. IEEE Computer Society.

[20] Y. Yoshida, T. Ueda, T. Tashiro, Y. Hirate, and H. Yamana. What's going on in search engine rankings? In *AINAW '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, pages 1199–1204, Washington, DC, USA, 2008. IEEE Computer Society.

Automatic image content annotation and indexing

Chun Fan WONG

Abstract

As the number of web images is increasing at a rapid rate, searching them semantically presents a significant challenge. Many raw images are constantly uploaded with little meaningful direct annotations of semantic content, limiting their search and discovery. In this paper, we propose an extension of image annotation models which are using ontology-based tree expansion and contextual feature-based index expansion. Our system is evaluated quantitatively using more than 100,000 web images and around 1,000,000 tags. Experimental results indicate that this approach is able to deliver highly competent performance.

1 Introduction

Since the past decade, image retrieval has been become one of the most popular activities on Internet. As the number of images available in online repositories is growing dramatically, exploring the frontier between image and language is an interesting and challenging task. Research in image retrieval has reflected the dichotomy inherent in the semantic gap, and is divided between two main categories: concept-based image retrieval and content-based image retrieval. The former focuses on retrieval by objects and high-level concepts, while the latter focuses on the low-level visual features of the image.

Low-level visual features are indicated by visual content descriptors in order to support users in accessing the knowledge embedded in images. These methods aim at capturing image similarity by relying on some specific characteristic of images; typically, these models are based on color, texture and shape [4, 6, 9, 11, 14, 23, 30, 32]. As discussed in [31], in order to compute these descriptors, the image often has to be segmented into parts, which aims to determine image objects. Current methods of image segmentation include [8, 12, 16, 17, 24, 29]: partitions, sign detection, region segmentation. They compute general similarity between images based on statistical image properties [1–3, 18, 22, 26, 27]. Some studies [12, 20] include users in a search loop with a relevance feedback mechanism to adapt the search parameters based on user feedback. Semantic annotation of the image database combined with a region based image decomposition is used, which aims to extract semantic properties of images based on spatial distribution of color and texture properties [9,11,14,15,23,30,32]. However, an advantage of using low-level features is that, unlike high-level concepts, they do not incur any indexing cost as they can be extracted by automatic algorithms. In contrast, direct extraction of high-level semantic content automatically is beyond the capability of current technology. Some research [5] focuses on implicit image annotation which involve an implicit and, in consequence, augments the original indexes with additional concepts that are related to the query.

With the advent of Semantic Web technology, ontology is playing a key role as the core element of knowledge representation architecture in Semantic Web. Some effort [7, 13, 19, 25, 28] has been made for image retrieval using Semantic Web techniques.

Our work is related to generative modelling approaches. In [31], a semantic annotation technique named Automatic Semantic Annotation (ASA) approach is developed which is based on the use of image parametric dimensions and metadata . Using decision trees and rule induction, a rule-based approach to formulate explicit annotations for images fully automatically is developed, so that, semantic query such as ”sunset by the sea in autumn in New York” can be answered and indexed purely by machine. In this paper, we propose an extension of such image annotation models by using ontology-based tree expansion and contextual feature-based index expansion. Our system is evaluated quantitatively using more than 100,000 web images and over 990,000 tags. Experimental results indicate that this approach is able to deliver highly competent performance.

2 Expansion algorithms

2.1 Ontology-based tree expansion

In certain applications, the presence of particular objects in an image often implies the presence of other objects. If term $U \Rightarrow V$, and if only U is indexed, then searching for V will not return the image in the result, even though V is present in the image. The application of such inferences

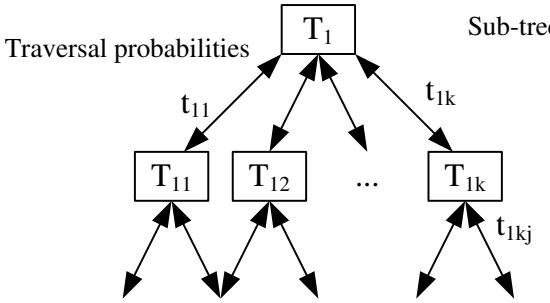


Figure 1. Ontology expansion tree

will allow the index elements T_i of an image to be automatically expanded according to some probability which will be related to the underlying ontology of the application.

There are two types of expansion:

(a) Aggregation hierarchical expansion

This relates to the aggregation hierarchy of sub-objects that constitute an object. The objects can be classified as:

(i) concrete, where the relevant objects are well-defined (e.g. an "orchestra" expanded to violins, trumpets, clarinets etc.)

(ii) abstract, where the objects are not concretely defined (e.g. although "conflict" is not a definite visual object, it contains certain common characteristics).

Associated with each branch is a tree traversal probability t_{ij} (Fig. 1) which signifies the probability of occurrence of the branch index given the existence of the parent index. In general, the traversal probabilities of different object classes exhibit different characteristics, with $t_{ij} > t'_{mn}$ for t_{ij} belonging to the concrete object class, and t'_{mn} belonging to the abstract object class.

(b) Co-occurrence expansion

This relates to the expectation that certain semantic objects tend to occur together. The relevant weighting may be expressed as a conditional probability given the presence of other objects. An expansion to associate an image object O_j given the presence of object O_i is taken to be indexable when $Prob[O_j|O_i] \geq h$, where h is a preset threshold value that depends on the tradeoff between precision and recall performance of the system. More generally, complex probabilistic rules taking the form $Prob[O_j|O_1, ..., O_n] \geq h$ will be applied. The ontology expansion tree may be traversed bi-directionally in the course of the expansion. Top-down traversal will lead to an expansion factor > 1, while bottom-up traversal will have an expansion factor < 1 at each level of expansion. There are, in general, many sub-trees whose roots are the nodes of the ontology expansion tree. Each sub-tree may be fully expanded, and it has an expansion reliability factor $0 < r < 1$, which signifies the dependability and completeness of the associated expansion. For high precision retrieval ($\pi \approx 1$), only sub-trees having a significant reliability factor need to be traversed, and nodes with a small value for r will be pruned. Decision rules linking expansibility with π and r can be determined.

2.2 Contextual feature-based index expansion

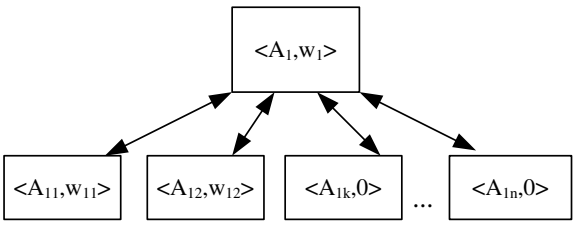
Here, we shall establish associations between low-level features with high-level concepts, and such associations will take the following forms.

(a) Associating basic features with semantic concepts

The presence of certain low-level features F may suggest a finite number of m object possibilities. The expansion will be carried out by examining all $Prob[O_j|F] > 0, j = 1, 2, ..., m$. The object O_k that maximizes the probability expression will be indexed. Sometimes, a combination of basic features may be used to infer the presence of high-level concepts for inclusion in the semantic index. Thus, in more complex situations, maximization will need to be carried out on the probabilities $Prob[O_j|F_1, ..., F_n]$ which correlates an object with multiple feature occurrences.

(b) Exploiting contextual constraints

Basic features alone may not be sufficient to infer the presence of specific objects, but such features if augmented by additional information may lead to meaningful inferences. When a particularly context is known, a concept may be indexed more precisely. Such contextual information will typically be provided



A particular annotation class A1 may exhibit a certain degree of ambiguity covering the n sub-classes $A_{11}, ...A_{1n}$. Our disambiguation algorithm will partially pinpoint the precise annotations by making use of a set of modal images, where each annotation concept C_i will have associated with it a representative set of modal images that are external to the images to be annotated.

Figure 2. Annotation Disambiguation

through ontological expansion, which may lead to the creation of a new index term, or a revision of the score of an existing index term. In general, we will need to consider the probabilities $Prob[O, |\overrightarrow{F}, \overrightarrow{O}]$, where $\overrightarrow{F}, \overrightarrow{O}$, are respectively the available feature set (an element of which may be a vector) and the object set. This will give rise to an iterative feedback loop where the determination of new objects will lead to new meaningful feature-object combinations, where further objects may be determined.

2.3 Index Scoring and Ranking

The initial creation of an index term will result in a certain probabilistic score, which indicates its probability of being present in an image, and every index term T_i in the index will have a score $p_i = f(t_{ij}, r)$ associated with it, which will be used for the ranking and delivery of search results. Such a score p_i may be caused by the uncertainty of the index terms in earlier stages or by the sub-tree reliability factor responsible for generating it or both. The algorithms for computing these probabilistic scores will be formulated and developed. In addition to initial allocation, index scores may be updated and revised. Such revision can take two forms: (i) long-term revision that causes a change to the database, and (ii) short-term revision that does not require changing the database. The former is caused by ontology-based expansion or feature-based expansion. The latter is caused by iterative relevance feedback as a result of user intentions and selection preferences for a particular query. Algorithms for short-term revision are different from the normal score updates and will need to be separately developed to achieve good overall retrieval performance. Short-term score re-computation may take place several times in the course of a query. It is expected that sampling from image database benchmarks will be used to determine the initial score and Bayesian algorithms will be used to effect the revision. In addition to these scores, search results will be determined by a query value for a given image which results from matching the relative importance of different concepts required by a given query with the occurrence of those concepts in the image. Final results ranking of images will be done according to such values.

2.4 QBE Disambiguation and Computation of Annotation Weights

As shown in Figure 2, a particular annotation class A_1 may exhibit a certain degree of ambiguity covering the n sub-classes $A_{11}, ..., A_{1n}$. Our disambiguation algorithm will partially pinpoint the precise annotations by making use of a set of modal images, where each annotation concept C_i will have associated with it a representative set of modal images that are external to the images to be annotated.

We first establish N simple concepts of interest $C = C_1, ..., C_N$, which collectively is referred to as the concept dictionary, whose content may be augmented over time. These concepts range over all M annotation classes $A_1, ..., A_M$ established from the previous step. Let concept C_i be associated with a set of k_i distinct modal images

$$S_i = \{I_i, ..., I_k\}, \;\; i = 1, ..., N, \tag{1}$$

where for $i \neq j, S_i \bigcap S_j$ need not be empty to cater for compound concepts. Each modal image I_i has a representativeness coefficient $|I_i|, (0 < |I_i| \leq 1)$ associated with it which indicates its relative representativeness of the particular concept Ci. Thus, the representativeness coefficient induces an ordering on S_i, and we take

$$|I_1| \geq |I_1| \geq ... \geq |I_{k_i}| \tag{2}$$

Instead of directly using SIFT techniques to process an image to obtain annotation of images on specific concepts C_i, we apply QBE to the image database using the first p images from S_i as a set of p separate queries. Each of these QBE queries will return a set of target images $T(I_j)$. The set of images $T(I_j)$ in the collection will then be annotated with a computed precision weight. For a given parameter p, images contained in $T(I_1) \bigcap T(I_2) \bigcap ... \bigcap T(I_p)$ will have the highest probability of containing the concept C_i compared with those present in $T(I'_1) \bigcap T(I'_2) \bigcap ... \bigcap T(I'_m)$,

for $I'_j \varepsilon S_i, m < p$. The annotation weight is a significant element in search and query processing. The algorithm takes into account

1. the controllable parameters p and h (see below),

2. the uncontrollable parameter m.

In this manner, the concepts $C_1, ..., C_N$ are systematically propagated to images in the database, and these are annotated by the corresponding concepts, with each annotation having a weight wi attached to it (see Figure 2). The set S_i may be updated and adjusted as a result of these processes and may be optimized over time. To optimize annotation efficiency, it would be desirable to minimize p, while maximizing $\sum_{1 \leq i \leq p} |I_i|$. In executing this process of annotation by concept propagation, a pre-defined threshold h is adopted, and the number of modal images p included for concept querying will be the smallest integer M such that

$$\sum_{i=1}^{M} |I_i| \geq h. \tag{3}$$

For images in the collection not covered in the concept dictionary, a corresponding weight of 0 will be assigned (Figure 2).

3 Experimental Evaluation

Our main purpose in introducing an ontology into the image retrieval problem and using the sub-objects as surrogate terms for general queries is to improve the precision in the image sets. In this evaluation, we mainly focus on the ontology-based tree expansion and contextual feature-based index expansion with QBE disambiguation.

The index elements are organized and used to build the basic content index within a relational database. The relational database is designed for maximum query effectiveness by distributing the semantic elements across different relations. A further index is built on top of these relations to support rapid discovery.

The effectiveness of our approach is evaluated experimentally. A set of standard evaluation queries are used for experimentation. Comparison is made between base-level indexing and the expanded level indexing, and the widely accepted measures of retrieval performance of precision, recall, fallout, and F-score are used to assess system performance.

To numerically assess the accuracy and effectiveness of our annotation approach, we have retrieved 103,521 sets of images with 991,074 associated tags from flickr.com which are a popular photo sharing web site and online community platform offering a fairly comprehensive web-service API that allows developers to create applications that can perform almost any function on images.

Here we exploit WordNet and OpenCYC to assist our experiments. The former is an application of semantic lexicon for the English language and the latter is a general knowledge base and commonsense reasoning engine. In order to produce a list of tags that are more intuitively usable, we perform a semantic lexicon checking by WordNet, where numeric, symbol characters misjudge phase have been removed, and in consequence 289,399 tags with 7,982 keywords are formed.

The quality assessment of the machine-inferred boundaries between parts of the depicted scenes is based on the precision. In our evaluation, we decide that a relevant image must include a representation of the category in such a manner that a human should be able to immediately associate it with the assessed concept.

3.1 Results

3.1.1 Ontology-based tree expansion

In relation to image acquisition, many images may be broken down to few basic scenes, such as nature and wildlife, portrait, landscape and sports. In the case of aggregation hierarchical expansion of ontology-based tree expansion, we decided to test our system using the aggregation hierarchy of basic categories ”night scenes” and extend the image hierarchy to find a sub-scene ”night scene of downtown”, ”downtown” can be expanded to ”business district”, ”commercial district”, ”city center” and ”city district”, while ”city district” can be expanded to ”road”, ”building”, ”architecture”, ”highway” and ”hotel”.

In [31] by using decision trees and rule induction, a rule-based approach to formulate explicit annotations for images fully automatically has been developed. To extend the approach, firstly, we annotate night scenes based on the prior rule-based approach to extract 422 out of 103,527 images. We also gather 1108 tags associated with those images and totally 417 unique terms are formed. We list the top 117 out of 417 unique terms list in Fig. 3,

We present the results of the evaluation in Fig. 4. For branch t_{14}, the traversal probability signifies the probability of occurrence of the branch index given the existence of the parent index with 100% precision rate (”downtown” $\Rightarrow$ ”city”). For each branch t_{14j}, where $1 < j \leq 5$, it is expanded from the original sub-tree (”city district”); traversal probabilities are varied, ranged from 33.3% to 75%.

3.1.2 Contextual feature-based index expansion with QBE disambiguation

To establish associations between low-level features with high-level concepts, associating basic features with seman-

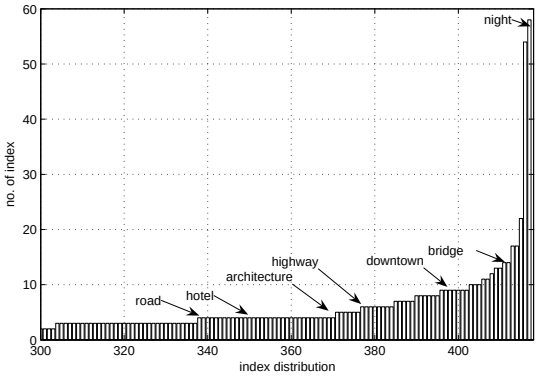


Figure 3. Index distribution associate with night scene images

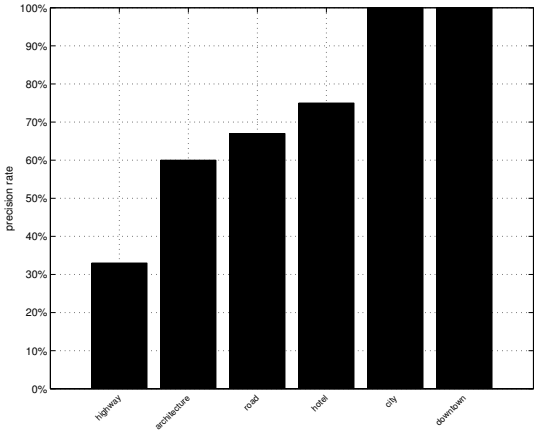


Figure 4. Experimental results on aggregation hierarchical expansion

tic concepts may be applied to arbitrary images for inclusion in the semantic index. Edge detection is a terminology in image processing and computer vision, particularly in the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. Here, we adapt edge detection algorithms [10, 21] to extract high-level concepts from low-level features.

From [10], the framework near-circular Gaussian-based image derivative operators have been developed via the use of a virtual mesh and are proven to reduce angular error when detecting edges over a range of orientations. The edge detection operators are based on first and second derivative approximations, corresponding to a first directional derivative $\partial u/\partial b \equiv \underline{b} \cdot \underline{\nabla u}$ and a second directional derivative $-\underline{\nabla} \cdot (B\underline{\nabla u})$, and are defined by the functionals [10]

$$E_i^{\delta}(U) = \int_{\Omega} \underline{b_i} \cdot \underline{\nabla} U \zeta_i^{\delta} d\Omega \tag{4}$$

and

$$Z_i^{\delta}(U) = \int_{\Omega} \underline{\nabla} U \cdot (B_i \underline{\nabla} U \zeta_i^{\delta}) d\Omega \tag{5}$$

Here $B = \underline{b}\ \underline{b}^T$ and $b = (cos\theta, sin\theta)$ is the unit direction. The special case of the Laplacian operator is represented by Z_i^{δ} with B taken to be the identity matrix [10].

Here, we select one "downtown" image manually from the image set and through the use of edge detection algorithms before Query-by-example similarity matching. We have carried out evaluation (shown in Fig. 6 by comparing the original Automatic Semantic Annotation (ASA) approach) with our approach which combines the original ASA approach with vertical edge detection algorithms and the use of human tags. Our experiment indicates that tags by human deliver excellent precision rate with 100% precision but this tagging approach relies heavily on human involvement. For the ASA Approach combining with edge detection algorithms, the precision rate grows to 87.1%. Clearly, compared to annotation without the contextual feature-based index expansion enabled, the performance is around 52.8%. From the joint application of these, we can formulate semantic annotations for specific image fully automatically and index images purely by machine without any human involvement.

4 Conclusion and future works

In this paper, we propose an extension of image annotation models which uses ontology-based tree expansion and contextual feature-based index expansion with QBE disambiguation. Our system is evaluated quantitatively, and

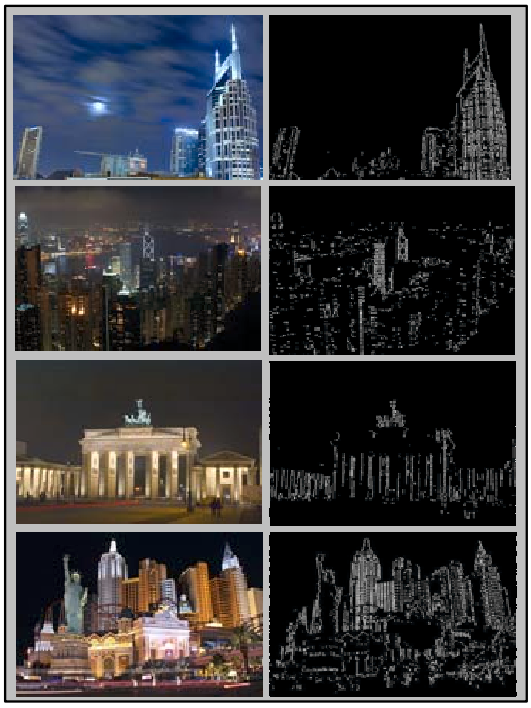


Figure 5. Contextual feature-based index expansion with edge detection algorithms

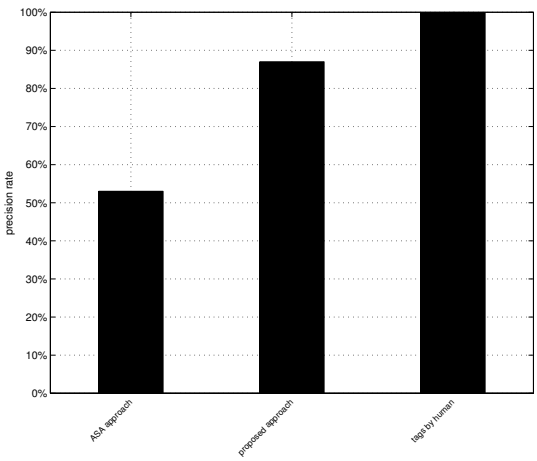


Figure 6. Experimental results on contextual feature-based index expansion

experimental results indicate that this approach is able to deliver highly competent performance. Our approach, not only demonstrates the applicability of ontology to the image annotation problem, but also using the sub-objects as surrogate terms for general queries is to improve the precision in the image sets.

References

[1] G. Amato and C. Meghini. Combining features for image retrieval by concept lattice querying and navigation. *iciapw*, 0:107–112, 2007.

[2] J. Amores, N. Sebe, and P. Radeva. Context-based object-class recognition and retrieval by generalised correlograms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1818–1833, October 2007.

[3] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 02:268–275, 2004.

[4] I. Azzam, A. G. Charlapally, C. H. C. Leung, and J. F. Horwood. Content-based image indexing and retrieval with xml representations. *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong*, pages 181–185, 2004.

[5] I. A. Azzam, C. H. C. Leung, and J. F. Horwood. Implicit concept-based image indexing and retrieval. In *Proceedings of the IEEE International Conference on Multi-media Modeling*, pages 354–359, Brisbane, Australia, January 2004.

[6] I. A. Azzam, C. H. C. Leung, and J. F. Horwood. A fuzzy expert system for concept-based image indexing and retrieval. *mmm*, 0:452–457, 2005.

[7] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[8] Y. Chen, J. Z. Wang, and R. Krovetz. Content-based image retrieval by clustering. In *MIR'03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 193–200, New York, NY, USA, 2003. ACM.

[9] J. S. Cho and J. Choi. Contour-based partial object recognition using symmetry in image databases. In *SAC'05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1190–1194, New York, NY, USA, 2005. ACM Press.

[10] S. Coleman, B. Scotney, and D. Kerr. Integrated edge and corner detection. *iciap*, 0:653–658, 2007.

[11] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[12] R. Datta, J. Li, and J. Z. Wang. Content-based image retrieval: approaches and trends of the new age. In *MIR'05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262, New York, NY, USA, 2005. ACM.

[13] L. Fan and B. Li. A hybrid model of image retrieval based on ontology technology and probabilistic ranking. *wi*, 0:477–480, 2006.

[14] J. Gausemeier, J. Fruend, C. Matysczok, B. Bruederlin, and D. Beier. Development of a real time image based object recognition method for mobile ar-devices. In *AFRIGRAPH'03*, pages 133–139, New York, NY, USA, 2003. ACM Press.

[15] K. Hornsby. Retrieving event-based semantics from images. *ismse*, 00:529–536, 2004.

[16] M. Jian, J. Dong, and R. Tang. Combining color, texture and region with objects of user's interest for content-based image retrieval. *SNPD'07*, 01:764–769, 2007.

[17] R. Krishnapuram, S. Medasani, S. H. Jung, Y. S. Choi, and R. Balasubramaniam. Content-based image retrieval based on a fuzzy approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1185–1199, 2004.

[18] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

[19] H. Lieberman and H. Liu. Adaptive linking between text and photos using common sense reasoning. *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, AH 2002, Malaga, Spain*, pages 2–11, May 2002.

[20] D. Liu and T. Chen. Content-free image retrieval using bayesian product rule. *IEEE International Conference on Multimedia and Expo*, 0:89–92, 2006.

[21] J. Manikandan, B. Venkataramani, and M. Jayachandran. Evaluation of edge detection techniques towards implementation of automatic target recognition. *iccima*, 2:441–445, 2007.

[22] A. P. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *MULTIMEDIA'07*, pages 991–1000, New York, NY, USA, 2007. ACM.

[23] R. Pawlicki, I. Kókai, J. Finger, R. Smith, and T. Vetter. Navigating in a shape space of registered models. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1552–1559, 2007.

[24] A. Perina, M. Cristani, and V. Murino. Natural scenes categorization by hierarchical extraction of typicality patterns. *ICIAP'07*, pages 801–806, 2007.

[25] A. Popescu, G. Grefenstette, and P. A. Moellic. Using semantic commonsense resources in image retrieval. In *SMAP '06*, pages 31–36, Washington, DC, USA, 2006. IEEE Computer Society.

[26] K. Stevenson and C. H. C. Leung. Comparative evaluation of web image search engines for multimedia applications. *IEEE International Conference on Multimedia and Expo*, 0:4 pp., 2005.

[27] Y. Sun, S. Shimada, and M. Morimoto. Visual pattern discovery using web images. *MIR'06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 127–136, 2006.

[28] A. M. Tam and C. H. C. Leung. Semantic content retrieval and structured annotation: Beyond keywords. *ISO/IEC JTC1/SC29/WG11 MPEG00/M5738, Noordwijkerhout, Netherlands*, March 2000.

[29] N. Vasconcelos. From pixels to semantic spaces: Advances in content-based image retrieval. *Computer*, 40(7):20–26, 2007.

[30] J. Vogel, A. Schwaninger, C. Wallraven, and H. H. Bülthoff. Categorization of natural scenes: Local versus global information and the role of color. *ACM Transactions on Applied Perception*, 4(3):19, 2007.

[31] R. C. F. Wong and C. H. C. Leung. Automatic semantic annotation of real world web images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)*, 2008.

[32] T. Zöller and J. M. Buhmann. Robust image segmentation using resampling and shape constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1147–1164, July 2007.

Mining of ownership for online forum participants

Tianjie ZHAN

Abstract

The mining of user preference from web data is a very important topic. The paper is focused on the discovery of ownership of online forum participants, which would help for product promotion and topical feedback if owned or not. With modification to the latent dirichlet allocation (LDA) model, we present a constrained LDA for the ownership problem. And there are ownership-associated topics generated from each class, which are of great importance for revealing the reason of the state of the ownership through the topical feature. Data in practical use have been used for test of the model. We present the experimental results showing this model could not only predict the state of the ownership accurately but also detect the time period of the ownership continuance, compared with transitional classifier such as SVM which misclassified more than 30% of the test data and switch state of the ownership frequently deviating of the practical situation severely.

1. Introduction

In the modern life, online forum plays an important role as a mass media. Especially, in the forum related to electrical product, there are kinds of people passing different of message about their equipment, including information sharing, asking question, giving suggestion and making comment on the product. So it is necessary to collect such information to reveal the potential buyers, concern or complaint about the product, which will be help for accurate one-to-one advertising, further product modification and making marketing strategy. Not only the state of ownership, but also how the state kept and switched in time sequence is of large significance for analysis of owners or potential customers. To solve the problem of discovery of ownership by text analysis for the messages posted in the online forum, classifiers sensitive to the time sequence and topic generation are necessary rather than the traditional classifiers such as SVM [10]. Aside mining of the topical feature for each state of ownership is important too. As a result, it is motivated to build a text classifier fit for both of the time feature and topical feature of ownership.

In recent years, the most popular model for text processing is latent dirichlet allocation model (LDA by Jordon and Blei 2003 [3]). Semantically learning could be done by generating topics. Extended models such as supervised topic model [1], Multi-Grain topic model was induced for supervised learning. There are also a few applications in classification such as in web spam classification [7]. In application on web content or customers review, M. Hu and B. Liu have a good example in [4]. Even in field of statically debugging, LDA could be extended to delta-LDA [2] giving power of semantically text analysis.

In analysis of time sequence, McCallum combined the time attribute into LDA giving the A non-markov continuous time model [6] to reveal the topic trend.

Combination of the supervised learning by LDA and time attribute, we present a constrained LDA (cLDA) to approach to the problem of ownership discovery and ownership associated topics detecting for on line forum users. This paper would be organized as follow. In section 2, ownership problem will be defined, and the cLDA would be introduced. In section 3, algorithm for making prediction would be discussed. Experimental result for a recent data set from a popular digital camera website, given by the constrained LDA and LIBSVM [10] for comparison would be illustrated, and the topical feature of each state of ownership is also illustrated in section 4. We conclude and discuss the future work in section 5.

2. The constrained LDA

We will first describe the problem of ownership discovery; review the standard LDA; and then present the constrained LDA. The standard LDA is efficient for general text processing but regard of the time attribute. But the constrained LDA could learn the supervised information from the labeled delta under which inference of class variable of sequential data on time would be taken. Without considering the time

attribute in the training stage could decreasing the need of sequential data and put more attention on the discriminative words related to the state of ownership which could benefit more in the topic analysis stage.

2.1. The problem of discovery of ownership of online forum

People may share information, ask question, give suggestion and even take some large social function in the on-line forum. Information from web is of large value for product promotion and online business. One of the most important of which is the ownership of the products, especially in electrical products which is expensive always. Not only the state of ownership, owned or not owned, but also the continuous time period of the state is crucial, so the time sequence is the indispensable attribute which is out of consideration of standard LDA, and less in its extended version except few such as TOT [6].

Since the ownership is individual, the messages posted should be organized by authors and ordered by time sequence, rather than traditionally by threads. How ever, in the stage of learning the probability of words associated with state of ownership, we just focused on the discriminative line of different classes, regardless of time and authors.

In this paper, a message d is a vector of N_d word tokens, w_d, where each w_{id} denotes the word index in the vocabulary of size V . A collection of D messages is defined by $\mathrm{M} = \{(w_1, a_1, c_1), \ldots, (w_D, a_D, c_D)\}$ where $c_i \in \{-1, 0, 1\}$ represents the state of the ownership of one specific product, as "not owned", "not mentioned", and "owned", a_i represents the authors of the message w_i . Here we define one more class of "not mentioned" as it could be detected the ownership in some messages which would be noise if allocated as either of state of "owned" or "not owned". In the inference stage, all messages of the same author a_i will be collected as a matrix defined by $m_i = \{w_{i1}; \ldots; w_{in_i}\}$ where each column w_{ij} is one message posted by author a_i in the order of time sequence.

2.2. Topics model

LDA is a popular generative model for document modeling in recent years. The process corresponds to the graphical model shown in Figure1. In LDA, ϕ denotes matrix of the words distribution with a multinomial distribution in a column for each of T topics which is drawn from the Dirichlet(β) prior independently. θ is the matrix of the topics distribution with a multinomial distribution in a column related to a document drawn from the Dirichlet(α) prior. In generative process for a document, first topic z is drawn from the θ related to the document, and then w is drawn from ϕ_z corresponding to z. Algorithm such as variational inference, collapse Gibbs sampling [9] have been presented to estimate the parameters ϕ and θ. Efficient estimation of hyperparameter α has been given as an non-iterative method in [9].

2.3. The constrained LDA

There are many shortcomings of latent models in supervised setting [11], so the unsupervised LDA is not fit for the problem of ownership discovery. We are motivated to introduce the constrained LDA model shown in Figure 2.

The constrained LDA is constructed as

(1) There are 3 hyperparameters α_c for each class, the class variable $c \in \{-1, 0, 1\}$ is visible, and $\alpha = \{\alpha_c\}$.

(2) The topics T is separated by class as T_{-1}, T_1, T_0 . So the parameters ϕ are separated as $\phi_{-1}, \phi_0, \phi_0$, and β as $\beta_{-1}, \beta_1, \beta_0$. The number of topics in T_i is N_{ti} and $N_T = N_{t-1} + N_{t0} + N_{t1}$.

(3) The rest of model is identical to LDA.

Each message is assumed as a mixture of three classes corresponding to the mixture of all topics T. But once the label c is known, the message could be set to just cover the topics related to the class c, T_c. It could be achieved by setting all the elements corresponding to topics related other classes except c to zero and initially allocate words only to class-related topics randomly T_c before Gibbs sampling so that there will be no words allocated out of T_c .

The generative process is described as

- $p(\theta \mid c, \alpha_c) \sim Dir(\alpha_c)$
- $p(z \mid \theta) \sim Multi(\theta)$
- $p(\phi \mid \beta) \sim Dir(\beta)$
- $p(w \mid z, \phi) \sim Multi(\phi_z)$ N_d

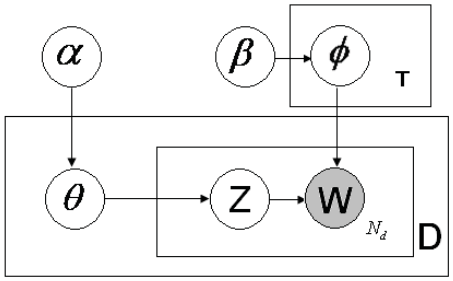


Figure 1: Graphical model of LDA

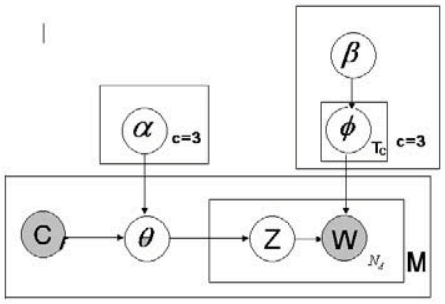


Figure 2: Graphical model of the constrained LDA

The conditional probability of collection of messages labeled as c is simplified as $p(w \mid c, \alpha, \beta)$, by integrating out the hyperparameters, giving

$$p(w \mid c) = \sum_Z p(w \mid z) p(z \mid c), \quad (1)$$

where $p(w \mid z) = \prod_i^{N_T} \int p(\phi_i \mid \beta) \prod_j^W \phi_{ij}^{n_i^j} d\phi_i$, (2)

and $p(z \mid c) = \prod_d^D \int p(\theta_d \mid c, \alpha) \prod_i^{N_T} \theta_{di}^{n_i^d} d\theta_d$. (3)

Here W denotes the size of the vocabulary, n_i^j denotes the number of times of j^{th} word assigned to the i^{th} topic, and n_i^d denotes the number of words in message d assigned to i^{th} topic. Identical to LDA, ϕ_{ij} is the probability of i^{th} word conditional on topic j, and θ_{di} is the probability of message d covering topic j.

2.3.1. Inference.

The hidden variables z, θ, φ and α is our target parameters used for class prediction. We could draw z samples from the posterior $p(z \mid w, c)$ approached by MCMC, especially by Gibbs sampling. Here we use the collapsed Gibbs sampling [9] to draw samples of z_i from $p(z_i \mid z_{-i}, c, w)$ where z_{-i} denotes all other topic variables z except z_i .The posterior could be inferred using Bayesian rules as

$$p(z_k = i \mid z_{-k}, c, w) \propto p(z_k = i, z_{-k}, w \mid c). \quad (4)$$

From the conditional independence inferred from the graphical model shown in Figure 2, the joint probability $p(z, w \mid c)$ could be factorize as

$$p(z, w \mid c) = p(w \mid z) p(z \mid c). \quad (5)$$

The factor in the right of (3) is given by (1) and (2). Similar to the Gibbs sampler for LDA [9], integrating out the hyperparameters, (1) and (2) could be computed as

$$p(w \mid z) = \prod_i^{N_T} [\frac{\Gamma(\sum_{j'}^W \beta_{j'}^i)}{\Gamma(\sum_{j'}^W \beta_{j'}^i + n_*^i)} \prod_j^W \frac{\Gamma(n_j^i + \beta_j^i)}{\Gamma(\beta_j^i)}] \quad (6)$$

$$p(z \mid c) = \prod_d^M [\frac{\Gamma(\sum_{i'}^{N_T} \alpha_{i'}^c)}{\Gamma(\sum_{i'}^{N_T} \alpha_{i'}^c + n_*^d} \prod_i^{N_T} \frac{\Gamma(n_i^d + \alpha_i^c)}{\Gamma(\alpha_i^c)}] \quad (7)$$

Here n_*^i is the sum of the number of words assigned to the topic i in all messages regardless of the label, and n_*^d is the number of words in message d . β_j^i is the hyperparameter associated with the j^{th} word related to topics i , where $\beta^i \in \{\beta_c\}$ depends on which class topic $i \in T_{-1} \mid T_0 \mid T_1$ belong to. α_i^c is the elements associated with topic i in α_c .

Using the probability product rule with (4) and (5), (3) could be given as

$$p(z_k = i \mid z_{-k}, c, w) = \frac{1}{C} (\frac{n_{-k,j_k}^i + \beta_{j_k}^i}{n_{-k,*}^i + \sum_{j'}^W \beta_{j'}^i}) (\frac{n_{-k,i}^{d_k} + \alpha_i^c}{n_{-k,*}^{d_k} + \sum_{i'}^{N_T} \alpha_{i'}^c}). \quad (8)$$

Here all n_{-k} are the counting of words of topic assignment or message except the current word token. j_k is the word index of the k^{th} word token in vocabulary and d_k is the document index of the k^{th} token.

With samples generated from Gibbs sampling, parameter ϕ_i and θ_i for topic i could be estimated as

$$\hat{\phi}_{ij} = \frac{n_j^i}{n_*^i + \sum_{j'}^W \beta_{j'}^i}, \quad (9)$$

$$\hat{\theta}_{ij} = \frac{n_i^d + \alpha_i^c}{n_*^d + \sum_{i'}^{N_T} \alpha_{i'}^c}. \quad (10)$$

As for hyperparamter α_c, there is an efficient estimation from [9], giving

$$\hat{a}_c = \frac{1}{N_{tc}^{\ 2}} \sum_{i=1}^{N_{tc}} \frac{E\{\vartheta_i\} - E\{\vartheta_i^2\}}{E\{\vartheta_i^2\} - E\{\vartheta_i\}^2}, \quad (11)$$

where $E\{\vartheta_i^p\} = \frac{1}{M_c} \sum_{d=1}^{M_c} \left(\frac{n_i^d}{\sum_{i=1}^{N_{tc}} n_i^d}\right)^p$. (12)

All the messages involved should have the same label c. Here M_c is number of messages labeled as c. If labels of all messages are omitted, a global α estimator yields without class preference.

The probability of message d labeled as $c = i$ given the words and label associated hyperparameter $\alpha_{c=i}$ is approximated as

$$p(c = i \mid z, w, \alpha_{c=i}) = \frac{\sum_{j \in d} l(z_j \in t_{c=i} \mid \alpha_{c=i})}{d_n}$$

where d_n is the number of words in message d and $l(z_j \in t_{c=i} \mid \alpha_{c=i})$ is the denoting function of the Gibbs sampling value of z_j under $\alpha_{c=i}$, belonging to the topics group $t_{c=i}$ associated with label $c = i$.

3. Classification

We have described the constrained model in section 2.3. The problem now is how to build up the classifier for the ownership problem.

We now present a pair of co-classifiers one of which focused on the class of "owned" and the other "not owned". With the constrained LDA discussed before, we could train the collection of messages as mixture of the 3 classes, and the hyperparameter α_c for each class and global α will be estimated and hidden variable β_i, θ_i will be estimated for each topic.

With α_{-1}, α_1 and α, we could build up three classifiers denoted as c_{-1}, c_1, c_0. We add a variable s as the state of the classifier to determine which classifier to be used and trustful. For each m_i which denoted the time-ordered messages of the same author, s is initially set as 0 to use the c_0 to classify the first message. If the probability given by c_0 for class 0 ("not mentioned") is less than a threshold p_0, the first message is predicted as class 0 and $s = 0$ is until there is such a message w_i that the probability for class 0 given by c_0 is less than p_0 and the state of classifier s jump out of the 0. Once $p(c_0 = 0) < p_0$, s will not return to 0 and would switch between -1 and 1. After $s \in \{-1, 1\}$, s will kept or switch to the other depend on whether the probability of the other class is more than the respective threshold p_{-1} or p_1. And prediction the c_{-1} in class -1 or 0 dependent which probability given by c_{-1} is larger, and c_1 in class 1 or 0. The illustration is shown in Figure 3.

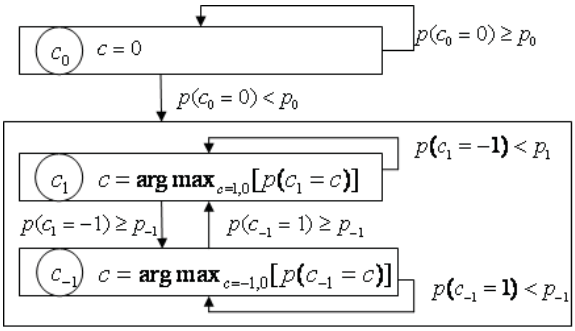


Figure 3: The process of the prediction by the 3 classifiers.

For good performance $\{\alpha_c\}$ is set to be the experienced value $50 / N_T$ to estimate the $\{\hat{\alpha}_c\}$. In the second time, elements of α_{-1} should be set to be zero except the one corresponding to the topics in T_{-1}, similar to α_0, α_1, so that estimated the global α.

In the training state, the main steps is as

(1) Set the hyperparameter α_c, β and number of topics for each class as N_{t-1}, N_{t0}, N_{t1}.

(2) Gibbs sampling and estimate α_c.

(3) Gibbs sampling to estimate global α regardless of label, and hidden variables φ_i, θ_i for each topics.

In the training stage, Gibbs sampling is label related so that the words is initialized to assign the topics associated to label of the message, shown by graphical model shown in Figure 4.

In the prediction state, the main steps are as follow.

(1) Set $\alpha_{-1}, \alpha_1, \alpha, \beta, \phi_{ij}$ equal to the ones estimated in training stage.

(2) For each m_i of author a_i iteratively predict the value of the ownership
(3) Initially set the state of classifier $s = 0$
(4) Use c_0 to classify the message d until $s <> 0$
(5) Use the c_{-1} or c_1 to classify d dependent on the state of classifier s
(6) If the condition of switching state of classifier is met, switch the state of classifier, go to (5).

Here the probability given by the classifier c^s is given by

$$p(c = k) \propto [\sum_{i \in T_k} (n_i^d + a_i^s)]. \quad (13)$$

4. Experimental result

For practical test of the model with real data, we prepare a dataset from an online forum in a popular digital camera website. The description of the data set is given later.

In the experiment, we take word sequences of 10 words around a camera model name excluding the model name, amount to 20 words. In the same message, all such words sequence related to the same model same would be merged together into a big word sequence. So a message may contain one or more camera model name with each corresponding to the unique words sequence. The data set totally contain words sequence of 58342 with 810 labeled but without consideration of time and authors. For testing the performance of the classifier on time attribute, messages of 14 authors have been labeled for test, where there are 10 camera models focused. In $m_i = \{d \mid author = i\}$, messages will be divided into several components associated to different camera model where the messages are ordered by time sequence. Each camera model component will be independently classified.

In this experiment, we take time attribute into consideration, so a new evaluation method is needed. The messages of the 14 authors are labeled by the continuous time period of the ownership. If message is assigned to class 0 until a message reveal the author owned or not owned the camera model. Once from a message, the author is found owned or not owned the camera model, its message after which should labeled as owned or owned until the ownership is found changed when the label should changed for the other one. For the prediction made by classifier, once a message is predicted as the owned or not owned class all message after that would be seen kept the same label until one message is classified as the opposite class. We used the LIBSVM [10] tool box for multi-class built on "one to against all" for comparison.

For 300 iterations of Gibbs sampler with α_c show in Table 1, β equal to 0.01 and 10 topics for each class, the constrained LDA could achieve 83.92% accuracy and LIBSVM 64.46%, since the prediction of SVM switch the state of the ownership too frequently. The constrained LDA show robust prediction in order of time.

The result is illustrated in Table 2, which show that the constrained LDA have high recall in the "owned" ant "not owned" class. The constrained not only detect the state of ownership effectively but also the time period of the ownership continuance.

Table 1: Setting of α_c for inference of global α in the left and of α_{-1}, α_1 in the right

Test	T_{-1}	T_0	T_1
α_{-1}	0.01, 5/3	0, 5/3	0, 5/3
α_0	0, 5/3	0.01, 5/3	0, 5/3
α_1	0, 5/3	0, 5/3	0.01, 5/3

Table 2: Classified result of constrained LDA and SVM. (a) lists the number of correct classified ones (b),(c) gives the confusion matrix given by both.

	Class -1	Class 0	Class 1	total
Ground truth	487	316	2008	2811
cLDA	350	161	1848	2359
SVM	292	85	1435	1812

(a) The number of messages classified correctly for constrained LDA and SVM

G \ P	Class -1	Class 0	Class 1
Class -1	0.718686	0.106776	0.174538
Class 0	0.294304	0.509494	0.196203
Class 1	0.066235	0.013446	0.920319

(b)Confusion matrix given by constrained LDA

G \ P	Class -1	Class 0	Class 1
Class -1	0.599589	0.034908	0.365503
Class 0	0.291139	0.268987	0.439873
Class 1	0.283865	0.001494	0.714641

(c) Confusion matrix given by the SVM

4.1. Evaluate performance on time sequence

Based on the fact that the camera ownership is usually stable for few months, one would not change its ownership from “owned” to “not owned” or vice verse frequently. Thereby classifier should have time continuous prediction, especially in capture of period of the state of ownership. The constrained LDA have good ability to predict the period of ownership in a robust manner. In semantically sense, it could correctly make prediction in the context of earlier messages of the author, and increase the accuracy in situation where there is not strong belief in any class. So compared with the SVM, the constrained LDA show good performance in switch times of ownership in a robust manner, which is shown in Figure 5.

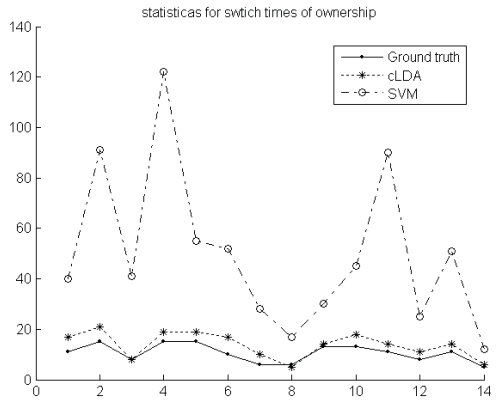


Figure 5 Statistics of switch times of ownership for the constrained LDA and SVM

4.2. Topical Feature

For each class value of ownership, there are 10 topics with 20 most associated words. From Figure 6, we could find that if one owned a camera, it may talk about advanced photographing in topic 2, imaging processing in topic 6, complaints or suggestion in topic 8. If one do not own, it involves having sold one’s camera in topic 11, willing to upgrade in topic 16 and comparison before purchasing. The topical feature will be help for further psychological analysis for the customers and business decision.

5. Conclusion and Future work

This paper proposes the problem for discovery of ownership of on-line forum and find the topical feature associated with the ownership.

The constrained LDA model proposed in this paper provides effective classifier of good predictive power for the problem, especially in the capture of the time period of the state of the ownership. Compared with the traditional classifier such as SVM, the constrained LDA could adaptive the prediction in the context of the front messages in the time sequence so that give a robust classification. In future work, N-gram method [5] should be introduced into the model since always the phrase is fitter to describe the discriminative feature of each class.

Topic 2		Topic 6		Topic 8	
Word	PROB.	Word	PROB.	Word	PROB.
eos	0.0072	images	0.012	complaints	0.017
kit	0.0066	oak	0.0086	underexposure	0.015
years	0.0056	bryan	0.0086	major	0.014
love	0.0055	material	0.0086	well	0.012
phounder	0.0052	grid	0.0086	opinion	0.0096
lens	0.0049	deal	0.0055	exposure	0.0091
plus	0.0046	depending	0.0043	worked	0.0087
my	0.0040	turned	0.0043	possible	0.0087
landscapes	0.0039	concerned	0.0043	wasn	0.0064
wildlife	0.0039	steps	0.0043	recommend	0.0048
Topic 11		Topic 16		Topic 18	
Word	PROB.	Word	PROB.	Word	PROB.
ago	0.011	year	0.016	know	0.023
two	0.011	upgrade	0.012	don	0.013
sold	0.0089	able	0.012	compare	0.0068
dead	0.0062	investing	0.0088	details	0.0046
amount	0.0062	holding	0.0058	spend	0.0046
months	0.005	next	0.0058	funds	0.0046
thx	0.0047	havent	0.0044	media	0.0034
expert	0.0047	arrival	0.0044	attracted	0.0034
budget	0.0047	sales	0.0044	happen	0.0034

Figure 6 some topics associated with the ownerships found by cLDA

References

[1] David Blei and Jon McAuliffe. Supervised topic models. In Yoram Singer John Platt, Daphne Koller and Sam Roweis, editors, Advances in Neural Information Processing Systems 21. MIT Press, 2008.

[2] Andrzejewski D., Mulhern, A., Liblit, B. Zhu J. Statistical Debugging using Latent Topic Models. ECML 2007.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In Proceedings of the 2004 ACM
SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM
Press New York, NY, USA, 2004.

[5] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In Proceedings of the 7th IEEE International Conference on Data Mining, pages 697--702, 2007.

[6] X. Wang and A. McCallum. Topics over time: A nonmarkov continuoustime model of topical trends. In

Knowledge Discovery and Data Mining (KDD), pages 424–433, 2006.
[7] I. B´ır´o, J. Szab´o, A. A. Bencz´ur. Latent Dirichlet Allocation in Web Spam Filtering. In Proc. 4th AIRWeb, 2008
[8] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In Proceedings of the Annual World Wide Web Conference (WWW), 2008.
[9] Gregor.Heinrich. Parameter estimation for text analysis. URL http://www.arbylon.net/publications/text-est.pdf
[10] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[11] Vijay Krishnan, Short comings of latent models in supervised settings, SIGIR 2005:625-626

Individualized Reaction Movements with Environments for Virtual Humans

Yuesheng He

Abstract

One of Virtual Humans' research objects aims to provide virtual characters with realistic behavior, which implies endowing them with autonomy in an inhabited virtual environment. Autonomous behavior consists in interacting with users or the environment and reacting to stimulus, events or different situations. This paper presents a method for the behavior of virtual human in the virtual environment to react individually. The virtual human is able to be controlled by users automatically. Individualization is achieved by endowing Virtual Human characteristics like personality, learning ability, etc. In this paper, we propose to use those individual descriptors to synthesize different kinds of reactions. We aim that individualized virtual humans react in a different way to the different stimuli. This approach is achieved by synthesize the framework of actions and learning ability with different parameters. Thanks to those action framework and machine learning method, we stereotyped reactive movements that can be described by individual characteristics. We use inverse kinematics techniques to synthesize the movements. This allows us to change reaction movements according to the characteristics of the stimuli and to the individuality of a character.

1 Introduction

Virtual humans, i.e. three-dimensional simulations of human beings, are being increasingly used in different domains. For example, they are used to explain physical and procedural human tasks , they are employed in military applications , in ergonomics , to simulate emergencies in first aid , and as virtual guides [2][3].

Proper development of virtual humans requires knowledge of different disciplines, such as computational geometry, kinematics, artificial intelligence, computer graphics, and bio-mechanics. The complexity of building virtual humans encourages to divide the problem in several subproblems; this can be done with the five levels hierarchy. The lowest layer of the modeling hierarchy is the geometric layer that concerns the definition of the virtual human model and its appearance[3][4].

At the kinematic layer, the virtual human is represented as a set of rigid bodies, called segments, hierarchically organized and connected by joints. From this point of view, an animation can be defined in two ways: by specifying joints rotations, or by defining (or automatically computing) positions of specific parts of the virtual human body (called end-effectors) in time. The latter approach uses inverse kinematics to compute the joints configuration (in terms of rotation values) needed to put end-effectors in particular positions; this approach is commonly used to control hands and feet movements. At the physical layer, the animation is obtained by applying physical laws to different parts of the virtual human body to compute complex animations, such as skin deformation or hair movement.

The behavioral layer represents the instinctive behavior of the virtual human (e.g. in terms of stimulus-action associations), while the highest layer, the cognitive one, binds various stimuli with reasoning processes that allow the virtual human to search for the most suitable action.

Cognitive models go beyond behavioral models in that they govern what the virtual human knows, how that knowledge is acquired, and how it can be used to plan actions. Providing the user with an intuitive interface to control the avatars motion is difficult because the characters motion is high dimensional and most of the available input devices are not. Input from devices such as mice and keyboard typically indicates a position (go to certain location), or behavior (jump or run). This input must then be supplemented with autonomous behaviors and transitions to compute the full motion of the avatar. Control of individual degrees of freedom is not possible for interactive environments unless the user can use his or her own body to act out or pantomime the motion. In this paper, we show that a rich, connected set of avatar behaviors can be created from extended, freeform sequences of motion, automatically organized for efficient search, and exploited for real-time avatar control using a variety of interface techniques. The motion is preprocessed to add variety and flexibility by creating connecting transitions where good matches in poses, velocities, and contact state of the character exist. The motion is then clustered into groups for efficient searching and for presentation in the interfaces. A unique aspect of our approach is that the original motion data and the generalization of that data are closely

linked; each frame of the original motion data is associated with a tree of clusters that captures the set of actions that can be performed by the avatar from that specific frame.

The learning ability allows us to take advantage of the power of clusters to generalize the motion data without losing the actual connectivity and detail that can be derived from that data. Thus, different layers structure can be efficiently searched at run time to find appropriate paths to behaviors and locations specified by the user. We explore the framework and learning ability of avatars provide the user with intuitive control of the avatars motion: crossing different terrains.

The motion control technique integrates animation, biomechanics, human gait experiments, and psychology, to represents an important initial step toward meeting the locomotion requirements in diverse environments.

First, any strategy level plan for the virtual human must be on the base of the basic action level motion control and can support any optimization approaches which have possibility to be integrated into the motion control mechanism to simulate motions in different environments.

Second, the method should give virtual human learning ability to adapt different virtual environment. Finally, it is responsive. Since relatively simple inverse kinematics mechanisms and optimal search algorithms are widely used in the computation, interactivity can be easily achieved, which would make the method well suited for virtual environment applications. To achieve the requirement, a statistics learning method is presented in the paper. The second section will give the description of it. The Third section will give some experimental result. Then, the fourth section will give a conclusion and discussion .

2 Interaction with the environment

To build an intelligent individual virtual human to interact with the digital virtual environment is the main goal of our research. To achieve the goal, the approach concentrates on: first to built a virtual human software environment to support the research. Second, construct the framework of virtual human's motion [1]. Third, figure out the individual behavior of virtual human, especially on the Statistics Learning.

Thus, in order to reach the effect we expect, the virtual human should have many basic properties. Moreover, this method depends on both software framework and machine learning. Then the research of work includes aspects below[5][9][10]:

1. Constructing the framework of virtual human's behaviors. On this topic, there are main three aspects: Interactive Control, Synthesizing Animations, and main framework of the computational model.

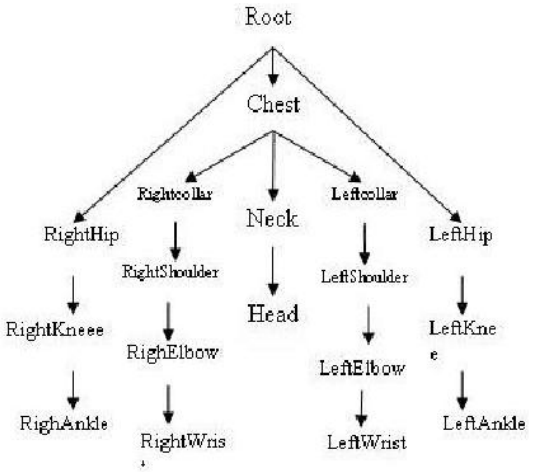


Figure 1. The tree structure of virtual human body

2. The learning ability based on machine learning that concentrates on the Statistics Learning (mainly on the SVM).

3. Individuality: describes parameters that can define the individuality of a person, like personality, emotional state, cultural background, etc.

4. Behavior Controllers: they are algorithms used to produce behavioral animation. The class specifies the inputs required for the algorithm to work and the outputs (usually animation sequences or specific joint values) it is capable to produce.

5. Reaction behavior: algorithm that simulates the evaluation process where, considering the virtual human attributes and the nature of a stimulus, it will generate a kind of reaction.

To describe a virtual human's body model in the virtual environment being constructed by the computer, it should be easy to be controlled. First of all, it should be convenient to simulate real human's action in the real environment. We define the virtual human's model structure as the figure 1.

When the Virtual Human's body is represented by a proper structure, the animation is going to be made. However, in the motion process, the Virtual Human's actions are based on the way that has been used to describe the elemental actions and behaviors that have been planned by the strategy layer.

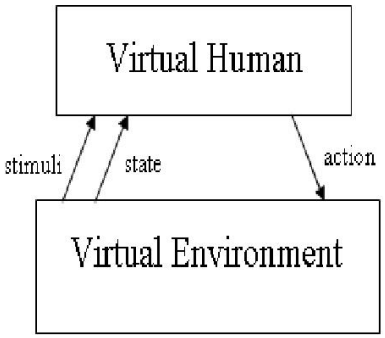


Figure 2. Relationship between Virtual Human and Virtual Environment

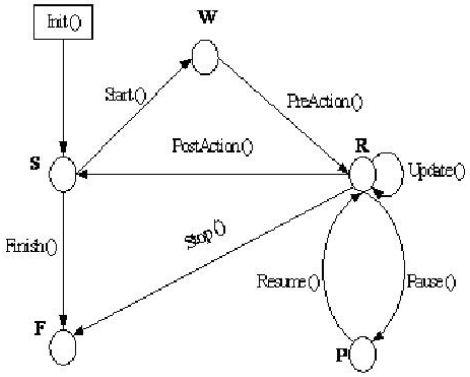


Figure 3. Virtual Human's action model

2.1 Action Framework

The relationship between Virtual Human and virtual environment is as the figure 2.

The characteristic of the action is a trial-and error feature. The reward will be given when the answer to a question is correct, while the penalty will be awarded when there is error. There are three elements involve in reinforcement learning of virtual human in the virtual environment, they are:

- environment state
- records of the information which impacts an action
- action which is performed to direct to the state

As the movement and postures should be used by the planning level, the low level is to solve the problem of how to describe the elemental action of a virtual human. So, it is described a finite automata as Figure 3[1].

The 5 different States are[1]:

1. S – Stopped : accomplished the initialization
2. W – Waiting: ready to start the simulating loop
3. R – Running: running the simulating loop
4. P – Paused: simulating loop is paused
5. F – Finished: the action is finished

Then the elemental motion of Virtual Human should be described by a regular language.

Based on the regular language, the actions are defined individualized constraints depending on the reaction to send to inverse kinematics library. In the intersection reactive movement, we compute the vector position for the end effectors of each hand. This position is the position of the stimuli at the moment when it is reachable by the virtual human arms, foot or legs plus the separation of them according to the type of the stimuli.

2.2 Learning Process

To endow Virtual Human the ability to classify different kinds of environments(e.g. the terrain), we used the statistics learning in this method.

The basic statistics learning task is to estimate a classification function $f : R^N \rightarrow \{\pm 1\}$ using input-output training data from two classes[6]:

$$(x_1, x_1) \ldots (x_l, x_l) \in R_n \times \{0, 1\}$$

The function f should correctly classify unseen examples (x,y), i.e. f(x) = y if (x,y) is generated from the same underlying probability distribution as the training data. In this work we limit discussion to linear classification functions. We will discuss extensions to the nonlinear case in Section 5. If the points are linearly separable, then there exist an n-vector w and scalar b such that

$$Y_i[x_i \cdot w - b] \geq 0$$

The "optimal" separating plane, w= b, is the one which is furthest from the closest points in the two classes. Geometrically this is equivalent to maximizing the separation margin or distance between the two parallel planes $w \cdot x = b+1$ and $w \cdot x = b-1$ In general the classes will not be separable, so the generalized optimal plane (GOP) problem is used. A slack term η_i is added for each point such that if the point is

misclassified, $\eta_i \geq 1$. The final GOP formulation is:

$$\min_{wb\eta} C \sum_{i=1}^{l} \eta_i + \frac{1}{2}\|w\|^2 \quad \eta_i \geq 0 \quad i = 1, \cdots, l$$

Because of the complexity of the virtual environment the Virtual Human should classify, the input data often cannot be separated by a linear function. In this case the SVM which maps the input data into a feature space to separate them linearly is a fitful method[18]. The figure 4 shows the SVM on gaussian kernel to classify the data.

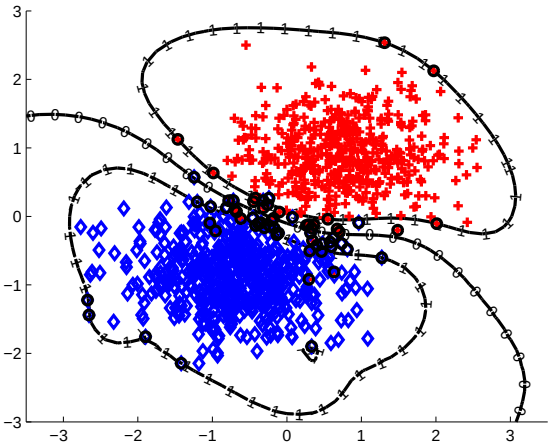


Figure 4. Gaussian Kernel

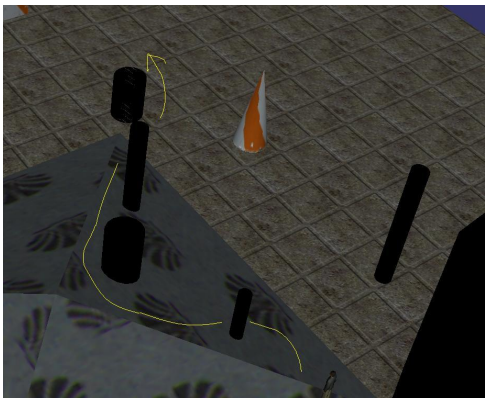
3 Experimental Result

We test our behavioral control method of Virtual Human by implementing the method in C++ and Python thanks to the open source code "ReplicantBody" on the virtual human simulations.

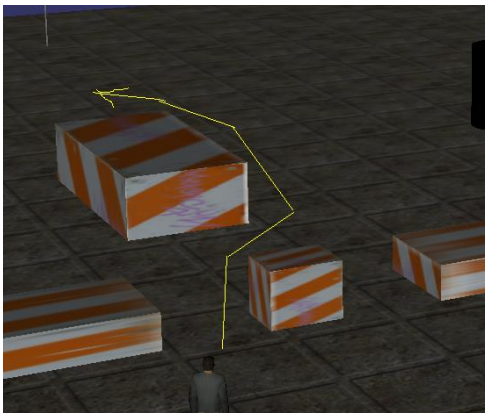
ReplicantBody is a character animation toolkit written in C++, built upon Cal3D, ConfigScript and OpenSceneGraph Features. Copyright (C) 2003 VRlab, Ume University. It is distributed under GNU LGPL.

Walking movement of Virtual Human, is function of the direction and shape of obstacles of the stimuli. We compute the direction vector of the stimuli to get the direction of the movement which consists the path to go. The classification of strategies is the function of training data of the size and shape of the obstacles. This movement is applied to the spine end-effector for collision detection. We define a new position to the arms and legs end-effectors with a lower level of priority and recruiting.

When the obstacles' shapes are mainly circular or rectangular, the path should be in different shapes as the figure 5 shows.



(a) Circular obstacles



(b) Rectangle obstacles

Figure 5. Different kind of path in the different scales of obstacles (a) and (b).

After training the strategy by the data of the terrains on the statistics learning, the Virtual Human will use the different strategy to pass through them.

As we can see, the method can control the virtual human to walk naturally and plan an optimal path to cross 3 dimensional obstacles to reach the goal. Moreover, when the system of animation is running, the frame rate is more than 50 frames per second as figure 6. That means it has ability to support real time animation.

(a) Walk in the circular obstacles

(b) Walk in the rectangle obstacles

Figure 6. Virtual Human walk in the different scales of obstacles The (a) and (b).

4 Conclusion and Discussion

We have developed an method for animating whole-body motions for virtual human characters that relies on constrained inverse kinematics and path planning to endow them some individual motions. The method synthesizes higher level planning strategy for different actions while respecting both environment and posture constraints. The work required to generate a sequence of motion, in addition to running the planning program, is first set the start and goal position and orientation of the object, and then set the initial configuration of the character. The time for this procedure would be in the order of a few minutes with the interface of typical 3D open source virtual human software packages. The total of man hours and CPU time is therefore acceptable, while even an expert artist would likely take an hour to keyframe similar motions.

One advantage of our method is that the data need not be a perfect match for the task. Because the process is model-based, the knowledge about human motion inherent in those models allows the search process of the planner and inverse kinematics algorithm to fill in some gaps.

The shape of the object is taken into account during collision detection and via the users specification of the grasp points. If however, human strategies for going through different terrains objects are fundamentally different, those strategies will appear only if the motion database includes sequences in which the actor used that strategy. A more detailed biomechanical model could also be used to create this particular effect by taking into account the forces in the back during a lifting task.

Extending an arm for balance is a solution that will not be found unless such a pose is included in the database. Basically the strategy used for the task should be represented in the database in some way for the system to succeed. Additional data with different characteristics can also increase the variety of motions if they are individual on different learning abilities.

We do not yet have a way of measuring when the data is appropriate to the task or how much data is sufficient to produce natural looking motion for a given task. We have also observed situations in which a very small and dissimilar data set degraded the quality of the motion by providing inappropriate examples.

Because of the method has been integrated with the low level behavioral model of the avatar well, it can well support the natural behavior of the virtual human such as walk and run.

Depends on the method the learning process does not always need the supervisor signal, but only the reward from the environment, the virtual human gets the ability to adapt different circumstance.

Also because the framework of the method has the ability to fit the levels structure of the description of Virtual Human's behavioral model. The virtual human under this method is responsive, realistic and easy to control.

We have shown that, once the kind of behavior is selected, we use the information of the stimuli parameters to compute the basic pose that the character may take. After that, we use characters individual traits and properties of the stimuli from the environment to set different parameters to the inputs of inverse kinematics to vary movements.

We have presented how we synthesize different kinds of reactive movements according to individual character descriptors. Based on an different parameter or kernel of SVM, we have kinds of reactive movements. To synthesize the movements identified, we use the strategy to control the lower level motion of elemental actions. one to be able to

reproduce them using Inverse Kinematics.

We want to define how we can mix the types of reactions reaction found. We will also establish the main inputs to the inverse kinematics in reactive movements. Moreover, we want to test different kinds of stimuli in a more complete and complicated scenario, like a house. On the other hand, though the method has ability to compute dynamic states, the computational complication is still high. Thus multi virtual humans in a same environment, the requirement of computational resources will be even higher.

Our future work is to simplify the model of the framework to make it to support much more complicated environment. Otherwise, we are going to analyze the multi virtual human's behavior more deeply to find out more efficient learning algorithm to control their actions.

References

[1] Yuesheng. He, Yuan-Yan. Tang, Path Planning of Virtual Human by Reinforcement Learning, ICMLC 2008 Proceedings, pp.987-992, 2008

[2] T. Conde, D. Thalmann, An Integrated Perception for Autonomous Virtual Agents: Active and Predictive Perception, Computer Animation and Virtual Worlds, Volume 17, Issue 3-4, John Wiley, 2006

[3] Moccozet L, Thalmann N. M., Dirichlet Free-Form Deformation and their Application to Hand Simulation[J], Proceedings Computer Animation97, IEEE Computer Society, 1997, pp.93-102.

[4] Molet T, Boulic R, Thalmann D. A real-time anatomical converter for human motion capture[J]. In Euro graphics Workshop on Computer Animation and Simulation, 1996, pp.79-94.

[5] N. Magnenat-Thalmann, A. Foni, G. Papagiannakis, N. Cadi-Yazli. Real Time Animation and Illumination in Ancient Roman Sites. The International Journal of Virtual Reality, IPI Press, Vol.6, No.1, pp.11-24. March 2007.

[6] Istv an Szita, B alint Tak acs deim, Andr as Lorincz, C MDPs: Learning in Varying Environments. Journal of Machine Learning Research pp.145-174 3 (2002)

[7] Theodore J. Perkins PERKINS, Andrew G. Barto BARTO, Lyapunov Design for Safe Reinforcement Learning, Journal of Machine Learning Research pp.803-832 3 (2002)

[8] Ronan Boulic, Pascal BCcheiraz, Luc Emering, Daniel Thalmann, Integration of Motion Control Techniques for Virtual Human and Avatar Real-Time Animation, pp.111-117, ACM VRST 97

[9] Weixiong Zhang, Randall W. Hill, Jr., A Template-Based and Pattern-Driven Approach to Situation Awareness and Assessment in Virtual Humans, pp.116-123, Agents 2000, ACM 2000 1-58113-230-1/00/6.

[10] Z.M. , D. Reidsma, A. Nijholt, Human Computing, Virtual Humans and Artificial Imperfection, ICMI'06, pp.179-184, ACM, 1-59593-541-X/06/0011.

[11] Karl Tuyls, Katja Verbeeck, Tom Lenaerts, A Selection-Mutation Model for Q-learning in Multi-Agent Systems, AAMAS'03, pp.693-700, ACM, 1-58113-683-8/03/0007

[12] Catherine Zanbaka1, Amy Ulinski, Paula Goolkasian, Larry F. Hodges, Social Responses to Virtual Humans: Implications for Future Interface Design, CHI 2007 Proceedings, pp.1561-1570, ACM 978-1-59593-593-9/07/0004.

[13] Edward M. Sims, Reusable, lifelike virtual humans for mentoring and role-playing, Computers Education pp.75-92, 49 (2007).

[14] Lucio Ieronutti , Luca Chittaro, Employing virtual humans for education and training in X3D/VRML worlds, Computers Education pp.93-109, 49 (2007).

[15] Joseph Laszlo, Michiel van de Panne, Eugene Fiume, Interactive Control For Physically-Based Animation, Department of Computer Science University of Toronto. Okan Arikan, D.A.Forsyth, Interactive Motion Generation from Examples, 2002 ACM 1-58113-521-1/02/0007, pp.483-490.

[16] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Interactive Control of Avatars Animated with Human Motion Data, 2002 ACM 1-58113-521-1/02/0007, pp.491-500.

[17] Katsu Yamane, James J. Kuffner, Jessica K. Hodgins, Synthesizing Animations of Human Manipulation Tasks, 2004 ACM 0730-0301/04/0800-0532, pp.532-539.

[18] Alain Rakotomamonjy,Variable Selection Using SVM-based Criteria,Journal of Machine Learning Research 3, (2003) pp.1357-1370

[19] Carlos Diuk, Alexander L. Strehl Michael L. Littman, A Hierarchical Approach to Efficient Reinforcement Learning in Deterministic Domains, AAMAS'06 May 8-12 2006, Hakodate, Hokkaido, Japan.

Empirical Mode Decomposition Applied in Face Recognition

Dan Zhang

Abstract

Two Empirical Mode Decomposition (EMD) based face recognition schemes are proposed in this paper to address variant illumination problem. Empirical Mode Decomposition is a data-driven analysis tool for nonlinear and non-stationary signals. It decomposes signals into a set of Intrinsic Mode Functions (IMFs), which capture different features of the original signal from high frequency to low frequency. It only relies heavily on the original signals. The IMFs are able to capture more representative features of the original signals, especially more singular information in high-frequency ones. This accords well with the requirements of face recognition under variant illuminations. Earlier studies show that only the low-frequency component is sensitive to illumination changes, which indicates that the corresponding high-frequency components are more robust to the illumination changes. Therefore, two face recognition schemes based on the IMFs obtained by EMD are generated. One is using the high-frequency IMFs directly for classification. The other one is based on the synthesized face images fused by high-frequency IMFs. The experimental results on the PIE database verify the efficiency of the proposed two schemes.

1 Introduction

Empirical mode decomposition (EMD) is an adaptive signal analysis method for nonlinear and non-stationary data. It has been originally introduced by Huang et al in [1], which is the first step of Hilbert-Huang transform [2, 3]. Essentially EMD decomposes an input signal into a set of Intrinsic Mode Functions (IMFs), which recover the original input signal features in a multiscale sense from high frequency to low frequency. The IMFs heavily rely on the original signal rather than using predetermined filter or wavelet functions. Due to the non-parametric and data-driven advantages, EMD has been successfully applied in one dimensional signal analysis such as ocean waves, rogue water waves, sound analysis, and earthquake time records. Recently it has been extended to the analysis of image data. The first track can be retrospected to [4], in which S. Long et al applied EMD to analyze digital slope images. The image data has been expressed in terms of an array of rows and columns, the EMD is applied to these arrays row by row. Linderhed et al also adopted this row-by-row EMD method in the fusion of visible and infrared images [5, 6]. The input images are vectorized in lexicographical order and EMD is performed on each channel vector separately. R. Bhagavatula et al addressed face recognition in the same way in [7].

The EMD has been extended for two dimensional data in the literature toward texture extraction [8] and image compression [10]. The two dimensional EMD (2DEMD) has a more wide application in various kinds of image analysis, such as image fusion [12], image compression [11], texture analysis [9], feature extraction [14], rainfall analysis [13], watermarking [15], and temperature variation [16].

Faces as two dimensional digital signal are generally processed by wavelets or other filters. However, these methods are not adaptive to the original faces and always affected by the predetermined functions. The captured facial features sometimes are anamorphic which always leads to low recognition rate. Though EMD based methods can capture the intrinsic feature of the iamges, few literatures consider face recognition. Face recognition refers to technologies for human authentication or verification based on human facial characteristics. Though this biometric technique has been improved a lot, however, many present facial biometric systems are still confused when identifying the same person with varying illumination. Earlier studies [17, 18, 19] show that only the low-frequency component is sensitive to illumination changes. It indicates that the corresponding high-frequency components are more robust to the illumination changes. Therefore, the high-frequency components alone without the low-frequency components are sufficient enough for classification. Moreover, due to the adaptiveness of the EMD based methods, the corresponding IMFs are able to capture more representative features of the original signals, especially more singular information in high-frequency ones. Therefore, two available face recognition schemes based on EMD are generated. One is using the high-frequency IMFs directly for classification. The other one is based on the synthesized face images fused by high-frequency IMFs. The proposed two methods efficiently address the illumination variance problem. The experimental

results on the PIE database verify their efficiency.

The paper is organized as follows. Section 2 presents a review of the EMD and extended 2DEMD sifting process. Section 3 presents the flow of the proposed two face recognition schemes. The experiments are demonstrated in Section 4. Finally, a conclusion and future work outlook are presented in Section 5.

2 Empirical Mode Decomposition and application in images

EMD is a signal analysis technique for adaptive representation of non-stationary signals as sum of a set of IMFs. It captures information about local trends in the signal by measuring oscillations, which can be quantized by a local high frequency or a local low frequency, corresponding to finest detail and coarsest content. Here we briefly review the sifting process of EMD and its implementation. Given a series signal x(t), four main steps are contained. S1, S2, S3 and S4 are abbreviation for Step 1 to Step 4.

S1. Identify all local minima and maxima of the input signals $x(t)$;

S2. Interpolate between all minima and maxima to yield two corresponding envelopes $E_{max}(t)$ and $E_{min}(t)$. Calculate the mean envelope $m(t) = (E_{max}(t) + E_{min}(t))/2$;

S3. Compute the residue $h(t) = x(t) - m(t)$. If it less than the defined threshold then it becomes the first IMF, go to Step 4. Otherwise, repeat Step 1 and Step 2 using the residue $h(t)$, until the latest residue meets the threshold and turns to be an IMF;

S4. Input the residue r(t) to the loop from Step 1 to Step 3 to get the second and more IMFs until it can not be decomposed further.

There are three keys determine the implementation of the EMD.

- Judge whether an residue is an IMF or not. Theoretically, the judgement should accord two conditions [1]. First, the number of extrema and the number of zero-crossing must be at most differ by one. Second, the mean envelops obtained by the maximum envelop and the minimum envelop must equal to zero. In the realistic implementation, researchers generally define a threshold. If $h(t)$ is small enough then it is an IMF.

- Choose an appropriate interpolation functions, such as linear interpolation, cubic spline interpolation and so on.

- Stop criterion. How to judge the residue can be decomposed further or not. Generally, the decomposition stops when the residue becomes so small that it is less than the predetermined threshold.

The EMD is originally proposed for one dimensional data. However, it has also been widely applied in image analysis. The general idea is to express images as an array of one dimensional vectors. There are two ways. One is to view the image as a long lengthened vector and then apply EMD on it. Another is apply on rows or columns as literatures [4, 5, 6, 7] does. In order to discriminant these two methods, we name the former one as 1DEMDa and the latter one as 1DEMDb. Fig.7 and Fig.2 shows all the decomposed IMFs and residues obtained by these two methods. The IMFs captures the information of the original face from the finest to the coarsest. Moreover, the illumination effects are distributed to just several low level IMFs and residues. By removing these low frequency IMFs, the remained high frequency IMF is not so sensitive to the illumination effects any more and themselves alone are sufficient enough for classification.

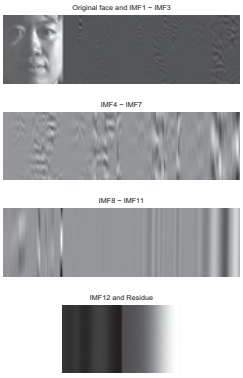


Figure 1. Twelve IMFs and the residue decomposed by 1DEMDa.

The extended 2DEMD has the similar sifting process with 1DEMD. Given the digital image signal $I = f(x, y)x = 1, ..., M, y = 1, ..., N$, the process is summarized as follows:

S1. Identify the extrema (both maxima and minima) of the image $I = f(x, y)$;

S2. Generate the 2D envelope by connecting maxima points (respectively, minima points) by surface interpolation. Determine the local mean m by averaging the two envelopes;

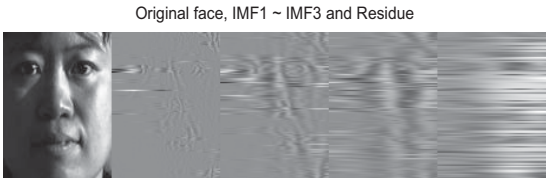


Figure 2. Three IMFs and the residue decomposed by 1DEMDb.

S3. Subtract out the mean from the image to get an residue $h = I - m$, judge whether h is an IMF, if it is, go to Step 4. Otherwise, repeat Step 1 and Step 2 using the residue h, until the latest residue turns to be an IMF;

S4. Input the residue h to the loop from Step 1 to Step 3 to get the second and more IMFs until it can not be decomposed further.

In the implementation of two dimensional case, researchers generally via Stop Criterion (SD) to judge whether residue h is an IMF and when to stop the loops: $SD = \sum_{k=0}^{m} \sum_{i=0}^{n} [\frac{|h_{i(j-1)}(k,l) - h_{ij}(k,l)|^2}{h_{i(j-1)}(k,l)^2}]$. Different with 1DEMD, extrema detection and surface interpolation seems more complicate. It is important to choose the appropriate extrema detection method and surface interpolation functions, otherwise, it will affect the decomposition results. You may see [13] for more details. Fig.3 demonstrated the decomposition results by 2DEMD. The IMFs are still from finest to coarsest and the illumination effects focus on the residue.

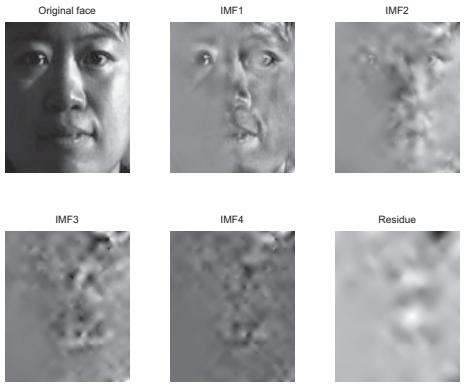


Figure 3. Four IMFs and residue decomposed by 2DEMD on a face image from the PIE database that with right point illumination.

3 Face recognition schemes

As described in the former sections, two face recognition schemes based on EMD are generated. One is using the high-frequency IMFs directly for classification. The other one is based on the synthesized face images fused by high frequency IMFs. Fig.4 and Fig.5 show the flowchart of two methods respectively. In our implementation, the following four keys should be explained in detail.

- Down-sampling is adopted twice before the EMD process which is in order to decrease the dimension of the faces, otherwise, the computational time is huge.
- We choose the 1st IMFs for representing facial features since the 1st IMF is the highest frequency component.
- Fusion the 1st IMFs of the train faces and test faces. In our experiment, we adopt only one neutral illumination face for train and aim to verify the other four faces

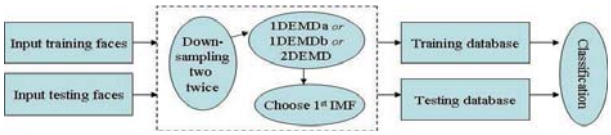


Figure 4. Flowchart of frequency face recognition scheme.

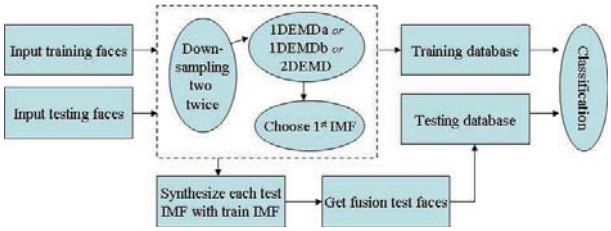


Figure 5. Flowchart of fusion based face recognition scheme.

with serious illumination effects. As described in former sections, illumination effects always exist in low frequency IMFs, i.e., the 1st IMF contains minimum illumination effects. Therefore the synthesized faces of the two 1st IMFs enhance their mutual features. On behalf of the test faces, the serious illumination effects are decreased a lot.

4 Experimental Results

We evaluated the proposed two schemes using the PIE face database which is accessible at http://www.ri.cmu.edu/projects/project_418.html. This database contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. Here, we only focused on the images varying with illumination. Fig.6 showed 21 face samples of one person. All the face images were cropped into 112×92 size.

Figure 6. Face image samples from the PIE database

In our experiments, we have established two sub-database for two schemes using independently. The first one is used for testing the frequency method. As the rectangles denoted in Fig.6, the first 6 images were chosen from each subject. Thereinto 3 face images were used for training and remained 3 images for testing. There are 20 combination cases, here we only evaluated 5 cases and the results listed in Table.1 are average ones. Another sub-database contains 5 faces per person. Each face is quite different from the other 4 in terms of illumination effects. We used the neutral one for training and the remained 4 for testing. Fig.7 showed an example.

Additionally, we have also compared our EMD based methods with other traditional face recognition methods such as wavelets. In order to guarantee equitable comparisons, Daubechies wavelet 'db4' is adopted for separable wavelet decomposition, since 'db4' generally got the best performances as reported in [20, 21]. The non-separable wavelet adopted here is constructed in our earlier work [22], the parameters were set as $\alpha = 0.1, \beta = 0.9, N = 1$ since the best performances were got at these points. Here we adopted Support Vector Machine (SVM) as classifier. Our experiments were implemented in a personal computer with Genuine Inete(R)T2300 CPU and 1.5G RAM and Matlab version 7.0 was used.

Table 1. Recognition rate and executing time versus different methods by scheme 1.

Methods	Average Correctness	CPU Time
1DEMDa	99.51% / 203	143.39
1DEMDb	97.55% / 199	496.71
2DEMD	93.63% / 191	112.25
Separable wavelet	78.92% / 161	2.20
Nonseparable wavelet	89.71% / 183	1.98

It is easy to see that the EMD based methods all get higher correctness than the wavelet based ones. Especially 1DEMD applying on the lengthened image vector performs best, and the following is 1DEMD applying row by row, however it cost the most time.

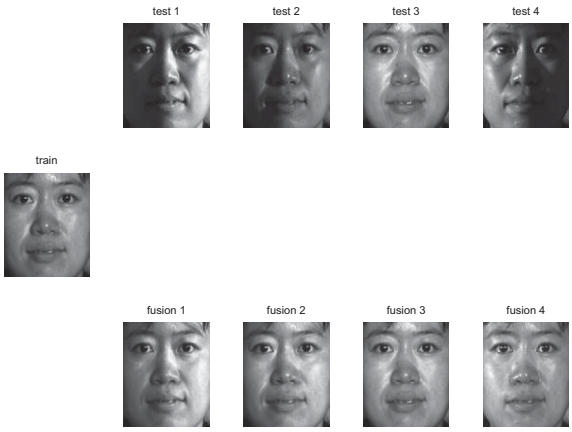


Figure 7. Original testing face images and fusion face images by 1DEMDa.

The next experiment we focused on testing the scheme

2. Fig.7 demonstrates some face samples from the database and simultaneously shows the corresponding test fusion images. It is obvious that the illumination on test images have been removed a lot. Table.2 shows the performances before fusion. Poor correctness are got. Table.3 demonstrated the results after fusion. The recognition rate are improved significantly, and all the EMD based methods performs perfectly. Moreover, the results indicate again that EMD is more efficient than the wavelets in face recognition.

Table 2. Recognition rate and executing time versus different methods before fusion.

Methods	Correctness	CPU time
1DEMDa	84.93% / 231	135.28
1DEMDb	32.73% / 89	440.48
2DEMD	63.24% / 172	80.81
Separable wavelet	74.24% / 202	10.25
Nonseparable wavelet	82.72% / 225	2.42

Table 3. Recognition rate and executing time versus different methods after fusion.

Methods	Correctness	CPU time
1DEMDa	100% / 272	124.65
1DEMDb	100% / 272	438.84
2DEMD	100% / 272	78.64
Separable wavelet	97.06% / 264	1.42
Nonseparable wavelet	100% / 272	1.57

5 Conclusions

We have proposed two EMD based face recognition schemes in this paper. Both of the schemes have efficiently addressed the illumination effects problem. Compared with the traditional wavelet based methods, the proposed EMD based methods have much advantages in recognition rate. However, the huge differences in executing time reminds us to find improved fast algorithms in future. Otherwise, it is expensive to apply EMD processing a mass of images. Additionally, compared with 1DEMD, 2DEMD performed faster but poorer recognition rate. This is properly affected by the choices of surface interpolation methods. Finding the appropriate surface interpolation functions to improve the 2DEMD's performance is our remaining work in the coming days as well.

References

[1] N. E. Huang, Z. Shen, S. R. Long, et al.. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A, vol. 454, no. 1971, pp. 903C995, 1998.*

[2] HILBERT-HUANG TRANSFORM AND ITS APPLICATIONS. *Book in Interdisciplinary Mathematical Sciences, Vol. 5, edited by N. E. Huang, and Samuel S P Shen, 2005.*

[3] N. E. Huang, M. L. C. Wu, S. R. Long, et al.. A confidence limit for the empirical mode decomposition and Hilbert spectral analysis. *Proceedings of the Royal Society A, vol. 459, no. 2037, pp. 2317C2345, 2003.*

[4] S. R. Long. Applications of HHT in image analysis. in Hilbert-Huang Transform and Its Applications, N. E. Huang and S. S. P. Shen, Eds., World Scientific, River Edge, NJ, USA, 2005.

[5] Harishwaran Hariharan, Andrei Gribok, Besma Abidi, and Mongi Abidi. Multi-modal Face Image Fusion using Empirical Mode Decomposition. *The Biometrics Consortium Conference, Crystal City, VA, September 2005.*

[6] H. Hariharan, A. Koschan, B. Abidi, A. Gribok, and M.A. Abidi. Fusion of visible and infrared images using empirical mode decomposition to improve face recognition. *IEEE International Conference on Image Processing ICIP2006, Atlanta, GA, pp. 2049-2052, October 2006.*

[7] Bhagavatula, R., Marios Savvides, and M. Acoustics. Analyzing Facial Images using Empirical Mode Decomposition for Illumination Artifact Removal and Improved Face Recognition. *IEEE International Conference on Speech and Signal Processing, 2007 (ICASSP 2007). Vol. 1, Issue , 15-20 April 2007 pp. I 505-508.*

[8] J. C. Nunes, Y. Bouaoune, E. Delechelle, O. Niang, and Ph. Bunel. Image analysis by bidimensional empirical mode decomposition. *Image and Vision Computing Volume 21, Issue 12, Pages 1019-1026, November 2003.*

[9] Nunes J. C., Guyot S., and Deléchelle E. Texture analysis based on local analysis of the Bidimensional Empirical Mode Decomposition. In *Machine Vision and Applications 16, 3, pp. 0932-8092, 2005.*

[10] A. Linderhed. 2-D empirical mode decompositions in the spirit of image compression. *in Wavelet and Independent Component Analysis Applications IX, vol. 4738*

of Proceedings of SPIE, pp. 1C8, Orlando, Fla, USA, April 2002.

[11] A. Linderhed. Compression by image empirical mode decomposition. *IEEE International Conference on Image Processing, 2005 (ICIP 2005), Vol. 1, pp. I 553-6, 2005.*

[12] H. Hariharan, A. Gribok, M. Abidi, and A. Koschan. Image Fusion and Enhancement via Empirical Mode Decomposition. *Journal of Pattern Recognition Research, Vol. 1, No. 1, pp. 16-32, January 2006.*

[13] Sinclair, S. and Pegram, G. G. S. Empirical Mode Decomposition in 2-D space and time: a tool for space-time rainfall analysis and nowcasting. *Hydrol. Earth Syst. Sci. Discuss., 2, 289-318, 2005.*

[14] Jian Wan, Longtao Ren, and Chunhui Zhao. Image Feature Extraction Based on the Two-Dimensional Empirical Mode Decomposition. *2008 Congress on Image and Signal Processing, Vol. 1, pp. 627-631, 2008*.

[15] Jalil Taghia, Mohammad Ali Doostari and Jalal Taghia. An Image Watermarking Method Based on Bidimensional Empirical Mode Decomposition. *2008 Congress on Image and Signal Processing, Vol. 5, pp. 674-678, 2008.*

[16] Fauchereau, N., Sinclair, S., and Pegram, G. 2-D Empirical Mode Decomposition on the sphere, application to the spatial scales of surface temperature variations. *Hydrol. Earth Syst. Sci. Discuss., 5, 405-435, 2008.*

[17] C Nastar. The image shape spectrum for image retrieval. *Technical report, INRIA, No. 3206, June 1997.*

[18] C Nastar, B Moghaddam and A Pentland. Flexible images: matching and recognition using learned deformations. *Computer Vision and Image Understanding, Vol. 65, No. 2, pp. 179- 191, 1997.*

[19] Zhang, Z. B., Ma, S. L. and Wu, D. Y. The application of neural network and wavelet in human face illumination compensation. *Proc. Advances in Neural Networks, pp. 828C 835, 2005.*

[20] Feng, G. C., Yuen, P. C. and Dai, D. Q. Human face recognition using PCA on wavelet subband. *Journal of Electronic Imaging, Vol. 9, No. 2, pp. 226C233, 2000.*

[21] Ekenel, H. K. and Sanker, B. Multiresolution face recognition. *Image and Vision Computing, Vol. 23, pp. 469C477, 2005.*

[22] Xinge You, Dan Zhang, Qiuhui Chen, Patrick Wang, and Yuan Yan Tang. Face representation by using non-tensor product wavelets. *18th International Conference on Pattern Recognition, pp.503–506, Aug 2006.*

A Novel Motion Based Lip Feature Extraction for Lip-reading

Meng Li

Abstract

In a lip-reading system, one key issue is how to extract the visual features, which greatly impact on the lip-reading recognition accuracy and efficiency. In this paper, we propose a novel motion based visual feature representation. Compared with the existing methods, our approach focuses on the crucial part of lip movement, but not all pixels around lip contours for different utterance, and captures the motion tracks of each part. Accordingly, distinctive feature vectors are built to represent the whole lip motion process for specified utterance, but not the separate frame images. Experimental result shows the efficacy of the proposed approach.

1. Introduction

It is well known that the useful information about speech content can be obtained through analyzing the lip movements of speakers [1]. In 1984, the first automatic lip-reading system was presented by Petajan [2]. From then on, lip-reading has received considerable attention from the community because of its potential attractive applications in speech recognition, secret communication, and so forth.

So far, several methods have been proposed to enhance the performance of an automatic lip-reading system. Although some progress has been made, the recognition rate of lip-reading is still far from our expectation. One primary reason is that the visual features used in the existing systems cannot represent the entire information conveyed by lip movement related to a specified utterance [3][4].

Actually, the existing methods of visual feature representation can be classified into two categories in view of spatio-temporal relativity — the stationary based methods, and the motion based ones. For the former one, a video of lip movements is split into a sequence of images, on which the shape, appearance or texture of lip in each frame can be used for visual feature representation [6][7][8]. The recent reviews of existing typical automatic lip-reading system utilizing the "static feature representation" can be found in [4] and [9]. Although a lot of efforts have been spent on improving the performance of system based on "static feature representation", the performance of almost all of them is still quite poor.

On the other hand, along with the progress in motion perception and interpretation field, the motion based feature representation has recently played a more and more important role in automatic lip-reading system [10]. Compare to the "static feature representation", the motion based one can hold more crucial information for recognition. Many of the existing motion based feature representation methods use the optical flow field as the foundation in general. Optical flow is the apparent velocity distribution of the brightness patterns in an image. It can arise from relative motion of objects and the viewer. Consequently, optical flow can give important information regarding the spatial arrangement of the viewed objects and the rate of change of this arrangement [11]. Nevertheless, the existing optical flow based feature representations have some limitations. For instance, they are sensitive to the translation, scaling, rotation, and specially the change of optical condition. Furthermore, they depend on the speakers. Also, from the viewpoint of algorithm efficiency, the feature vector calculated by optical flow based methods always has high dimension and redundancy. Hence, the speed and efficiency of such an algorithm are relatively low [10].

In this paper, we propose a novel motion based feature representation. Different from the traditional optical flow methods, the proposed one dose not utilize all pixels in each image, but just some landmarks with the obvious lip movement. Since the lip movement for each utterance is different, the positions of landmarks are various in each image, and the number of these landmarks may not be constant. When the landmarks are chosen, the corresponding movement tracks will be also captured, whereby the feature is built to represent the lip movements.

2. Visual Feature

2.1. Feature Extraction

To extract the features exactly, some pretreatments are performed. We split the video of a number of utterances into an image sequence. The sample of a frame is shown in Figure 1(a), and the contours of nostrils and lips are shown in Figure 1(b). We calculate the center points of both nostrils as the datum mark, and adjust (translation and rotation) all images into the same co-ordinate.

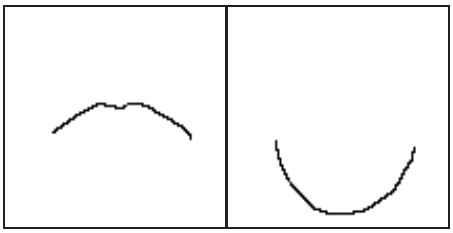
It is known that the most visual information can be conveyed from the lip contours, thus the four crucial curves involving upper outer, lower outer, upper inner and lower inner contours of lips are extracted, respectively, as shown in Figure 2.

(a) (b)

Figure 1. (a) Lip area in an example frame. (b) The contours of nostrils and lips.

Suppose the deformation of lips in horizontal and vertical directions is linear. That is, the relative position of each macroblock in the entire lip shape is constant during the deformation process. Hence, we divide the mouth into n scales in horizontal as shown in Figure 3. Subsequently, we obtain the points that are defined by scale lines and the four crucial curves so as to capture the motion tracks.

For a certain utterance, we calculate the landmarks in each frame and build two curve sets, respectively, by utilizing the horizontal and vertical motion of points on each crucial curve.



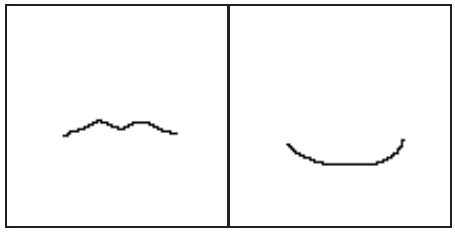


Figure 2. The four crucial curves of lip contours

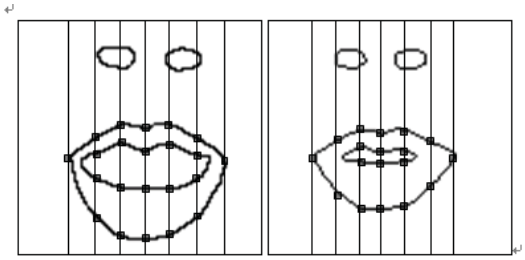


Figure 3. Two sample images when the seven landmarks are employed.

$$C_{i,x}=\begin{bmatrix} x_{i,1}^{1} & x_{i,1}^{2} & \cdots & \cdots & x_{i,1}^{k-1} & x_{i,1}^{k} \\ x_{i,2}^{1} & x_{i,2}^{2} & \cdots & \cdots & x_{i,2}^{k-1} & x_{i,2}^{k} \\ & & \vdots & \vdots & & \\ x_{i,n-1}^{1} & x_{i,n-1}^{2} & \cdots & \cdots & x_{i,n-1}^{k-1} & x_{i,n-1}^{k} \\ x_{i,n}^{1} & x_{i,n}^{2} & \cdots & \cdots & x_{i,n}^{k-1} & x_{i,n}^{k} \end{bmatrix}$$

$$C_{i,y}=\begin{bmatrix} y_{i,1}^{1} & y_{i,1}^{2} & \cdots & \cdots & y_{i,1}^{k-1} & y_{i,1}^{k} \\ y_{i,2}^{1} & y_{i,2}^{2} & \cdots & \cdots & y_{i,2}^{k-1} & y_{i,2}^{k} \\ & & \vdots & \vdots & & \\ y_{i,n-1}^{1} & y_{i,n-1}^{2} & \cdots & \cdots & y_{i,n-1}^{k-1} & y_{i,n-1}^{k} \\ y_{i,n}^{1} & y_{i,n}^{2} & \cdots & \cdots & y_{i,n}^{k-1} & y_{i,n}^{k} \end{bmatrix}$$

where $x_{i,j}^{k}$ is the horizontal position of the jth point on the ith crucial curve extracted from the kth frame, and $y_{i,j}^{k}$ is the corresponding vertical one.

Hence, for each utterance, we obtain eight curve sets that describe the motion track of each point. Each curve set is composed of n curves.

Ideally, the motion track of each point in a lip contour is consecutive and smooth. The stochastic reciprocate motion can be regarded as noise. In order

to avoid this kind of noise, a low pass filter is therefore employed.

Toward a filtered data set, we calculate the correlation coefficients between motion tracks of each nearby two points using the equations below:

$$\bar{x}_{i,j}^{l} = \frac{\sum_{l=1}^{k} x_{i,j}^{l}}{k}$$

$$r_{x_{j,j+1}} = \frac{\sum_{l=1}^{k} (x_{i,j}^{l} - \bar{x}_{i,j}^{l})(x_{i,j+1}^{l} - \bar{x}_{i,j+1}^{l})}{\sqrt{\sum_{l=1}^{k} (x_{i,j}^{l} - \bar{x}_{i,j}^{l})^2 \sum_{k=1}^{n} (x_{i,j+1}^{l} - \bar{x}_{i,j+1}^{l})^2}}$$

The detailed procedure is described as follows:

- Step 1: Get the curve set that should be operated.
- Step 2: Arrange the curves by the index of corresponding landmarks descend.
- Step 3: Assume that dimension of the curve set is *n*, we get the correlation coefficients between each adjacent two points.
- Step 4: For each correlation coefficient, if the value is greater than or equal to a pre-specified threshold, the two curves corresponding to the correlation coefficient are replaced by the mean curve of the two.
- Step 5: Update the curve set by the new one, and the dimension of the new curve set is *m*.
- Step 6: If *m* is equal to *n*, we let the curve set be the final one, otherwise go to Step 2.

For example, when we analyze the lip movement during the utterance "5" (*n*=30, *k*=20), the horizontal motion track of each point on the upper outer contour ($C_{1,x}$) is shown in Figure 4(a). The new set (marked as $C'_{1,x}$) is composed by the three feature curves as shown in Figure 4(b), and the middle points are 6, 17 and 25.

We use several samples to train our system, and obtain the scatter of middle points for each sample as shown in Figure 5. A set of intervals are utilized to describe these points (marked as $P_{1,x}$).

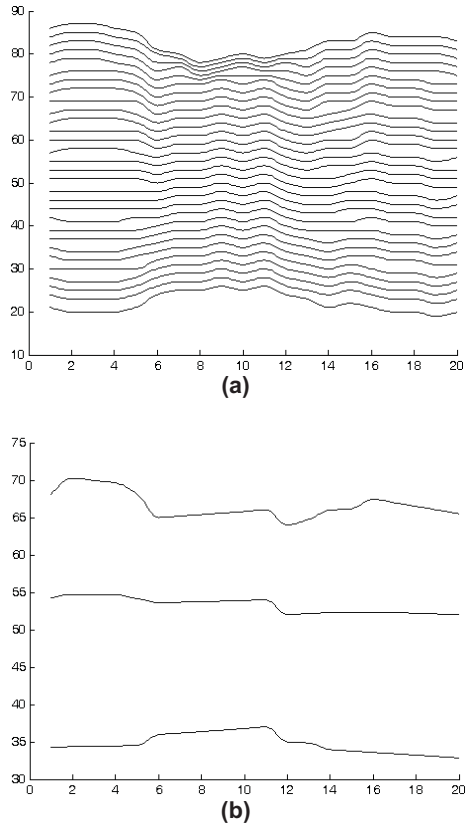


Figure 4. Horizontal motion track on the upper outer lip contour when speaking "5".

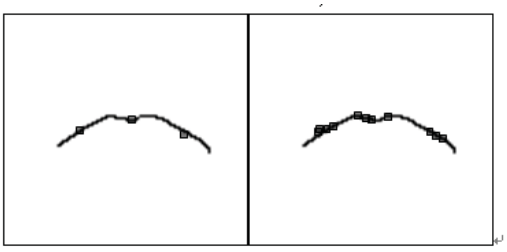


Figure 5. Scatter of middle points, with the initial lip shape overlaid.

The feature representation for each utterance is then shown below:

$$C = \left(C'_{1,x}, C'_{1,y} \cdots, C'_{4,x}, C'_{4,y}\right)$$
$$P = \left(P_{1,x}, P_{1,y}, \cdots, P_{4,x}, P_{4,y}\right)$$

2.2. Matching

The testing data are processed through the method above, the feature curves and the corresponding middle points are extracted. Compared middle points and intervals of them in training result, several utterances that have the same motion parts on lip contours are screened out.

Since the speaking velocities are different for each people, a cubic spline interpolation algorithm is employed to make *k* in each curve is same.

Then, we calculate the correlation coefficients between the feature curves of testing data and the corresponding curves in the same position for each utterance. The mean correlation coefficient is used to explain the comparability between the two curve sets. Since there are the eight curve sets for one utterance, namely $C'_{1,x}, C'_{1,y} \cdots, C'_{4,x}, C'_{4,y}$, the corresponding correlation coefficient between them and testing data are marked as $r_{1,x}, r_{1,y} \cdots, r_{4,x}, r_{4,y}$.

We use the weighted sum, written as *r*, with

$$r = r_{1,x} + r_{1,y} + r_{2,x} + r_{2,y} + w(r_{3,x} + r_{3,y} + r_{4,x} + r_{4,y})$$

to explain the comparability between the training utterance and testing utterance. As it is difficult that the inner contours of lip can be extracted exactly, the weight *w* of $r_{3,x}, r_{3,y}, r_{4,x}, r_{4,y}$ is suggested to use a small value.

When a testing is processed, *r* for each utterance in training set is calculated. The utterance that has the maximum *r* is considered as the class that the testing data belongs to.

4. Experiment Results

Since most of the existing database for lip-reading either are not available for public or do not have the appropriate language samples, a number of research groups develop their own visual-speech database [3][12]. Under the circumstances, we established a database for our experiments. The database consisted of six speakers (3 males and 3 females). Each speaker uttered ten isolated digits from zero to nine in Mandarin Chinese. Short pauses were inserted between each digit utterance for some speakers, but not all. In order to generate the normal lighting condition on speakers, two 36W fluorescent lamps were utilized, where they were placed at the left and right side in top of a speaker, respectively. Furthermore, to test the robustness of our system, several cameras with different resolutions and sample rates were used.

To investigate the efficacy of the proposed feature extraction in lip-reading, we have conducted several experiments. Because of the space limitation, we conducted an experiment to perform multi-speaker recognition task involving isolated digits (0 to 9). We utilized three data sets in the database as a training data set, and the remaining three data sets as the testing set. Figure 6 shows the two instances of horizontal motion trace on the upper outer lip contour when speaking “5”. Their weighted correlation coefficient is 0.913. In contrast, Figure 7 shows the two instances of vertical motion track with the correlation coefficient 0.802 as speaking “8”. The overall recognition rate of our approach on the testing set was 73.3%. To the best of our knowledge, this rate is considerably higher than the existing results reported in the literature.

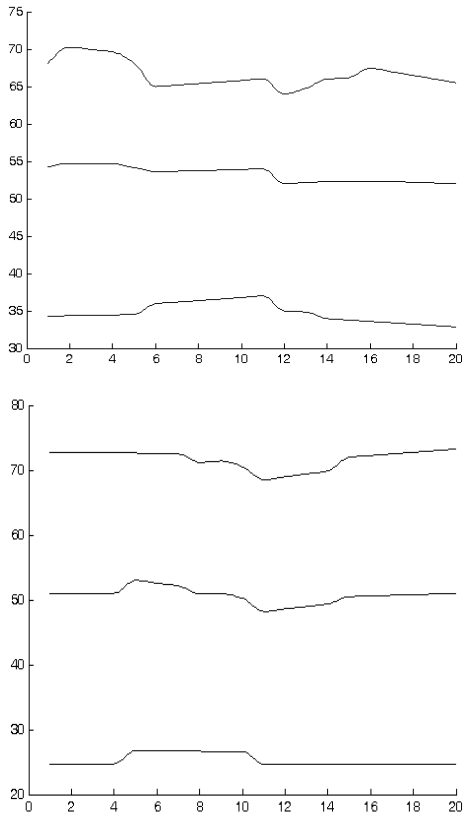


Figure 6. Two samples of horizontal motion track on the upper outer lip contour when speaking “5”, where the weighted correlation coefficient is 0.913.

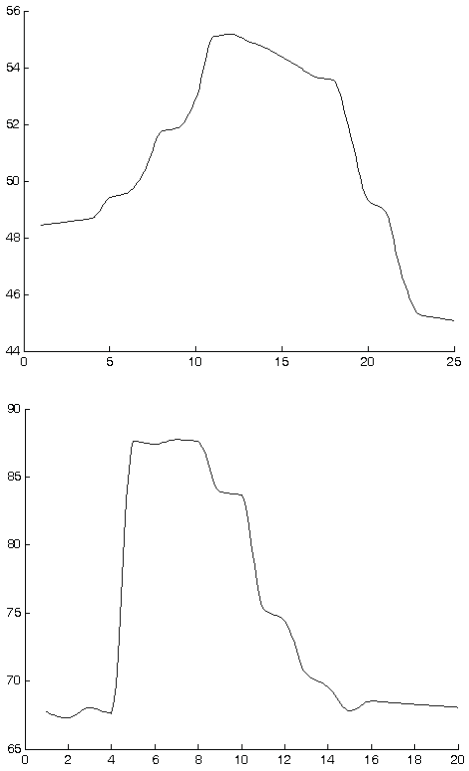


Figure 7. Two samples of vertical motion track on the upper outer lip contour when speaking "8", where the weighted correlation coefficient is 0.802.

5. Conclusion

In this paper, a motion based lip feature extraction method has been proposed for a speaker independent lip-reading system. Such a method provides an effective way to extract the features from the videos of lip movement. The proposed approach has been empirically investigated by different speakers. The experiments have shown the promising results.

References

[1] J. Bulwer. *Philocopus, or the Deaf and Dumbe Mans Friend*. Humphrey and Moseley, 1648.

[2] E. D. Petajan. Automatic lipreading to enhance speech recognition. PhD thesis, University of Illinois, 1984.

[3] I. Matthews, T. F. Cootes, J. A. Bangham, et al. Extraction of Visual Features for Lipreading. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 24(2):198-213, February 2002.

[4] H. X. Yao, W. Gao, et al, A Survey of Lipreading – One of Visual Languages. *ACTA Electronica Sinica*, 29(2): 239-246, February 2001.

[5] M. S. Gray, J. R. Movellan, T. J. Sejnowski. A*dvances in Neural Information Processing Systems*. The MIT Press, 1997.

[6] J. Luettin, N. A. Thacker. Speechreading Using Probabilistic Models. *Computer Vision and Image Understanding*, 65(2):163-178, February 1997.

[7] G. I. Chiou, J. N. Hwang. Lipreading by Using Snakes, Principal Component Analysis and Hidden Markov Models to Recognize Color Motion Video. *IEEE Trans. on Image Processing*, 6(8):1192-1195, 1997.

[8] T. F. Cootes, C. J. Taylor. A Mixture Model for Representing Shape Variation. *Image and Vision Computing*, 17(8):567-574, 1999.

[9] T. Chen, R. R. Rao. Audio-Visual Integration in Multimodal Communication. *Proc. IEEE, Special Issue on Multimedia Signal Processing*, 86(5):837-852, May 1998.

[10] C. Cedras, M. Shah. Motion-Based Recognition: A Survey. *Image and Vision Computing*, 13(2):129-155, March 1995.

[11] J. Barron, D. J. Fleet, S. S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43-77, February 1994.

[12] L. Liang, Y. Luo, F. Huang, A. V. Nefian. A Multi-Stream Audio-Video Large-Vocabulary Mandarin Chinese Speech Database. *IEEE International Conference on Multimedia and Expo*, 1787-1790, 2004.

Joint Feature Selection for Local Learning Based Clustering

Hong Zeng

Abstract

*In this paper, we present an effective method that jointly performs the feature selection and clustering. It is based on the recently proposed Local Learning based Clustering (*LLC*) algorithm [16], which adapts the local regression, a supervised learning technique, for clustering. We propose to incorporate the feature selection by imposing a weighted l_2 norm regularization on the regression coefficients, plus a l_1 norm constraint on the feature weights to encourage sparsity. The influence of the irrelevant features can be greatly mitigated by this adaptive regularizer, and the resulting feature weight vector is very sparse. An iterative algorithm is developed for solving the integrated feature selection and clustering problem. Experimental results on UCI, handwritten digits and microarray datasets demonstrate the effectiveness of the proposed algorithm.*

1. Introduction

In many pattern recognition and data mining problems, e.g. the computer vision, text processing and the more recent gene data analysis, etc., the input raw data sets often have a huge number of possible explanatory variables, but there are much fewer samples available. The abundance of variables makes the distinguishing among patterns much harder and less accurate. Under such circumstances, selecting the most discriminative or representative features of a sample inevitably becomes an important issue.

In the literature, most feature selection algorithms have been developed for supervised learning, rather than the unsupervised learning. It is believed that the unsupervised feature selection is more difficult due to the absence of class labels that can guide the search for the relevant information. Until very recently, several algorithms have been proposed to address this issue for clustering, and they can be generally categorized as the *filter* and *wrapper* methods. The *filter* approaches [7, 19] utilize the intrinsic properties of the features to filter out poorly informative ones before the clustering. They demonstrate great computational efficiency since they do not involve clustering when evaluating the feature quality. However, the issue of determining how many relevant features should be selected may cause difficult problem in practical applications, because the cross-validation, a commonly used model selection technique in supervised learning, cannot be directly applied in clustering, in which the ground truth class labels are unavailable. By contrast, the *wrapper* approach, e.g. [5, 9, 13, 15], repeatedly constructs a candidate feature subset, and then assesses its goodness by investigating the performance of a specific clustering on this feature subset according to certain criteria. In general, the *wrapper* approaches are computationally demanding, but they are expected to be more accurate than the *filter* ones, due to the performance feedback of the clustering when searching for the useful features. Some *wrapper* approaches, e.g. [5, 9], employ the heuristic (nonexhaustive) search through the space of all feature subsets, thus cannot provide any guarantee of optimality of the selected subset. This problem, as well as the problem of determining how many relevant features to select for *filter* approaches, can to some extent be alleviated by adopting feature weighting [13, 15], which assigns each feature a real-valued quantity, rather than a binary one, to indicate its relevance to the clustering. By casting the feature selection as an estimation problem, the combinatorial explosion of the search space could be avoided. Nevertheless, all the existing algorithms [13, 15] are constructed globally, they may get degenerated on high-dimensional datasets, since the dissimilarities among data in the high-dimensional space might become unreliable.

To this end, we propose a novel feature selection method for the clustering via a local learning approach, it extends the recently proposed Local Learning based Clustering (*LLC*) algorithm [16] with the feature selection capability. The *LLC* algorithm is essentially built upon the local regression technique, see Section 2 for a brief review. The complexity of the locally fitted regression model can be largely increased by the nuisance features, which cannot provide much information for the clustering. Thus in Section 3, we introduce real-valued feature weights for regularizing the squared magnitude of the regression coefficients, formulating a weighted l_2 norm regularized least square problem. Furthermore, we control the sparsity of feature weights by adding a l_1 norm regularization constraint on them. An alternating optimization algorithm is developed

for solving the integrated feature selection and clustering problem, presented in Section 3. We will show in Section 4, such an adaptive l_2 norm regularization, which balances the penalization on each coefficient according to the feature relevance to the prediction, is equivalent to the sparse-promoting squared block l_1 norm regularization. In Section 5, experiments with real-world datasets demonstrate that our method is able to infer both more accurate partitions and sparse subset of features. Finally, Section 6 provides some conclusions.

2. Review of the Local Learning based Clustering algorithm

Given n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n (\mathbf{x}_i \in \mathbb{R}^d)$, the dataset will be partitioned into C clusters. The result of the clustering algorithm is represented by a cluster indicator matrix $\mathbf{P} = [p_{ij}] \in \{0,1\}^{n \times C}$, such that $p_{ij} = 1$ if $\mathbf{x}_i$ belongs to cluster $\mathcal{C}_j$, and $p_{ij} = 0$ otherwise. The scaled cluster indicator matrix used in this paper is defined by:

$$\mathbf{Y} = \mathbf{P}(\mathbf{P^T P})^{-\frac{1}{2}} = [\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^C].$$

Thus $y_{ij} = p_{ij}/\sqrt{|\mathcal{C}_j|}$, it is easy to verify that

$$\mathbf{Y^T Y} = \mathbf{I}, \tag{1}$$

where $|\mathcal{C}_j|$ denotes the number of points in the cluster $\mathcal{C}_j$, $\mathbf{I}$ is the identity matrix. $\mathbf{y}^c = [y_1^c, \ldots, y_n^c]^T \in \mathbb{R}^n (1 \leq c \leq C)$, is the c-th column of $\mathbf{Y}$. The *LLC* algorithm tries to find a good clustering, or equivalently, the scaled cluster indicator $\mathbf{Y}$.

2.1. The *LLC* algorithm

The starting point of the *LLC* [16] is that the cluster indicator label of a data point should be well estimated by a model trained locally with its neighbors and their indicator labels. For each $\mathbf{x}_i$, the model is built with the training data $\{(\mathbf{x}_j, y_j^c)\}_{\mathbf{x}_j \in \mathcal{N}_i} (1 \leq c \leq C, 1 \leq i, j \leq n)$, where $\mathcal{N}_i$ denotes the set of neighboring [1] points of $\mathbf{x}_i$ (not including $\mathbf{x}_i$ itself) according to the Euclidean distance. The output of the linear model is of the following form:

$$f_i^c(\mathbf{x}) = \mathbf{w}_i^{cT} \mathbf{x}, \forall \mathbf{x} \in \mathbb{R}^d \tag{2}$$

where $\mathbf{w}_i^c \in \mathbb{R}^d$ is the local regression coefficients. In [16], the model is obtained by solving the following l_2 norm regularized least square problem:

$$\min_{\mathbf{w}_i^c \in \mathbb{R}^d} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \beta (y_j^c - \mathbf{w}_i^{cT} \mathbf{x}_j)^2 + \|\mathbf{w}_i^c\|^2 \tag{3}$$

[1] the K-mutual neighbors are adopted in order to well describe the local structure., i.e. $\mathbf{x}_j$ is defined as a neighbor of $\mathbf{x}_i$ only if $\mathbf{x}_i$ is also one of the K-nearest neighbors of $\mathbf{x}_j$.

where β is a pre-specified parameter. By solving (3), the predicted cluster label for $\mathbf{x}_i$ can be calculated by:

$$\widehat{y_i^c} = f_i^c(\mathbf{x}_i) = \mathbf{w}_i^{cT} \mathbf{x}_i = \mathbf{x}_i^T \mathbf{w}_i^c = \alpha_i^T \mathbf{y}_i^c \tag{4}$$

where

$$\alpha_i^T = \beta \mathbf{x}_i^T (\beta \mathbf{X}_i \mathbf{X}_i^T + \mathbf{I})^{-1} \mathbf{X}_i \tag{5}$$

$\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{in_i}]$ with $\mathbf{x}_{ik}$ being the k-th neighbor of $\mathbf{x}_i$, and $\mathbf{y}_i^c = [y_{i1}^c, y_{i2}^c, \ldots, y_{in_i}^c]^T$, $n_i = |\mathcal{N}_i|$. Note that α_i is independent of $\mathbf{y}_i^c$ and the cluster index c, and it is different for different $\mathbf{x}_i$.

After all the local predictors having been constructed, by combining them together, *LLC* aims to find an optimal cluster indicator matrix $\mathbf{Y}$ which could make the sum of the label prediction errors be minimized:

$$\begin{aligned} \mathcal{J}_{LLC} &= \sum_{c=1}^{C} \sum_{i=1}^{n} (y_i^c - \widehat{y_i^c})^2 \\ &= \sum_{c=1}^{C} \|\mathbf{y}^c - \mathbf{A}\mathbf{y}^c\|^2 \\ &= trace[\mathbf{Y}^T (\mathbf{I} - \mathbf{A})^{\mathbf{T}} (\mathbf{I} - \mathbf{A}) \mathbf{Y}] \\ &= trace(\mathbf{Y}^T \mathbf{T} \mathbf{Y}) \end{aligned} \tag{6}$$

where $\mathbf{T} = (\mathbf{I} - \mathbf{A})^{\mathbf{T}} (\mathbf{I} - \mathbf{A})$, $\mathbf{A}$ is an $n \times n$ sparse matrix with its (i,j)-th entry being:

$$a_{ij} = \begin{cases} \alpha_i^j, & \text{if } \mathbf{x}_j \in \mathcal{N}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

As in the spectral clustering [11, 17], $\mathbf{Y}$ is relaxed into the continuous domain while keeping the property (1) for the problem (6). *LLC* then solves the following tractable continuous optimization problem:

$$\begin{aligned} \min_{\mathbf{Y} \in \mathbf{R}^{n \times C}} \quad & trace(\mathbf{Y}^T \mathbf{T} \mathbf{Y}) \\ s.t. \quad & \mathbf{Y^T Y} = \mathbf{I} \end{aligned} \tag{8}$$

The solution to (8) can be obtained by setting columns of the $\mathbf{Y}$ matrix to be the first C eigenvectors of the matrix $\mathbf{T}$, corresponding to the first C smallest eigenvalues. Similar to the spectral clustering, the final partition result is obtained by discretizing $\mathbf{Y}$ via the method in [17] or by k-means as in [11]. Encouraging experimental results are reported in [16].

3. Joint feature selection for the Local Learning Clustering

From the last section, we could observe that the most important step in *LLC* algorithm is to construct the $\mathbf{T}$ matrix, thus we need to compute α_i, which in turn requires

calculating $\mathbf{w}_i^c$ $(\forall c)$ and predicting the labels of $\mathbf{x}_i$ through the inner product in (2). The inclusion of irrelevant features in the data may lead to a less accurate prediction by (2). If the regression coefficients which correspond to the irrelevant features could be shrunken towards zero, their influence could be reduced. To this end, rather than imposing an isotropic l_2 norm regularization on $\mathbf{w}_i^c$ in (3), which assumes all features are equivalently important, we propose a weighted l_2 norm regularization term, giving heavier penalization to the coefficients corresponding to the irrelevant features.

Specifically, we propose to solve the following weighted l_2 norm regularized least-square problem for $\mathbf{w}_i^c$:

$$\min_{\mathbf{w}_i^c \in \mathbb{R}^d, b_i^c \in \mathbb{R}} \sum_{\mathbf{x}_j \in \mathcal{N}_i^\tau} \beta(y_j^c - \mathbf{w}_i^{cT}\mathbf{x}_j - b_i^c)^2 + \mathbf{w}_i^{cT} diag(\tau^{-1})\mathbf{w}_i^c \tag{9}$$

with $\sum_{l=1}^d \tau_l = 1, \tau_l \geq 0$, which will be inferred latter. In (9), τ_l essentially controls the squared magnitude of the l-th element in $\mathbf{w}_i^c$, a smaller value for τ_l will result in a larger penalization for $\mathbf{w}_i^{c(l)}$; and when $\tau_l = 0$, we will prove in the sequel that it leads to $\mathbf{w}_i^{c(l)} = 0 \ \forall c$. Hence the l-th feature will be eliminated from the prediction, and the problem in (9) stays well-defined. Furthermore, a sparse-promoting l_1 norm constraint is added on τ, which should result in many τ_l's being close to zero. Therefore, τ_l indicates the "relevance" of the l-th feature to the clustering task, we will use it as the feature weight in the following discussion. As a consequence, the nearest neighbors $\mathcal{N}_i^\tau$ should be re-evaluated according to the τ-weighted square Euclidean distance, i.e.:

$$d_\tau(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_\tau^2 = \sum_{l=1}^d \tau_l (\mathbf{x}_1^{(l)} - \mathbf{x}_2^{(l)})^2. \tag{10}$$

To perform the joint feature selection for *LLC*, we develop an alternating update algorithm to estimate the clustering captured in $\mathbf{Y}$ and the feature weight τ.

3.1. Update Y for a given τ

With fixed feature weight τ, analytic solutions for problem (9) can be easily derived as follows:

$$\mathbf{w}_i^c = \beta[\beta \mathbf{X}_i \mathbf{\Pi} \mathbf{X}_i^T + diag(\tau^{-1})]^{-1} \mathbf{X}_i \mathbf{\Pi} \mathbf{y}_{\mathcal{N}_i^\tau}^c \tag{11}$$

$$b_i^c = \frac{1}{n_i} \mathbf{e}^T (\mathbf{y}_{\mathcal{N}_i^\tau}^c - \mathbf{X}_i^T \mathbf{w}_i^c) \tag{12}$$

where $n_i = |\mathcal{N}_i^\tau|$, $\mathbf{e} = [1\ 1 \cdots 1]^T \in \mathbb{R}^{n_i}$, $\mathbf{\Pi} = \mathbf{I}_{n_i} - \frac{1}{n_i}\mathbf{e}\mathbf{e}^\mathbf{T}$.

For high dimensional data, the matrix inversion in (11) will be very computational inefficient in time $(O(d^3))$. Fortunately, by applying the Woodbury's matrix inversion lemma, we could get:

$$\begin{aligned}\mathbf{w}_i^c =& \beta diag(\tau)\mathbf{X}_i\mathbf{\Pi}[\mathbf{I} - (\beta^{-1}\mathbf{I} + \mathbf{\Pi}\mathbf{X}_i^T diag(\tau)\mathbf{X}_i\mathbf{\Pi})^{-1} \\ & \mathbf{\Pi}\mathbf{X}_i^T diag(\tau)\mathbf{X}_i\mathbf{\Pi}]\mathbf{y}_{\mathcal{N}_i^\tau}^c \end{aligned} \tag{13}$$

Then the time complexity of the matrix inversion in (13) is only $O(n_i^3)$. In general we often have $n_i \ll d$, hence the computational burden could be largely alleviated. Besides, from (13), we could easily note that $\mathbf{w}_i^{c(l)}$ $(\forall i, c)$ goes to 0 as the feature weight τ_l vanishes.

Then the predicted cluster label for $\mathbf{x}_i$ will be obtained as follows:

$$\widehat{y_i^c} = \mathbf{w}_i^{cT}\mathbf{x}_i + b_i^c = \mathbf{x}_i^T \mathbf{w}_i^c + b_i^c = \alpha_i^T \mathbf{y}_{\mathcal{N}_i^\tau}^c \tag{14}$$

where

$$\begin{aligned}\alpha_i^T =& \beta(\mathbf{k}_i^\tau - \frac{1}{n_i}\mathbf{e}^T\mathbf{K}_i^\tau)\mathbf{\Pi}[\mathbf{I} - (\beta^{-1}\mathbf{I} + \mathbf{\Pi}\mathbf{K}_i^\tau\mathbf{\Pi})^{-1}\mathbf{\Pi}\mathbf{K}_i^\tau\mathbf{\Pi}] \\ & + \frac{1}{n_i}\mathbf{e}^T \end{aligned} \tag{15}$$

where we let $\mathbf{k}_i^\tau = \mathbf{x}_i^T diag(\tau)\mathbf{X}_i$, $\mathbf{K}_i^\tau = \mathbf{X}_i^T diag(\tau)\mathbf{X}_i$.

As in *LLC*, we construct the key matrix $\mathbf{T}$ by (6)-(7) with α_i. To solve the same optimization problem in (8), the columns of $\mathbf{Y}$ are simply set to the first C eigenvectors of $\mathbf{T}$ corresponding to the smallest C eigenvalues.

3.2. Update τ for a given Y

The inference of τ is based on the local model parameter $\mathbf{w}_i^c$ $(\forall i, c)$ already fitted by samples and $\mathbf{Y}$ through (13). Specifically, note that with the fixed $\mathbf{w}_i^c$ in the weighted l_2 norm regularization problem (9), estimating τ actually requires solving the following minimization problem:

$$\begin{aligned}\min_{\tau \in \mathbb{R}^d} \quad P(\tau) =& \sum_{c=1}^C \sum_{i=1}^n \mathbf{w}_i^{cT} diag(\tau^{-1}) \mathbf{w}_i^c \\ =& \sum_{l=1}^d \frac{\sum_{c=1}^C \sum_{i=1}^n (\mathbf{w}_i^{c(l)})^2}{\tau_l} \\ s.t. \quad & \sum_{1=1}^d \tau_l = 1, \tau_l \geq 0. \end{aligned} \tag{16}$$

At optimality, we have with Lagrangian method:

$$\tau_l = \frac{\sqrt{\sum_{c=1}^C \sum_{i=1}^n (\mathbf{w}_i^{c(l)})^2}}{\sum_{m=1}^d \sqrt{\sum_{c=1}^C \sum_{i=1}^n (\mathbf{w}_i^{c(m)})^2}} \tag{17}$$

The equation (17) is very intuitive: the l-th feature weight τ_l is determined by the magnitude of the l-th element in regression coefficients for all the clusters, which

have been locally solved at each point. If this element in the regression coefficients has neglectable magnitude for all the clusters at each point, it probably indicates that the corresponding feature is unimportant for predicting the label.

3.3. The joint feature selection and clustering algorithm

The overall iterative joint feature selection and clustering algorithm is described in Algorithm 1, which has a simple interpretation: in each iteration, it first constructs cluster-specific local models, with fixed common feature weights, for solving the optimal labeling; then it learns the common across-clusters feature weights using the fitted local model parameters, with the current labeling. The algorithm stops

input : $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, size of the neighborhood K, β
output: $\mathbf{Y}, \tau$

Initialize $\tau_l = \frac{1}{d}$, for $l = 1, \ldots, d$;
while *not converge* **do**
  Find K-mutual neighbors for $\{\mathbf{x}_i\}_{i=1}^n$, using the metric defined in (10);
  Update $\mathbf{Y}$ as in Section 3.1;
  Update τ as in Section 3.2;
end

Algorithm 1: Joint feature selection for local learning based clustering algorithm.

when the relative variation of the trace value in (8) between two consecutive iterations gets below a threshold (we set it to 10^{-2} in this paper), indicating the partitioning has almost stabilized. After the convergence, the same method in [11] is adopted to obtain the discrete clustering result.

4. Sparse norm equivalence

In this section, it will be shown that the proposed weighted l_2 norm regularization plus a l_1 norm constraint on these weights, is equivalent to a well-known sparse-promoting squared block l_1 norm regularization. We simply address this equivalence based on the fact that the variational formulation of the squared block l_1 norm regularization is equal to the adaptive weighted l_2 norm ones.

Theorem 1.

$$\min_{\tau_l \geq 0, \sum_l \tau_l = 1} \sum_l \frac{\|\widetilde{\mathbf{W}}_l\|^2}{\tau_l} = \left(\sum_l \|\widetilde{\mathbf{W}}_l\| \right)^2 \qquad (18)$$

where we define $\|\widetilde{\mathbf{W}}_l\| = \sqrt{\sum_{c=1}^C \sum_{i=1}^n (\mathbf{w}_i^{c(l)})^2}$.

Proof. Actually, we have obtained that the optimal solution to the problem (16) or (18) has been given by (17). Then at the optimality, plugging (17) into (18) proves the variational formulation of $\left(\sum_l \|\widetilde{\mathbf{W}}_l\| \right)^2$. □

In the literature, without regard to the index i indicating for a local model, the squared block l_1 norm regularization, in the form $\left(\sum_l \|\mathbf{W}_l\| \right)^2 = \left[\sum_l \sqrt{\sum_{c=1}^C (\mathbf{w}^{c(l)})^2} \right]^2$ (where $\mathbf{W} = [\mathbf{w}^1\ \mathbf{w}^2\ \ldots \mathbf{w}^C] \in \mathbb{R}^{d \times C}$, $\mathbf{w}^c = [\mathbf{w}^{c(1)}\ \mathbf{w}^{c(2)}\ \ldots \mathbf{w}^{c(d)}]^T \in \mathbb{R}^d$) has been successfully applied in several applications, e.g., multiple kernel learning [3], multi-task feature selection [2, 4], etc.. In fact, the block l_1 norm $\left(\sum_l \|\mathbf{W}_l\| \right)$ could be viewed as a combination of the l_1-norm regularization on the feature level and the l_2-norm regularization on the class level. Sparsity is encouraged due to the l_1-norm regularization, with many rows of $\mathbf{W}$ being close to zero. Therefore according to (18), the proposed adaptive l_2 norm regularization should be able to produce at least as sparse as that of the squared block l_1 norm regularization.

5. Experimental results

In this section, we conducted extensive experiments to demonstrate the effectiveness of the proposed algorithm. Ten benchmark data sets were used in our experiments, and their characteristics are summarized in Table 1. On each data set, we investigated whether our joint feature selection for local learning based clustering algorithm (denoted as *LLC-fs*) could improve the *LLC* algorithm and the *k-means* clustering, which assume all features are equivalently important. Furthermore, *LLC-fs* was compared with the state-of-the-art *wrapper* method, Q-α algorithm [15] [2], which is also an eigen-decomposition based approach as ours. It is based on optimizing the desired spectral property of the

Table 1. Characteristics of the datasets used in experiments

Data Set	#DIM (d)	#INST (n)	#CL (C)
wdbc	30	569	2
mfea-kar	64	2000	10
mfea-fac	216	2000	10
mfea-pix	240	2000	10
USPS 4 vs.9	256	1673	2
USPS 0 vs.8	256	2261	2
colon cancer	2000	62	2
SRBCT	2308	63	4
leukemia	3051	38	2
breast cancer	3303	44	2

[2] Its MATLAB source code was obtained from the authors in [15].

feature weight incorporated affinity matrix, globally constructed with all the data. The algorithm in [13] was not compared as it can only deal with binary-class clustering.

In the experiments, we set the number of clusters equal to the number of classes C for all the clustering algorithms. As the labels of all five benchmark datasets are available, we evaluated the their performance by comparing the clusters generated by these algorithms with the ground truth classes, in terms of the accuracy (ACC) index [18]. Given a data point $\mathbf{x}_i$, let c_i and t_i be the obtained cluster label and the true class label from the data, respectively. The accuracy measurement is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(t_i, map(c_i))}{n} \tag{19}$$

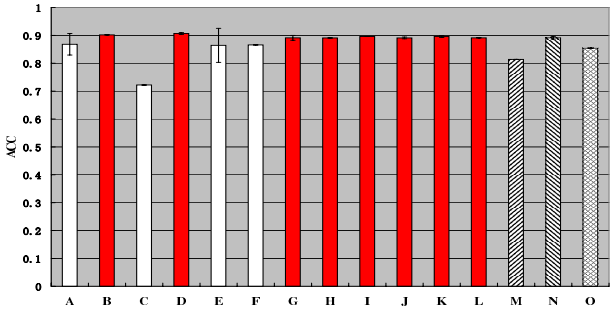
where n is the number of the data set, and $\delta(t_i, t_j)$ is the delta function that equals 1 if $t_i = t_j$ and equals 0 otherwise. The $map(.)$ is a permutation mapping function that maps each cluster index c_i to a true class label. This optimal mapping can be found with the Kuhn-Munkres algorithm [12].

5.1. UCI datasets

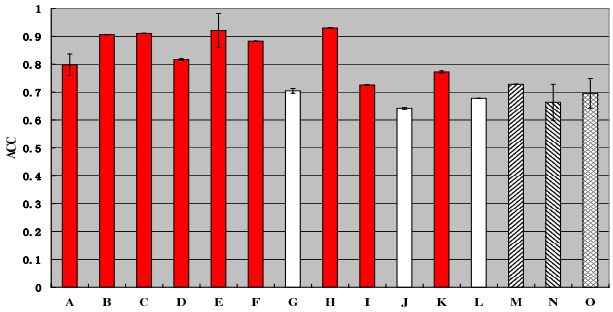
The first four datasets in Table 1 are from the UCI repository [10]. The Wisconsin diagnostic breast cancer dataset (`wdbc`) was used to obtain a binary diagnosis (benign or malignant) based on the features extracted from cell nuclei presented in an image. The `mfea-kar`, `mfea-fac` and `mfea-pix` datasets are all from the "multiple feature database"[10]. This database consists of features of handwritten numerals ("0-9") extracted from a collection of Dutch utility maps. Digits are represented in serval sets of features, we used the datasets using the Karhunen-Love coefficients (`mfea-kar`), the profile correlations (`mfea-fac`) and the pixel averages (`mfea-pix`). No preprocessing was performed except on the `mfea-fac` dataset, which has many features of significantly different scales, each feature was then normalized to zero mean and unit variance.

Both the *LLC* and the *LLC-fs* algorithms have 2 parameters: the size of the mutual neighbors K, and the local regulation constant β^{-1}. In this section, for both *LLC* and *LLC-fs*, K is limited to $\{10, 20, 30, 40\}$ and β^{-1} from $\{0.1, 1, 10\}$, providing the same computation resources. They were executed with each combination of K and β^{-1}, and the whole process was repeated 10 times. We report the mean and the standard deviation of the ACC index for *LLC-fs* on each combination; for simplicity, we just report the results of *LLC* with the best parameter setting among the $4 \times 3 = 12$ combinations. Q-α and *k-means* have no parameters, their performance over 10 runs are presented.

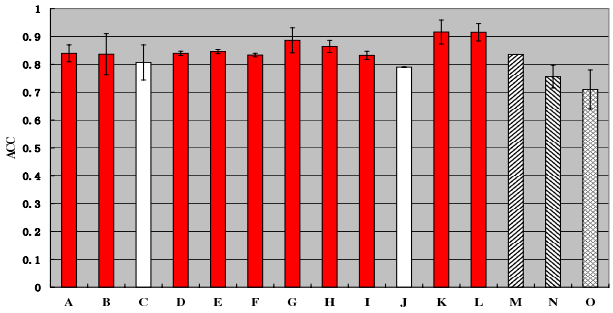
The results are summarized in Figure 1. From figure 1, it could be noted that *LLC-fs* with most of combinations



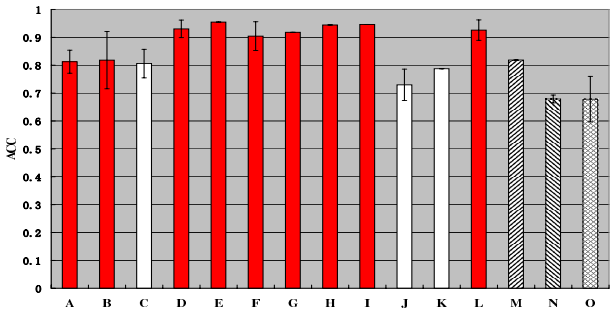
(a) wdbc



(b) mfea-kar



(c) mfea-fac



(d) mfea-pix

Figure 1. Clustering accuracy measurements on four UCI datasets. A-L denote the *LLC-fs* with different combinations of parameters $\{K, \beta^{-1}\}$, where A: $\{K(1), \beta^{-1}(1)\}$, B: $\{K(1), \beta^{-1}(2)\}$, ..., L: $\{K(4), \beta^{-1}(3)\}$. M: *LLC* (best); N: Q-α; O: *k-means*. Among A-L, the ones which outperform or get tied with the best one among M, N and O, are marked in red.

of $\{K, \beta^{-1}\}$ outperforms the *LLC* with the best parameter setting, as well as the baseline *k-means*, since it is able to alleviate the influence of irrelevant features. Compared to the global based *wrapper* approach Q-α, our local based one shows similar performance on low dimensional `wdbc` dataset, but demonstrates remarkable superiority over all the three "multiple feature" datasets of medium and high dimensionalities. A reasonable explanation is given as follows. The dissimilarities in the high-dimensional space is known as much less reliable. However, the neighboring information is believed to be more capable to preserve the authentic grouping within the dataset, even though there may be many irrelevant features.

5.2. Handwritten digits datasets

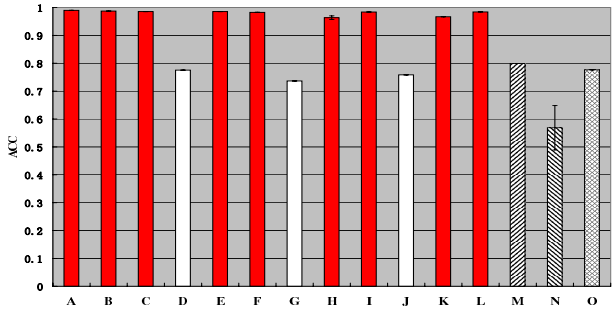
In this case study, we focus on the task of clustering on the USPS ZIP code[3] handwritten digits, which are 16×16 grayscale images. Each image is thus represented by a 256-dimensional vector. In particular, we considered two difficult binary-class clustering problems, i.e. digits "4 vs. 9" and "0 vs. 8".

The experimental settings were almost the same as in the last section, except the size of mutual nearest neighbors K. We increased its value since it is known that there are heavy overlappings between each pair, a larger neighborhood may help obtain a more accurate prediction of the label ($K = \{30, 40, 50, 60\}$ for "4 vs. 9", and $K = \{35, 40, 50, 60\}$ "0 vs. 8"). The figure 2 presents the results obtained over 10 runs for all the algorithms on these two datasets.

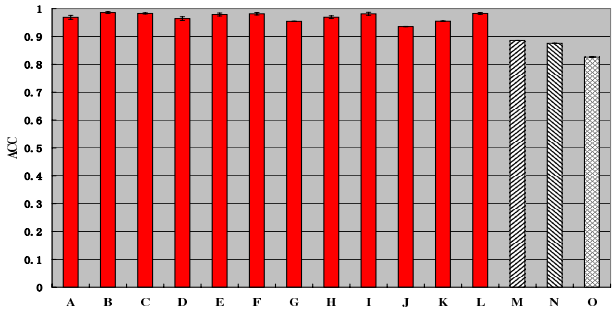
It could be observed from Figure 2 that *LLC-fs* with most combinations of $\{K, \beta^{-1}\}$, significantly outperforms other algorithms compared on both datasets. Encouragingly, the accuracy of *LLC-fs* with the best parameters could achieve around 98% on both "4 vs. 9" and "0 vs. 8" datasets (see in Figure 2), in an unsupervised manner.

To get a better understanding of what features have been ranked top by our weighting scheme, we show in Figure 3 the top features in the image domain. Firstly, the sorted τ in a typical run on each dataset is presented in Figure 3(a),3(b) respectively, with $K = 30, \beta^{-1} = 1$ for the "4 vs. 9" dataset, and $K = 35, \beta^{-1} = 1$ for the "0 vs. 8" dataset. As can be seen that the both τ vectors are sparse, and only few of the feature weights are above a very clear threshold. Nextly, we plot the 15 top-ranked features in image domain. One can find that these features have covered almost all the strongly discriminative regions for each digit pairs, hence resulting in more accurate partitions by the proposed *LLC-fs* algorithm (see Figure 2).

[3]http://www-stat-class.stanford.edu/∼tibs/ElemStatLearn/data.html



(a) USPS 4 vs.9

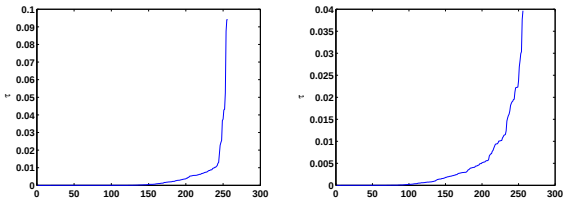


(b) USPS 0 vs.8

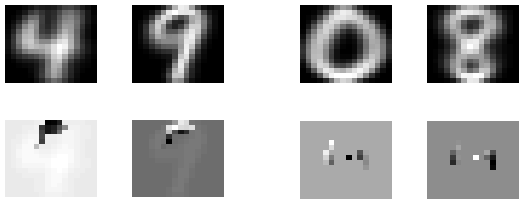
Figure 2. Clustering accuracy measurements on USPS 4 vs.9 and USPS 0 vs.8 datasets. A-L denote the *LLC-fs* with different combinations of parameters $\{K, \beta^{-1}\}$, where A: $\{K(1), \beta^{-1}(1)\}$, B: $\{K(1), \beta^{-1}(2)\}$, ..., L: $\{K(4), \beta^{-1}(3)\}$. M: *LLC* (best); N: Q-α; O: *k-means*. the ones which outperform or get tied with the best one among M, N and O, are marked in red.

5.3. Microarray datasets

In this experiment, we studied the joint feature selection and clustering on four public gene expression datasets: `colon cancer` [1], `SRBCT` [8], `leukemia` [6], `breast cancer` [14]. The characteristics of these datasets are summarized in Table 1. For the `colon cancer` dataset, we preprocessed the data by carrying out a base 10 logarithmic transformation. For `SRBCT`, the expression profiles already preprocessed in [8] was used. For `leukemia`, the expression values were first thresholded with a floor of 100 and a ceiling of 16,000. Then we filtered out genes with $max/min \leq 5$ or $(max - min) \leq 500$, where max and min are the maximum and minimum expression values of a gene. After a base 10 logarithmic transform, each gene was standardized to zero mean and unit variance across samples. For `breast cancer` dataset, 5 conflicting samples were excluded from the analysis. We

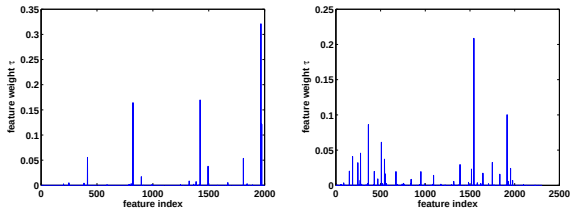


(a) USPS 4 vs.9 (b) USPS 0 vs.8

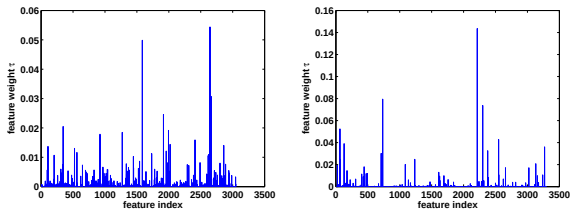


(c) USPS 4 vs.9 (d) USPS 0 vs.8

Figure 3. Unsupervised feature selection for clustering the USPS digits. (a),(b): the (sorted) τ values on the "4 vs. 9" and "0 vs. 8" datasets, respectively. In (c) and (d): the first row plots the class mean images; the second row shows the top 15 features ranked by the τ weight vector in each mean image.



(a) colon cancer (b) SRBCT



(c) leukemia (d) breast cancer

Figure 4. The feature weight vector τ for the microarray datasets.

then thresholded the raw data with a floor of 100 and a ceiling of 16,000, filtered out genes with $max/min \leq 10$ or $(max - min) \leq 1,000$. After a base 10 logarithmic transform, each gene was standardized to zero mean and unit variance across samples.

For these genomics data, we set the K in the range $\{20, 30, 40\}$ for all the data, except the leukemia dataset that has only 38 samples, for which we limit it to $\{20, 25, 30\}$. β^{-1} was still selected in the grid $\{0.1, 1, 10\}$. The *LLC-fs* and *LLC* with all the $3 \times 3 = 9$ combinations of parameters, as well as the Q-α algorithm and *k-means* were repeated 10 times, the mean and standard deviation of the ACC index are summarized in Figure 5. For clarity, we only report the results for *LLC* with the best parameters.

From Figure 5, one can observe that *LLC-fs* with most combinations of $\{K, \beta^{-1}\}$ performs better than the pure clustering algorithms *LLC* and *k-means* on all the four datasets, even with the best parameters tried among all the 9 combinations in the experiments. Again, our local based joint feature selection and clustering approach wins over the global based counterpart Q-α algorithm on these high-dimensional data.

The typical feature weighting results in the 10 runs are also plotted in Figure 4, all with $K = 30, \beta^{-1} = 1$ (other successful combinations of parameters produce similar results, thus we do not plot them for clarity). For each dataset, the τ is quite sparse, only few of them have significant magnitudes while most feature weights are close to zero. This could explain the reason why the *LLC-fs* significantly improves the performance of *LLC* on these data: it can jointly identify these most relevant genes for class discovery.

6. Conclusions

In this paper, we have presented a novel approach of jointly selecting discriminative features for the local learning based clustering [16]. Based on the *LLC* algorithm which is essentially built upon the local ridge regression, we propose to replace the conventional l_2 norm regularization with a weighted one, and impose sparsity requirement on the feature weights via a l_1 norm constraint. The influence of the irrelevant features can be greatly reduced and the estimated feature weight vector is very sparse. Furthermore, we have proved the equivalence between this adaptive l_2 norm regularizer and the sparse-promoting squared block l_1 norm regularizer. The experimental results demonstrate that the proposed *LLC-fs* algorithm with most of the combinations of parameters has shown better clustering performance, than *LLC* even with its best parameters among the pre-specified parameter pools. Compared to the global *wrapper* approach, the local method *LLC-fs* could drastically outperform Q-α algorithm on the high-dimensional datasets.

However, it also could be noted from the experiments,

not all the combinations of the parameters for *LLC-fs* could outperform the other three algorithms, but the performance of these "failed" combinations do not fall much behind the best one among the other three algorithms compared (c.f. Figure 1, Figure 2, Figure 5). In other words, the proposed *LLC-fs* algorithm shows robustness in the choice of parameters to some extent. However, it would be highly desirable to provide an automatical parameters tuning method, and we intend to improve the this issue in our future work.

References

[1] U. Alon, N. Barkai, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *PNAS*, volume 96, pages 6745–6750, 1999.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *NIPS*, pages 41–48, 2007.

[3] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of ICML*, pages 41–48, 2004.

[4] J. Bi, T. Xiong, S. Yu, M. Dundar, and R. B. Rao. An improved multi-task learning approach with applications in medical diagnosis. In *Proceedings of ECML*, 2008.

[5] J. Dy and C. Brodley. Feature selection for unsupervised learning. *JMLR*, 5:845–889, 2004.

[6] T. Golub, D. Slonim, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[7] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *NIPS*, 18:507–514, 2005.

[8] J. Khan, J. Wei, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, 7:673–679, 2001.

[9] M. Law, A. Jain, and M. Figueiredo. Feature selection in mixture-based clustering. *NIPS*, pages 609–616, 2002.

[10] D. Newman, S. Hettich, C. Blake, and C. Merz. *UCI Repository of Machine Learning Databases*. 1998.

[11] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002.

[12] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithm and Complexity*. Dover, 1998.

[13] V. Roth and T. Lange. Feature selection in clustering problems. *NIPS*, 2004.

[14] M. West, C. Blanchette, et al. Predicting the clinical status of human breast cancer by using gene expression profiles. In *PNAS*, volume 98, pages 11462–11467, 2001.

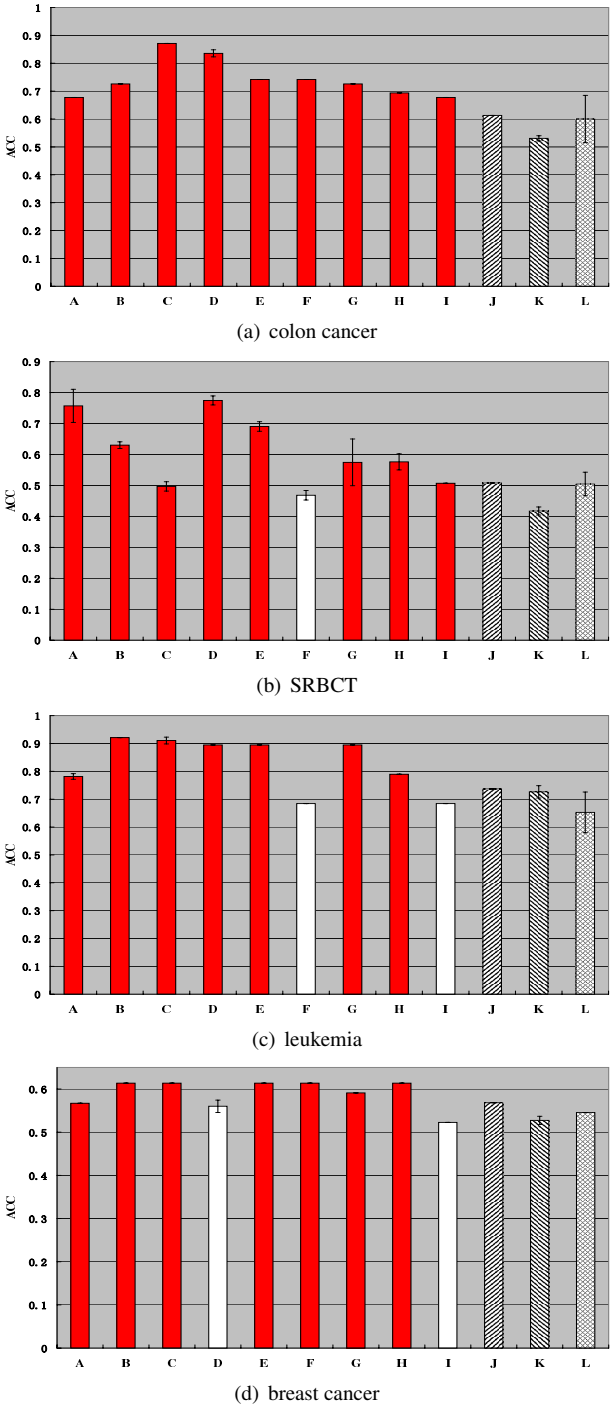
[15] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *JMLR*, 6:1855–1887, 2005.

[16] M. Wu and B. Schölkopf. A local learning approach for clustering. *NIPS*, 19:1529–1536, 2007.

[17] S. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of ICCV*, pages 313–319, 2003.

[18] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *NIPS*, 2001.

[19] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of ICML*, pages 1151–1158, 2007.



(a) colon cancer

(b) SRBCT

(c) leukemia

(d) breast cancer

Figure 5. Clustering accuracy measurements on Genomics datasets. A-I denote the *LLC-fs* with different combinations of parameters $\{K, \beta^{-1}\}$, where A: $\{K(1), \beta^{-1}(1)\}$, B: $\{K(1), \beta^{-1}(2)\}$, ..., I: $\{K(3), \beta^{-1}(3)\}$. J: *LLC* (best); K: Q-α; L: *k-means*. the ones which outperform or get tied with the best one among M, N and O, are marked in red.

RBF Kernel-Based Discretization for Face Template Security

Feng Yicheng

Abstract

Being an important issue, the security of biometric authentication systems has received more and more attentions. Cryptographic schemes (cryptosystems) have been applied to solve these kind of problems. However, as one of the most important biometrics, security enhancement of face recognition systems come up against extreme difficulties. The intra-class variations become a huge hindrance standing in front of the cryptosystems. This paper addresses the face variations problem via transforming the original face templates into binary strings. It provides a completely new way to do the discretization, Kernel mapping is applied here because of its distance rank preserving property. Through the kernel mapping and kernel discretization, the binary strings in the same class only have small variations and can be easily handled by cryptographic schemes like the fuzzy commitment scheme, while the binary strings in different classes will have large distances with the help of assisting PINs. Thus our algorithm can achieve a high performance. Experimental results show that our algorithm reaches an extremely high performance (the equal error rate is 1.91% with CMU PIE database, 0.41% with FERET database and 1.93% with FRGC database) with the help of PINs.

1 Introduction

Biometric recognition is a reliable, robust and convenient way for person authentication [6, 7, 4]. Due to growing concerns about information security and terrorism, several large biometric systems such as US-VISIT program have been successfully deployed. With the growing use of biometrics, there is a rising concern about the security and privacy of the biometric template itself. Since the uniqueness of the biometric data to each person, it will be unable to find a replacement while the biometric is compromised. And the compromised biometric may be used for criminals. Therefore, biometric template security [6, 7, 4, 21] becomes one of the most important issues of the biometric authentication system. To protect the biometric templates, templates stored in database should be transformed and in matching process, the original templates should not be extracted from the transformed data. Otherwise the matcher may be hijacked and the information will be stolen. Thus, the matching process should be executed in a transformed domain. The main approach for doing this is the cryptosystem [6, 4, 21], which combines cryptography with biometric authentication systems.

Enhancing security of face recognition systems is especially hard. The intra-class variations existing in face templates have been a significant problem while constructing a face cryptosystem. The transform (or encryption) applied to protect the templates should be non-invertible, that is, a one-way function. However, one-way functions suffer from the sensitivity of variations. Even small variation may cause large difference through the transformation. Thus, different templates in the same class may be recognized as two totally different individuals by the cryptosystem after transform. Error-correcting codes (fuzzy schemes) [9, 10, 11] have been applied to model the variations in biometrics. It can eliminate the variations to a certain extent. However, their correcting ability are limited. And unfortunately, due to the environment affections and alignment problems, there are often large variations existing in the original face feature templates extracted.

So the consideration turns to this: first refine the original feature templates, transform them to new ones in which the intra-class variations can be modelled by fuzzy schemes. This is always a quantization [23] or discretization [14, 15, 17, 18, 24, 25] process. Then the error-correcting process can be applied to model the variations and transform the discretized templates into a protected (eg. hash) ones. Thus it is a two-stage structure (shown in Fig. 1). Apparently, binarization comes to the no doubt choice in the dicretization stage because binary string is the most common used, convenient, and error-correcting feasible format in cryptography. Like the BCH coding (one kind of error-correcting codes), it can only handle binary strings. This kind of schemes have been proposed and testified to be feasible. However, no one has explained why its discretization process can handle the variations. Actually, the performances of this kind of schemes proposed by Monrose *et al.* [14, 15] are not very well. And the biohashing algorithms [17, 18] proposed by Teoh *et al.* need an extra user data to help enhancing discriminability, which is easily

compromised. Without the extra user data, the performance of the algorithm will fall dramatically.

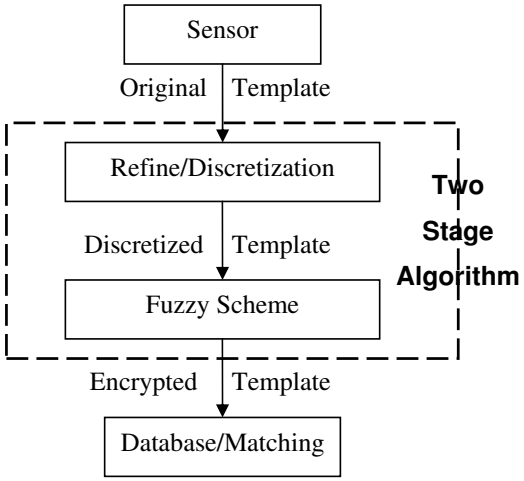


Figure 1. The two-stage crypgosystem approach

Another issue which should be considered is the cancelability. If some face template is compromised, the cryptosystem should be able to generate a new different one to replace it. A main approach named Cancelable transform [20, 21, 22, 19] is applied to solve this problem. The same problem occurs: the transform should be both one-way and discriminability preserving. The cancelable transform schemes always have to choose a trade off between security and performance.

In this paper, in order to enhance the security of the face recognition systems, we follow the two-stage discretization approach while an extra stage is combined together to form a three-stage structure. In the extra (first) stage, RBF kernel mapping is applied as a cancelable transform, because it is distance rank preserving (thus discriminability preserving). Also, because the mapped data is totally unknown to the system, the kernel mapping step will be entirely secure. In the second stage, the mapped kernel templates are discretized to binary strings with the kernel parameters used in the first stage. And in the last stage, fuzzy commitment scheme is applied for error-correcting and protection. Our main contribution, the RBF kernel based transform not only preserves the discriminability of the original templates, but also provides a complete new way as a cancelable transform and discretization process. The extracted binary strings are not directly computed, but are educed via a kernel bases learning process from the training data. In this leaning process, the discriminability lost in transform is reduced gradually via iteration, until a set of desirable kernel bases are learned with a satisfyingly preserved discriminability.

The rest of this paper is organized as follows. Section 2 & 3 gives the description of our proposed scheme. Experimental results and security analysis are given in Section 4 & 5. And the final conclusion is given in Section 6.

2 RBF Kernel-Based Three-Stage Algorithm

In this paper, we have proposed a hybrid cryptosystem (illustrated in Fig. 2) based on the RBF kernel function. In the first stage, a kernel mapping $\varphi()$ is applied to the original feature templates. For the property of the kernel mapping, the mapping result is actually unknown to the system. Thus it will be completely non-invertible and therefore secure. The RBF kernel mapping is distance rank preserving thus discriminability preserving. Via changing the parameter of this kernel mapping process, the mapped kernel templates are cancelable. Thus, the first stage is a well cancelable transform. However, since we don't know the mapped results, we are unable to use them as the stored reference. And a further process, that is, the discretization process is employed here. This stage will transform the kernel templates into binary strings. The bits of each binary string is actually the coefficients of an approximation to the kernel templates, which are trained with iteration algorithm. Furthermore, these binary templates are encrypted with the fuzzy commitment scheme to protect them.

2.1 Stage 1: kernel mapping

First we define the **distance rank preserving** property: Assume Ω is a set of vectors with specified length p. If there is a mapping $f()$, which satisfies for any four vectors $T_1, T_2, T_3, T_4 \in \Omega$,

$$\| f(T_1) - f(T_2) \| \geq \| f(T_3) - f(T_4) \|$$

if

$$\| T_1 - T_2 \| \geq \| T_3 - T_4 \|,$$

we call the mapping $f()$ is **distance rank preserving** on set Ω. With the INN classifier the classification result of Ω will be the same before and after mapping.

In this stage, we apply a popular RBF kernel mapping $\varphi()$ which for any four vectors T_1, T_2, T_3 and T_4 with the same length, it satisfies

$$\| \varphi(T_1) - \varphi(T_2) \| \geq \| \varphi(T_3) - \varphi(T_4) \|$$

if

$$\| T_1 - T_2 \| \geq \| T_3 - T_4 \| .$$

Thus, the mapping $\varphi()$ is distance rank preserving. This kernel function can be written as

$$K(x, y) = < \varphi(x), \varphi(y) > = \exp(- \| x - y \|^2 / d).$$

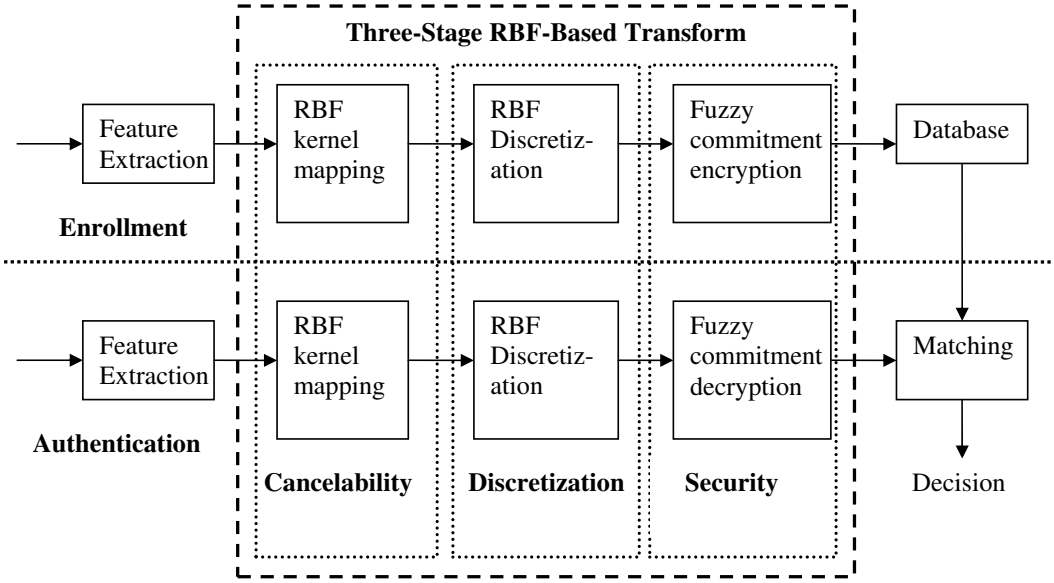


Figure 2. Our proposed RBF-kernel based algorithm

and thus

$$\| \varphi(x) - \varphi(y) \| = 2 - 2\exp(- \| x - y \|^2 / d). \quad (1)$$

Here d is a positive parameter alternative. In our experiments we set it as the square mean of the original templates. Because the RBF mapping is distance rank preserving, the performance of authentication on kernel templates after mapping will be the same as the performance of the original templates with the INN classifier. Thus, the mapping process is discriminability-preserving. Because the exact position after mapping of the original template is unknown, attackers will never have the chance to recover the original template from the mapped one. Thus the RBF mapping is both one-way and non-invertible. Unfortunately, because even the system itself doesn't know what is the mapped kernel template, we can not store it to the database as an enrolled reference. A discretization process is employed here to further transform the kernel template into binary strings.

2.2 Stage 2: RBF discretization

2.2.1 Enrollment

In enrollment, assume the wanted binary string length is k, then k random vectors $\{V_1, V_2 \dots V_k\}$ are preliminarily defined for the binarization/discretization process. After kernel mapping they are transformed into kernel vectors $\{\varphi(V_1), \varphi(V_2), \varphi(V_3) \dots \varphi(V_k)\}$. Assume there are total c classes and the mean vectors of the training set of each class are $\{T_1, T_2, T_3 \dots T_c\}$ respectively. For the j^{th} class, k bases $\beta_1^j, \beta_2^j \dots \beta_k^j$, in which

$$\beta_i^j = \sum_{p=1}^{k} b_{ip}^j \cdot \varphi(V_p); \quad (2)$$

And for any kernel template $\varphi(T_j)$ of the j^{th} class, a binary string $\omega_j = (w_1^0, w_2^0, w_3^0 \dots w_k^0)$ is employed so that $X_j = \sum_{i=1}^{k} w_i^0 \beta_i^j$ will approximate $\varphi(T_j)$. That is, the following equation is satisfied at ω_j:

$$J_j = \min_{w_1, w_2 \dots w_k} \| \varphi(T_j) - \sum_{i=1}^{k} w_i \beta_i^j \|^2 \quad (3)$$

The approximation X_j is then used to replace the kernel template $\varphi(T_j)$. Because X_j has the minimum distance to $\varphi(T_j)$, the replacing will only affect the distance rank a little. And because the $\{\varphi(V_1), \varphi(V_2), \varphi(V_3) \dots \varphi(V_k)\}$ are actually unknown to the system, it will not release the information of $\varphi(T_j)$. Since the kernel templates are vectors, assume A_j is the matrix with k rows in which the i^{th} row is β_i^j. Similarly, B_j is the $k \times k$ matrix consisting of $\{b_{ip}^j\}$, matrix U consists of $\{\varphi(V_p)\}$. Therefore,

$$A_j = B_j \cdot U. \quad (4)$$

$$\| \varphi(T_j) - \omega_j \cdot A_j \|^2 = \min. \quad (5)$$

After the approximation, the binary string ω_j is used to represent the approximating templates X_j). To be distance rank preserving, we assume matrix A_j (un-normalized) is orthogonal. That is:

$$A_j \cdot A_j^T = m^2 \cdot I \qquad (m > 0).$$

Substitute Eq. (4) to this equation, we have:

$$B_j \cdot U \cdot U^T \cdot B_j^T = m^2 \cdot I. \tag{6}$$

Then for any j_1 and j_2,

$$\begin{aligned} \| X_{j1} - X_{j2} \|^2 &= \\ \| \omega_{j1} \cdot A_j - \omega_{j2} \cdot A_j \|^2 &= \\ \| (\omega_{j1} - \omega_{j2}) \cdot A_j \|^2 &= \\ (\omega_{j1} - \omega_{j2}) \cdot A_j \cdot A_j^T \cdot (\omega_{j1} - \omega_{j2})^T &= \\ m^2 \cdot (\omega_{j1} - \omega_{j2}) \cdot (\omega_{j1} - \omega_{j2})^T &= \\ m^2 \cdot \| \omega_{j1} - \omega_{j2} \|^2 & \end{aligned}$$

The distance is completely preserved through the representing of X_j by ω_j with ratio m. Thus, the distance rank is completely preserved with this process. Finally, we get a binary string ω_j to represent the original kernel template $\varphi(T_j)$. Because the distance rank of $\{\omega_j\}$ is identical to $\{X_j\}$ and the distance rank of $\{X_j\}$ is only a little different from $\{\varphi(T_j)\}$, it makes us be able to claim that the kernel discretization which transforms $\varphi(T_j)$ to ω_j is approximately distance rank preserving.

So the next question is: How to find the ω_j and B_j satisfying Eq. (5) and Eq. (6) with restriction that ω_j is binary. This is described in Sec. 3.

At the end of the RBF discretization, a PIN is generated from derived B_j and issued to the user j. B_j is stored in the database.

2.2.2 Authentication

In authentication, the user j' presents his biometric data T'_j and the user PIN to find the right B'_j for RBF discretization. With known B'_j, we can construct the kernel bases A'_j by the following equation:

$$A'_j = B'_j \cdot U.$$

And we want to find the binary string ω'_j, which makes

$$\| \varphi(T'_j) - \omega'_j \cdot A'_j \|^2 = \min. \tag{7}$$

Here

$$\begin{aligned} \| \varphi(T'_j) - \omega'_j \cdot A'_j \|^2 &= \\ 1 - 2\omega'_j \cdot A'_j \cdot \varphi(T'_j)^T + \omega'_j \cdot A'_j \cdot A_j'^T \cdot \omega_j'^T &= \\ 1 - 2\omega'_j \cdot A'_j \cdot \varphi(T'_j)^T + m_j'^2 \| \omega'_j \|^2 & \end{aligned}$$

The last equation is because $A'_j \cdot A_j'^T = m_j'^2 \cdot I$. Let $\gamma^T = A'_j \cdot \varphi(T'_j)^T$, then

$$\| \varphi(T'_j) - \omega'_j \cdot A'_j \|^2 = 1 - 2\omega'_j \cdot \gamma^T + m_j'^2 \| \omega'_j \|^2 .$$

Denote l as the binary string with length k and all bits being '1'. Then $\| \omega'_j \|^2 = \omega'_j \cdot l^T$. Substitute it to the above equation, we have:

$$\| \varphi(T'_j) - \omega'_j \cdot A'_j \|^2 = 1 + \omega'_j \cdot (m_j'^2 l - 2\gamma)^T.$$

Assume $\gamma = (z_1, z_2 \ldots z_k)$, $\omega'_j = (w'_1, w'_2 \ldots w'_k)$ then

$$m_j'^2 l - 2\gamma = (m_j'^2 - 2z_1, m_j'^2 - 2z_2 \ldots m_j'^2 - 2z_k).$$

Obviously, the equation will reach its minimum while:

$$w'_i = \begin{cases} 1 & \text{if} \quad 2z_i > m_j'^2; \\ 0 & \text{if} \quad 2z_i \le m_j'^2. \end{cases} \tag{8}$$

And the derived binary string ω'_j is applied to represent the original template T'_j. It is input to the stage 3 and then compared with the ω_j representing class j.

If the original template T'_j belongs to the class j, then $B'_j = B_j$, $A'_j = A_j$. Thus,

$$\begin{aligned} m'_j \| \omega_j - \omega'_j \| &= \\ \| (\omega_j - \omega'_j) \cdot A_j \| &= \\ \| X_j - \varphi(T_j) + \varphi(T_j) - \varphi(T'_j) + \varphi(T'_j) - X'_j \| &\le \\ \| X_j - \varphi(T_j) \| + \| \varphi(T_j) - \varphi(T'_j) \| + \| \varphi(T'_j) - X'_j \| & \end{aligned}$$

Here $X_j = \omega_j \cdot A_j$ and $X'_j = \omega'_j \cdot A_j$. From Eq. (5) and (7), $\| X_j - \varphi(T_j) \|$ and $\| X'_j - \varphi(T'_j) \|$ reaches its minimum, thus

$$\| \omega_j - \omega'_j \| \approx \| \varphi(T_j) - \varphi(T'_j) \| / m'_j.$$

It shows clearly that the distance rank preserving property still maintains in the testing data within the same class.

If T'_j doesn't belong to class j, then $A'_j \neq A_j$. Similarly, we have

$$\| X_j - X'_j \| \approx \| \varphi(T_j) - \varphi(T'_j) \| .$$

That is,

$$\| \omega_j \cdot A_j - \omega'_j \cdot A'_j \| \approx \| \varphi(T_j) - \varphi(T'_j) \| . \tag{9}$$

However, because A_j and A'_j are totally unrelated, the above relation between ω_j and ω'_j is totally useless, which means we can treat ω_j and ω'_j as two totally unrelated binary string. This will cause a large distance between them, thus highly improves the performance of the algorithm.

2.3 Stage 3: fuzzy commitment scheme

The fuzzy commitment scheme [9] and BCH [2] coding algorithm is applied for security. It encrypts the training binary template s to a hashed BCH codeword $Hash(c)$ which is stored in the database with an extra information $s - c$. And the matching process is done in hash space. Assume a query

s' is input to the system with a Hamming distance $s' - s$ small to s. The system will first compute $s' - (s - c)$ with the extra information $s - c$. $s' - (s - c) = (s' - s) + c$. For $s' - s$ is rather small, it will be corrected by the BCH decoding process. Then $c' = \text{BCHdecoding}((s' - s) + c) = c$ will be released, hashed and compared with the original stored data Hash(c).

3 B_j And $\{\omega_j\}$ Determination

3.1 Optimization with Eq. (5)

In our algorithm, We should find B_j and ω_j that make Eq. (5) be minimum. Here

$$\begin{aligned} J_j &= \| \varphi(T_j) - \omega_j \cdot B_j \cdot U \|^2 \\ &= (\varphi(T_j) - \omega_j \cdot B_j \cdot U) \cdot (\varphi(T_j) - \omega_j \cdot B_j \cdot U)^T \\ &= (1 - 2\omega_j \cdot B_j \cdot U \cdot \varphi(T_j)^T \\ &\quad + \omega_j \cdot B_j \cdot U \cdot U^T \cdot B_j^T \cdot \omega_j^T) \end{aligned}$$

Denote $\alpha_j = (a_1, a_2 \dots a_k) = \omega_j \cdot B_j$, substitute it to the above equation:

$$J_j = (1 - 2\alpha_j \cdot U \cdot \varphi(T_j)^T + \alpha_j \cdot U \cdot U^T \cdot \alpha_j^T).$$

Since the above equation reaches its minimum at α_j, partial differential J_j with respect to a_i respectively, we can get:

$$\frac{dJ_j}{d\alpha_j} = 2U \cdot U^T \cdot \alpha_j^T - 2U \cdot \varphi(T_j)^T = 0.$$

Thus,

$$\alpha_j = \varphi(T_j) \cdot U^T \cdot (U \cdot U^T)^{-1}.$$

That is,

$$\omega_j \cdot B_j = \varphi(T_j) \cdot U^T \cdot (U \cdot U^T)^{-1}. \quad (10)$$

Assume $W = U \cdot U^T$, then W is a RBF kernel matrix which is positive definite and symmetric. Thus, W^{-1} exists. We should notice here that though we don't know $\varphi(T_j)$ or U, we can compute $\varphi(T_j) \cdot U^T$ and W because the elements of them are all inner products. Similarly, the finally derived equations below to determine B_j and ω_j are computable.

3.2 Optimization with Eq. (6)

For the distance rank preserving property, we add the following condition:

$$B_j \cdot U \cdot U^T \cdot B_j^T = m^2 \cdot I.$$

Substitute $W = U \cdot U^T$ in it:

$$B_j \cdot W \cdot B_j^T = m^2 \cdot I.$$

For W is a symmetric definite positive matrix, assume its Cholesky decomposition is $R \cdot R^T$, in which R is a nonsingular triangular matrix. Substitute it to the above equation:

$$B_j \cdot R \cdot R^T \cdot B_j^T = m^2 \cdot I.$$

Assume $B_j \cdot R = m_j \cdot Q_j$, then

$$Q_j \cdot Q_j^T = I.$$

Q_j is orthogonal. And

$$B_j = Q_j \cdot R^{-1}/m_j \quad (R \text{ is nonsingular}). \quad (11)$$

Substitute it to Eq. (7) and assume $v_j = \varphi(T_j) \cdot U^T \cdot (U \cdot U^T)^{-1}$, we have:

$$\omega_j \cdot Q_j = v_j \cdot R/m_j.$$

Assume $u_j = v_j \cdot R$. Since Q_j is orthogonal,

$$\| \omega_j \| = \| \omega_j \cdot Q_j \| = \| u_j/m_j \| .$$

$$m_j = \frac{\| u_j \|}{\| \omega_j \|} \quad (12)$$

Normalize u_j and ω_j to u_{j0} and ω_{j0}, then

$$\omega_{j0} \cdot Q = u_{j0}. \quad (13)$$

3.3 Determine $\{\omega_j\}$ and $\{B_j\}$

Since $X_j = \omega_j \cdot A_j$ is an approximation to $\varphi(T_j)$, then $\| X_j \|$ should also be close to $\| \varphi(T_j) \|$. That is:

$$\omega_j \cdot A_j \cdot A_j^T \cdot \omega_j^T \approx 1.$$

Since $A_j \cdot A_j^T = m_j^2 \cdot I$,

$$\omega_j \cdot \omega_j^T \approx 1/m_j^2.$$

Thus,

$$\| \omega_j \| \approx 1/m_j.$$

Because in our algorithm the scalar m_j is not protected, it may be exposed to the attacker. Then the attacker will use m_j to get an approximate value of $\| \omega_j \|$, which is just the number (denoted as r) of bit '1' in the string. Then the attacker may apply the Bruce-force attack to guess the binary string with C_k^r times. For security concern, we should maximize C_k^r, thus, choose r to be $[k/2]$.

With the given $r = [k/2]$, ω_j is randomly generated with just r bits of value '1'. Then we have to derive B_j from ω_j.

From Eq. (8) we can see that we only have to determine the orthogonal matrix Q_j, which is restricted by Eq. (10).

We apply the following process to construct Q_j, thereby determine the B_j:

- Randomly generate a $k \times k$ non-zero upper triangular matrix P_j, thus P_j is nonsingular.
- Find the first non-zero element in vector ω_{j0}, and denote the position of this element in ω_{j0} as x.
- replace the x^{th} row of P_j with ω_{j0}. Thus P_j is still a nonsingular upper triangular matrix.
- Exchange the x^{th} row and the first row of P_j. Finally, P_j is not upper triangular but is still nonsingular, while the first row of P_j is ω_{j0}.
- Apply the Gram-Schmidt Orthonormalization to P_j. Because ω_{j0} is normalized, it will not change in this step. After this step, P_j is orthogonal with the first row being ω_{j0}.
- Repeat the above steps to construct another orthogonal matrix S_j, while the first row of S_j is u_{j0}.
- $Q_j = P_j^T \cdot S_j$.

With the constructed Q_j, we can see

$$\omega_{j0} \cdot Q_j = \omega_{j0} \cdot P_j^T \cdot S_j = [1, 0, 0 \ldots 0] \cdot S_j = u_{j0}.$$

and Q_j is orthogonal because P_j and S_j are orthogonal. Thus, the constructed Q_j is what we wanted. And

$$B_j = Q_j \cdot R^{-1}/m_j.$$

4 Experimental Results

4.1 Experiments setting

In the experimental results, we apply there face databases, the CMU PIE, FERET and FRGC databases for our algorithm testing. The Fisherface [1] algorithm is chosen for feature extraction. The CMU PIE database includes 68 individuals and 105 images per person. The FERET database includes 250 individuals and 4 images per person. And the FRGC database includes 350 individuals and 40 images per person. We choose 10 (CMU PIE), 2 (FERET) and 5 (FRGC) images in each class for training, and the rest for testing. In our test, the length of the binary string is chosen 200 for all the three databases. Our algorithm is compared with the random multispace quantization (RMQ) [19] algorithm. The length of the transformed binary string is chosen 50 (CMU PIE) and 200 (FERET, FRGC) respectively. We consider two scenarios similar as the RMQ algorithm.

4.2 Performance analysis

Here we assume the PINs of the individuals are secure thus no one have the other one's PIN. Thus, everyone trying to access the system should present his own PIN and own face data. The experimental results of this scenario are shown in Fig. 3-5. The symbol "RBF-D" denotes our RBF kernel method with different PINs for different individuals, and "Original" means the performance of the original Fisherface algorithm.

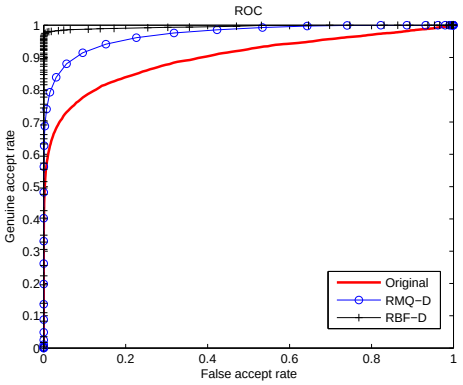


Figure 3. The experimental results of the CMU PIE database while the query and the reference use different PINs.

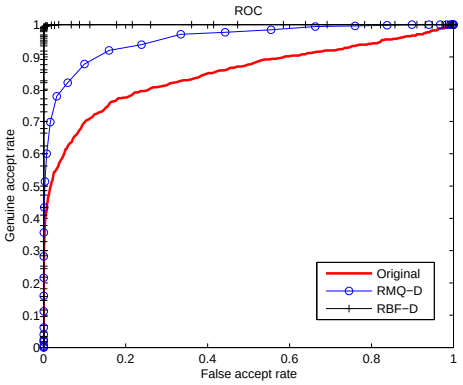


Figure 4. The experimental results of the FERET database while the query and the reference use different PINs.

From these figures it is obvious that our RBF kernel method outperforms the RMQ algorithm and the original Fisherface algorithm. The equal error rates (EER) of the three different methods are shown in Tab. 1.

With the above experimental results, we can see that our

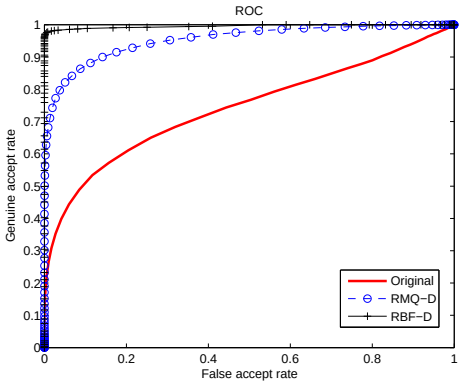


Figure 5. The experimental results of the FRGC database while the query and the reference use different PINs.

Table 1. The EERs of the experiments

ERR(%)	Original	RMQ-D	RBF-D
CMU PIE	17.32	9.01	1.91
FERET	21.66	11.08	0.41
FRGC	31.75	11.55	1.93

algorithm will get an almost perfect performance in this scenario. It is because the different user PINs significantly increase the distances between the binary templates of different classes (analyzed in Sec. 2.2.2).

5 Security Analysis

The security of the system relies on the security of the transformed binary templates because the kernel templates are totally undeterminable. And if the binary templates are compromised, attackers may use them with the random vectors $V_1, V_2 \dots V_k$ to reconstruct T_j via inner product computations. So the security of the system depends on the security of the binary templates. In the third stage, the binary templates are encrypted via the fuzzy commitment scheme. The hash function in the scheme is rather strong (e.g. the MD5 [3] hash function has a security level of 2^{128}), which can be treated as secure enough.

However, from Sec. 3.3, we know that the attacker may successfully get the number of bits '1' in the binary template ($[k/2]$). In this case, he/she may apply a brute-force attack with trying about $C_k^{[k/2]}$ times. Thus, the security of the binary templates are about $C_k^{[k/2]}$. That is, the attacker should spend so much times to access the system. In our experiments, $k = 200$ thus the security level is about $C_{200}^{100} \approx 2^{195}$.

The cancelability of our algorithm is rather high because actually even the reference binary templates are randomly generated (Sec. 3.3). If someone is compromised, it can be cancelled and new reference can be generated to replace it easily. So our algorithm benefits from the high cancelability.

6 Conclusion

In this paper, we have proposed a bran-new way to discretize the original face templates into binary strings, such that the transformed binary templates can be easily protected by biometric cryptosystem. Kernel mapping and discretization is applied because it is secure and distance rank preserving, thus preserves the performance of the original system. The performance of the system can even be highly enhanced via the help of the assistant PIN. Our experimental results show that the performance of our algorithm is nearly perfect (EER is only 1.91% in CMU PIE database, 0.41% in FERET database and 1.93% in FRGC database) with the help of PINs. In conclusion, our algorithm can not only highly enhance the security of the original system, and also enhances the original performance of it, with the benefit of cancelability.

References

[1] 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.

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

[3] R. L. Rivest, "The MD5 Message-Digest Algorithm," *RFC1321, Network Working Group, MIT Laboratory for Computer Science and RSA Data Security, Inc.*, 1992.

[4] N Ratha, J Connell and R Bolle, "Enhancing security and privacy in biometric-based authentication systems," *IBM Systems Journal*, Vol. 40. No. 3, pp. 614 - 634, 2001.

[5] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric Recognition: Security and Privacy Concerns," *IEEE Security and Privacy Magazine*, Vol. 1, No. 2, pp. 33-42, March-April 2003.

[6] U Uludag, S Pankanti, S Prabhakar, and A K Jain, "Biometric cryptosystems: issues and challenges," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948-960, 2004.

[7] A K Jain, A Ross and S Pankanti, "Biometrics: A Tool for Information Security", *IEEE Transactions on Information Forensics and Security,* Vol. 1, No. 2, pp. 125-143, 2006.

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

[9] A Juels, M Wattenberg, "A fuzzy commitment scheme", *Sixth ACM Conf. on Comp. and Comm. Security,* pp. 28-36, 1999.

[10] A Juels and M Sudan. "A Fuzzy Vault Scheme", *IEEE International Symposium on Information Theory,* 2002.

[11] T C Clancy, N Kiyavash, and D J Lin, "Secure smartcard-based fingerprint authentication", *Proc. ACMSIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop,* pp. 45-52, 2003.

[12] S. Draper, A. Khisti and E. Martinian, A. Vetro and J. Yedidia, "Using distributed source coding to secure fingerprint biometric," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[13] Y Sutcu, Q Li and N Memon, "Protecting biometric template with sketch: theory and practice," *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3 Part 2, pp. 503-512 , 2007.

[14] F. Monrose, M.K. Reiter and S. Wetzel, "Password Hardening Based on Key Stroke Dynamics," *Proc. ACM Conf. Computer and Comm. Security*, pp. 73-82, 1999.

[15] F. Monrose, M. Reiter, Q. Li and S. Wetzel, "Cryptographic Key Generation from Voice," *Proc. IEEE Symp. Security and Privacy*, pp.202-213, May 2001.

[16] A. Teoh, D. Ngo and A. Goh, "Biohashing: Two Factor Authentication Featuring Fingerprint Data and Tokenised Random Number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245-2255, Nov. 2004.

[17] A Goh and D C L Ngo, "Computation of cryptographic keys from face biometrics", *in Proc. 7th IFIP TC6/TC11 Conf. Commun. Multimedia Security,* vol. 22, pp. 1-13, 2003.

[18] D Ngo, A Teoh, and A Goh, "Biometric Hash: High-Confidence Face Recognition", *IEEE transactions on circuits and systems for video technology,* vol. 16, no. 6, 2006.

[19] A. Teoh, A. Goh, D. Ngo, "Random Multispace Quantization as an Analytic Mechanism for BioHashing of Biometric and Random Identity Inputs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1892-1901, Dec. 2006.

[20] R Ang, R Safavi-Naini, L McAven, "Cancelable Key-Based Fingerprint Templates," *ACISP 2005*, pp. 242-252.

[21] N Ratha, J Connell, R Bolle, S Chikkerur, "Cancelable biometrics: A case study in Fingerprints", *Proceedings of International Conference on Pattern Recognition*, 2006.

[22] N Ratha, S Chikkerur, J Connell, R Bolle, "Generating Cancelable Fingerprint Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, pp. 561-752, 2007.

[23] Y. C. Feng and P. C. Yuen, "Protecting Face Biometric Data on Smartcard with Reed-Solomon Code," *IEEE CVPR Workshop on Biometrics*, pp. 29-34, 2006.

[24] Y C Feng and P C Yuen, "Class-Distribution Preserving Transform for Face Biometric Data Security," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 141-144, 2007.

[25] Y C Feng and P C Yuen, "Selection of Distinguish Points for Class Distribution Preserving Transform for Biometric Template Protection," *Proceedings of IEEE International Conference on Biometrics (ICB)*, In press, 2007.

Boosting EigenActions: A new algorithm for Human Action Categorization*

Chang Liu

Abstract

This paper proposes a new algorithm for human action categorization by a boosted classifier from EigenActions which are represented by information salient features. To determine the EigenActions, a spatio-temporal information saliency is first calculated from the video sequence by estimating pixel density function. Salient action units are extracted and projected into an informative feature space by principle component analysis. A human action recognition system is finally trained on this feature space by multi-class Adaboost algorithm. We have tested our proposed method using a widely used human action database with ninety different human actions. Experimental results are encouraging. A comparison with two latest methods on human action recognition is also reported.

1. Introduction

Human action categorization has been receiving increasing attentions from researchers in computer vision research area. The aim of human activity categorization is to teach a computer to recognize human actions from videos so as to make further classification or semantic description of the actions [12] [7] [16] [14]. It has wide applications such as visual surveillance, human-computer interfaces, content based video retrieval etc. Human action recognition is a challenging research area because the dynamic human body motions have unlimited underlying representations and meanings. The popular approach for human action recognition is based on action features [1] [13] [6] [2] and encouraging results have been reported. However, the problem is how to extract the most discriminative motion features and shape features for action classification.

Recently, the spatio-temporal shape features in the 3D XYT volume have been used for human activity recognition. These kinds of features often shows more informative properties in video analysis. Yilmaz et al [15] used spatio-temporal volume for human activity recognition. The 3D contour of motion is projected to a 2D surface and the projection with the time axis forms the new spatio-temporal volume. Then the human moving speed, moving direction and human shape can be extracted from the volume. Activity recognition can then be performed. Laptev [9] proposes a space-time interest point detector, it can find local structures in space-time volume where the image values have significant local variations in both spatial and temporal domain. However, a small number of stable interest points may not sufficient to characterize complex events. Shechtman et al. have recently [13] proposed a spatio-temporal patch matching based method for human activity recognition. Small spatio-temporal volumes (reference volume) are correlated against the entire video sequences in all XYT three dimensions, in each location, all spatio-temporal patches in the reference volume are correlated with relative spatio-temporal patches in the target volume. The overall peak correlation values correspond to similar activities. Gorelick et al [6] [3] utilized the properties of the poisson equation solution to analyze the spatio-temporal volume. Three dimensional space-time shapes are generated from the silhouettes of the spatio-temporal volume. The space-time salient features are then extracted from the space-time shape, and it shows that these features are very useful for activity recognition. Juan et al [8] proposed a mixture hierarchical model for human activity recognition based on spatial and spatio-temporal features. They showed that static shape features can improve the recognition performance when using the spatio-temporal features.

Inspired from the recent research progress on spatio-temporal human action recognition, we propose to recognize human actions by boosting EigenActions based on spatio-temporal information saliency. We calculate an information saliency map which shows temporal periodic properties on periodic motions which have local maximum value in saliency. We construct a Salient Action Unit (SAU) from the spatio-temporal volume. The EigenActions that contain more information on the underlying actions are then extracted and projected to their eigen space in a lower dimension. Finally, a multi-class Adaboost classifier is trained with optimal Bayes weak classifier. Our proposed classifier can recognize human actions accurately and efficiently.

The rest of this paper is organized as follows. Section 2 will report the details of Salient Action Unit construction based on the Information Saliency Map. Section 3 will re-

* This paper has been accepted in "IEEE International Conference on Automatic Face and Gesture Recognition, 2008" for oral presentation.

port the EigenActions Representation and Adaboost classifier training. Experimental results and the conclusion are given in Sections 4 and 5 respectively.

2. Salient Action Unit

Motions of objects often have a periodic nature. The repeated action atomic plays a fundamental role for action recognition and categorization. In this section, we propose to employ the spatio-temporal Information Saliency Map (ISM) [11] for Salient Action Unit(SAU) segmentation. The salient action units will then be used for training a human behavior recognition system.

2.1. Information Saliency Map

From Shannon's information theory, a rarely happened event contains high information while an event which happens frequently contains low information. An Information Saliency Map (ISM) can be built [11] from the video while each entry of ISM reflects the spatio-temporal saliency of the corresponding pixel in that video frame. Considering the current frame Im_0 is divided into $h \times w$ smaller patches $\{Im_{1,1}, Im_{1,2}, ..., Im_{h,w}\}$. The ISM for Im_0 can be obtained by the Spatio-Temporal ISM model [11]

$$I_{r,s} = -\log_2\left(\frac{P(X|V)P(x_0|X)+P(X'|V)P(x_0|X')}{P(X|V)[1-P(x_0|X)]+P(X'|V)[1-P(x_0|X')]}\right) \quad (1)$$

where x_0 is the vector form of $Im_{i,j}$, the temporal vector set $X = (x_0, x_1, ..., x_{N-1})$ is constructed from concatenating N temporal patches located at $Im_{r,s}$, which is also called a sub-volume. The spatial vector set $X' = (x'_0, x'_1, ..., x'_{N'-1})$ is constructed by the patch (x_0) in Im and its $N'-1$ spatial neighborhoods. V is chosen as the spatio-temporal cube that contains X and X', where $\{X, X'\} \subset V$. Figure 1 shows the block diagram of the proposed method in calculating the ISM.

The central part in Eq.(1) is to compute the conditional probabilities, we estimate their density function by using kernel density estimation (KDE) method, the multivariate kernel estimator is adopted and defined as:

$$\hat{f}(y) = \frac{1}{N}\sum_{i=0}^{N-1} K_H(y - y_i) \quad (2)$$

where the kernel $K_H(y) = \|H\|^{-1/2} K(H^{-1/2}y)$, H is the bandwidth matrix which specifies the spread of the kernel around sample y_i, y_i is the first q principal component of x_i. We consider the bandwidth matrix H as a function $H(y_i)$ of the sample point y_i. So different samples should have kernels with different sizes. $H(y_i)$ is then calculated [11],

$$H(y_i) = D_{KL}(\hat{f}||\hat{f}_i) \cdot I \quad (3)$$

where $D_{KL}(\hat{f}||\hat{f}_i)$ is the kullback-Leibler divergence between distribution $\hat{f}_i$ and the candidate distribution $\hat{f}$.

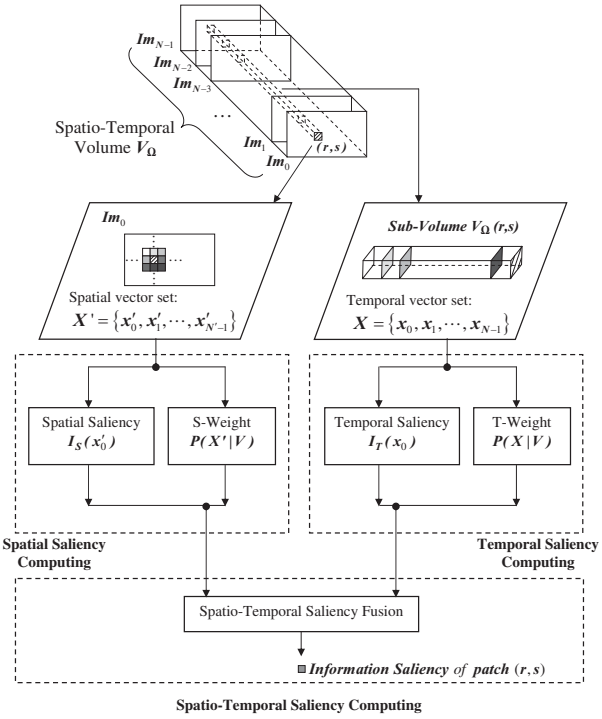


Figure 1. Flowchart of Information Saliency Map (ISM) computing from one single patch. It mainly contains three parts, namely spatial saliency computing, temporal saliency computing and spatio-temporal saliency computing

We have shown that by using this method for density function estimation, we can detect salient motions while robust to illumination changes[11]. Then a Gaussian kernel is used and the density estimator in Eq.(2) becomes Eq.(4) and can be solved.

$$\hat{f}(y) = \frac{1}{(2\pi)^{q/2}N}\sum_{i=0}^{N-1}[(D_{KL}(\hat{f}||\hat{f}_i))^{-q/2} \cdot \exp(-\tfrac{1}{2}(y-y_i)^T(D_{KL}(\hat{f}||\hat{f}_i))^{-1}I)(y-y_i))] \quad (4)$$

We will use model Eq.(1) to detect and analyze human motions, because the temporal information is more important to represent the motion, so we simplify this model by setting $P(x_0|X') = 0$, then we get

$$I_{r,s} = -\log_2\left(\frac{P(x_0|X)}{1-P(x_0|X)}\right) \quad (5)$$

We will use Eq.(5) to calculate the ISM, we consider the patch saliency is only dependent on the temporal motion saliency computed from sub-volumes.

2.2. Constructing Salient Action Unit from ISM

In this paper, we define the repeated motion atomic as an action unit. Because repeated motions show similar underlying shape in the spatio-temporal volume, pixel intensities

under motion will also show similar variations. This implies pixels on an object follow similar distribution functions under repeated conditions. Along this line, the whole object information saliency, which sum up object pixels' information saliency, will also shows this repetition property.

We define the periodic action unit that shows distinct characteristics of a human action as a Salient Action Unit (SAU). The SAU contains fundamental information of different kinds of human actions and they are easier to be classified. The construction of SAU is illustrated in Figure 2. Figure 2(a) shows a person sliding from the right part of the scene to the left part of the scene, with his two legs jointing and separating for several times. In this case, we consider the legs jointing and separating for one time as an action unit. Figure 2(b) shows the object information saliency during this period while the man is moving in the scene. Figure 2(c) shows the moving person's overall information saliency curve, each value on the curve is according to the summation of information saliency from all human body parts. It is clear that there are some local maximum points such as $\{b1, b3\}$ and some local minimum points such as $\{b2, b4\}$ on this curve. These points imply motion corners, which means the person's motion changes at these positions. There are repeated action units in this video clip. We define the interval between two local minimum as a SAU which has a local maximum. It represents the basic information atomic in the whole video. To extract SAU, all the ISM(s) between local minimum points are merged into one single context as shown in Figure 2(d), using its bounding regions to crop the original frames and concatenating all the cropped regions into one volume. The SAU will be generated, as shown in Figure 2(e). Figure 3 shows the totally ten kinds of SAU extracted from database [17].

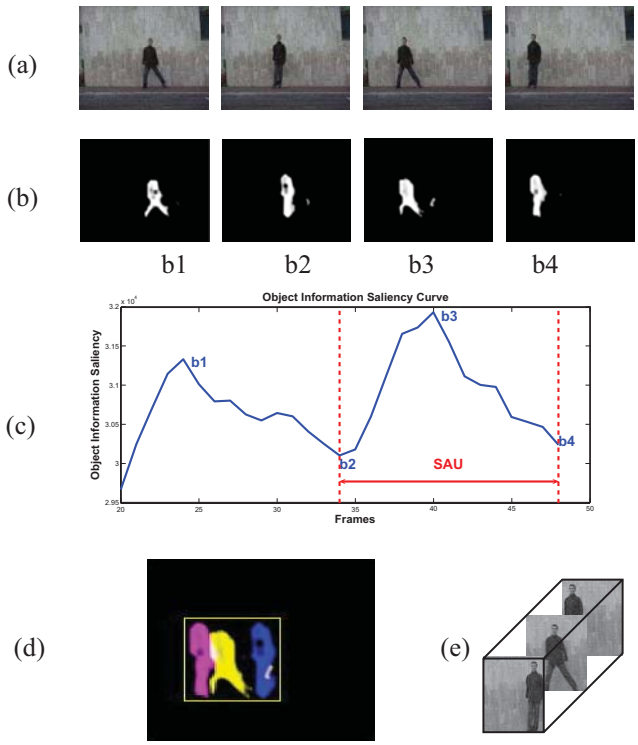


Figure 2. (a) Original video frames, they are from "*eli_side*" video clip in [17] database at frame $23, 30, 38, 45$. (b) Information Saliency Map. (c) Object information saliency curve, the four points $b1, b2, b3$ and $b4$ are according to the four frames, the period between local minimum b2 and b4 is called a Salient Action Unit (SAU). (d) Accumulation of ISM in the SAU between the 16 frames from $b2$ to $b4$, here we show ISM from three frames $b2, b3$ and $b4$. (e) SAU volume cropped from the original video

3. Boosting EigenActions

It has been found the repetitive patterns of human motions have distinct signatures in some other feature space. Some researchers extract frequency domain signatures to estimate the periodic motions in the spatial domain [4]. In this paper, we employ eigen space for human periodic motion representation because the eigen space contains more underlying information from sample data. Then we train an Adaboost classifier to categorize human actions in the eigen space.

3.1. EigenActions Representation

The eigen-based technique has shown good performance in face recognition. Inspired from the truth that EigenFaces show general facial patterns from faces, we extract EigenActions from the SAU. EigenActions will show more underlying information from actions. To obtain the EigenActions, we take the SAU that has been obtained from Figure 2(e) and make temporal subsampling and spatial normalization to the action unit, in order that all SAUs have the same spatio-temporal size and they are comparable to each other. In particular, as SAU may have different number of frames, caused by different action speeds or frequencies, we select the same number of key frames from all SAUs by temporal subsampling in order that all SAUs have the same temporal length. Then we normalize each of the totally m frames to $l \times l$ size using cubic interpolation method. Therefore, each SAU can be represented by a normalized $l \times l \times m$ volume.

To obtain the EigenActions from a normalized SAU, we employ the traditional principle component analysis approach. Let the whole set of normalized SAUs be $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_M\}$, where Γ_i is an action unit column vector with dimension of $l^2 m$ by 1, M is the number of SAUs, the average human action is then computed as $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$, the covariance matrix C is calculated as $C = \frac{1}{M}\sum_{i=1}^{M}(\Gamma_i - \Psi)(\Gamma_i - \Psi)^T$. Then the eigenvectors which have corresponding highest eigenvalues are the EigenActions which will be used to project the sample vectors to training sam-

ple space for training an Adaboost classifier. Figure 4 shows the first five EigenActions from totally 90 actions in [17] database.

3.2. Adaboost Classifier

Adaboost is an adaptive algorithm to combine a collection of weak classifiers to form a strong classifier, where the weights are dynamically updated according to the errors in previous learning, so that a wrongly classified sample will more likely to be correctly classified in the next learning process. The vital part in Adaboost classifier training is to construct the weak classifier. We employ Bayes classifier which has been proved to yield minimum error rates when the underlying density function is given. The maximum a posteriori (MAP) decision is

$$\begin{aligned} \omega_i^* &= \arg max_i P(\omega_i|x) \\ &= \arg max_i \frac{P(\omega_i)p(x|\omega_i)}{p(x)} \end{aligned} \tag{6}$$

The MAP decision rule shows that sample x is classified to ω_i of whom the posteriori probability given x is the largest among all the other classes.

Because there are often not enough samples to estimate the conditional PDF for each class, Liu and Wechsler [10] proposed a Probabilistic Reasoning Model (PRM), they gave a compromise to assume the within class densities can be modelled as Gaussian distributions, and assume the within class covariance matrices are identical and diagonal. This model combines PCA and the Bayes classifier, it shows good ability to find meaningful patterns in spaces of very high dimensionality. By using this model, the Gaussian model (μ_i, Σ_i) can be estimated as follows,

$$\begin{aligned} \mu_i &= \frac{1}{N_i}\sum_{j=1}^{N_i} x_j^{(i)} \\ \Sigma_i &= diag\{\sigma_1^2, \sigma_2^2, ..., \sigma_m^2\} \end{aligned} \tag{7}$$

where N_i is the number of samples in the ith class, $x_j^{(i)}$ is the jth sample from class ω_i. σ_i^2 is then estimated,

$$\sigma_i^2 = \frac{1}{L}\sum_{i=1}^{L}\left\{\frac{1}{N_i-1}\sum_{j=1}^{N_i}(x_{jk}^{(i)} - \mu_{ik})^2\right\} \tag{8}$$

where $x_{jk}^{(i)}$ is the kth element of sample $x_j^{(i)}$, and L is the number of classes.

The EigenActions obtained in the last section are then used for projecting the original sample vectors to the training sample space, with Bayes classifier working as the weak classifier. These weak classifiers is proved to be more flexible to make predictions and they can make useful contributions to the strong classifier even their weak hypothesis predict the correct label with probability less than 1/2 [10]. We then employ the Adaboost.M2 [5] algorithm for training the action recognition system. Adaboost.M2 shows good performance in multi-class classification problems, but this algorithm has not yet been employed to classify human actions. The detailed algorithm is shown in Algorithm 1.

Algorithm 1 Adaboost.M2 [5]

Input: (1) M labelled examples $< (x_1,y_1),...,(x_M,y_M) >$, $y_n \in \{1,...,L\}$; (2) Distribution D over the examples; (3) Bayes weak Learner; (4) Number of iterations T

Initialize the weight vector: $w_{n,y}^1 = D(n)/(k-1)$ for $n = 1,...,N, y \in Y - \{y_n\}$

Do for t=1,2,...,T

1. Set $W_n^t = \sum_{y \neq y_n} w_{n,y}^t, q_t(n,y) = \frac{w_{n,y}^t}{W_n^t}$ for $y \neq y_n$ and set $D_t(n) = \frac{W_n^t}{\sum_{n=1}^{N} W_n^t}$
2. Set $m_{k,n}$ the kth element of M_i, and $M_i = \frac{1}{N_i}\sum_{n=1}^{N_i} x_n$ $h_t = \left\{ min_i \left(\sum_{j=1}^{l^2 m} \frac{(x_{n,j} - m_{i,j})^2}{\sigma_j^2} \right) \Rightarrow x_n \in \omega_i \right\}$
3. Calculate the pseudo-loss of h_t: $\varepsilon_t = \frac{1}{2}\sum_{n=1}^{N} D_t(n)\left(1 - h_t(x_n,y_n) + \sum_{y \neq y_n} q_t(n,y) h_t(x_n,y)\right)$
4. Set $\beta_t = \varepsilon_t/(1-\varepsilon_t)$
5. Set the new weights vector to be $w_{n,y}^{t+1} = w_{n,y}^t \beta_t^{(1/2)(1+h_t(x_n,y_n)-h_t(x_n,y))}$ for $n = 1,...,N, y \in Y - \{y_n\}$

Output the hypothesis

$$h_f(x) = \arg max_{y \in Y} \sum_{t=1}^{T} \left(log \frac{1}{\beta_t} \right) h_t(x,y)$$

4. Experimental Results

Our proposed method is evaluated with human action database [17] which contains 90 low-resolution (180×144, 50 fps) video sequences with nine people, each performing 10 natural actions: 'run', 'walk', 'skip', 'jumping-jack' ('jack'), 'jump-forward-on-two-legs' ('jump'), 'jump-in-place-on-two-legs' ('pjump'), 'galloping-sideways' ('side'), 'wave-two-hands' (or 'wave2'), 'wave-one-hand' (or 'wave1'), and 'bend'. First, we generate the ISM for the 90 video clips (temporal window size=20, patch size=4), then we extract SAUs by finding local minimum points from the information saliency curves, after temporal subsampling and spatial normalization, we get the normalized SAUs, a set of these results are shown in Figure 3.

Based on the normalized SAU, principle component analysis is employed to obtain a set of EigenActions, see Figure 4 for example, these EigenActions contain general patterns of 10 different actions, and they are considered to be basis vectors for the real actions.

We use the EigenActions to project selected normalized salient action unit to the training sample space, then these

Figure 3. Salient Action Units, totally ten kinds of actions, first column: *bend*, *jump*, *run*, *skip*, *wave*1, second column: *jack*, *pjump*, *side*, *walk*, *wave*2. The six frames for each action are combined into one single image in one row, and they are considered as the Salient Action Unit

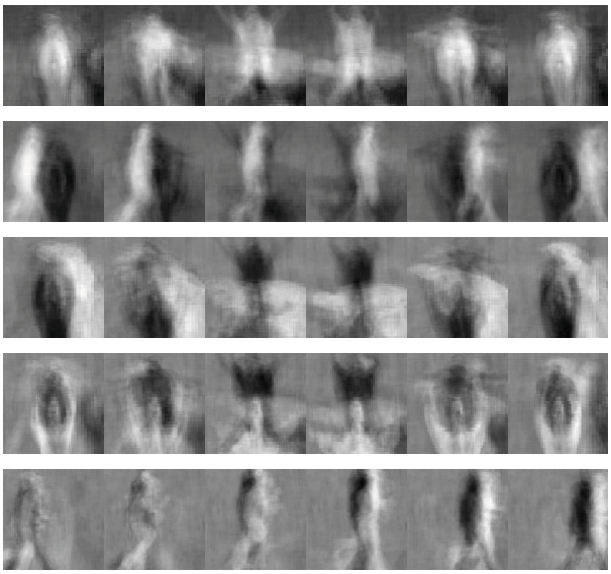


Figure 4. The top five EigenActions extracted from totally 90 actions in database [17], in which there are totally ten kinds of actions.

samples are used to train an Adaboost classifier for categorizing new actions. To test the performance of the proposed classifier, we employ the leave-one-out scheme for evaluation, by randomly taking one person (totally 10 actions) as the testing data, and the other 8 persons for testing. The total boosting round is chosen to be 50. The confusion matrix of our method is shown in Figure 5(a), our algorithm can correctly categorize 98.3% of the testing videos.

We find that most errors are caused by the three actions of "*skip*","*jump*","*run*". This is because these three actions have both temporal and spatial similarities. Two latest methods which also extract action information from the spatial-temporal volume for action recognition are also selected for comparison. It shows that our proposed method outperforms 72.8% correct rate in the Hierarchical Model

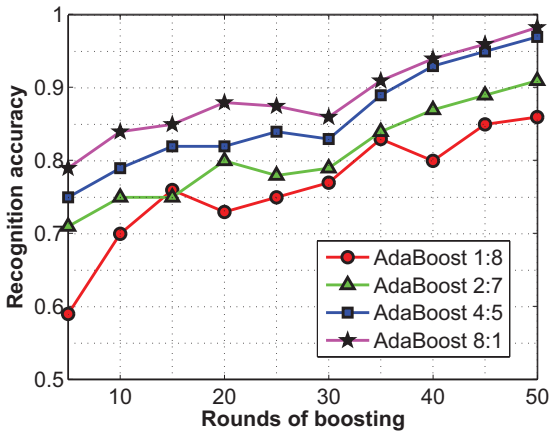


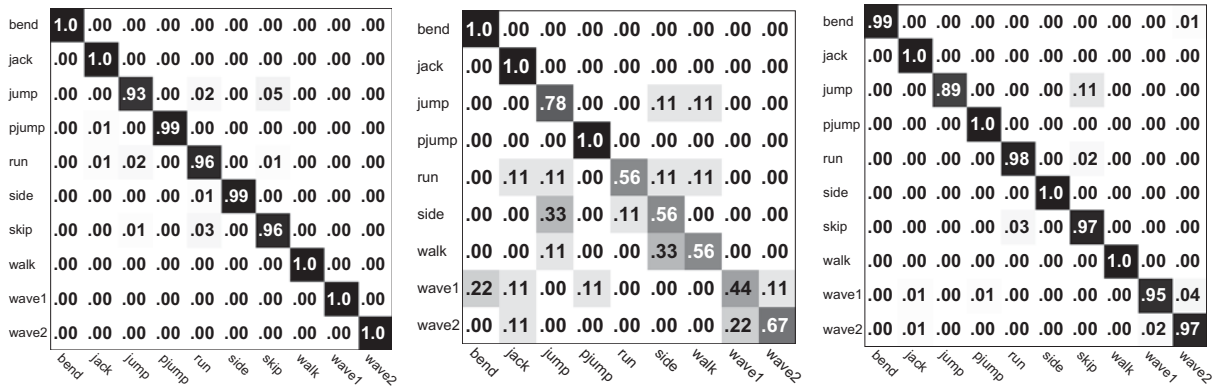
Figure 6. The recognition accuracy of Adaboost using different number of EigenActions, "Adaboost 1:8" is to use EigenActions from one person for training and the other 8 persons for testing.

[8] and 97.5% correct rate in the Space-Time Shape Model [6].

One advantage of our proposed method is the action recognition accuracy will increase with more boosting rounds and more training samples. This can be illustrated from Figure 6. The four curves show the change of recognition accuracy with the increasing of boosting rounds. Each curve represents one experimental setting of the proportion of the training sample number and testing sample number. The result will be more stable when the training samples contains at least $1/3$ of total samples, and when boosting round is larger than 30. Another advantage of our proposed method is the high efficiency to categorize human actions. Considering 4 persons for training (40 videos) and 5 persons for testing (50 videos), the training time is 8 minutes and the testing time (including constructing SAU and action classification) is 10 seconds. Our method shows higher efficiency than spatio-temporal feature matching method reported in [13].

5. Conclusions and Future Works

A novel human action categorization method by boosting EigenActions is proposed in this paper. First, the salient action unit is extracted from object information saliency curve, the salient action unit can be seen as an atomic action which is repeated for times and combined with other atomic actions to form dynamic human actions. The informative EigenActions are then extracted and used to project original action samples to another action space where classification is easier to achieve. Experimental results show that our proposed method is accurate, efficient and flexible, and no human tracking, or prior knowledge about the background is needed.



(a) (b) (c)

Figure 5. Confusion matrix obtained from leave-one-out scheme, (a) Our proposed method. 98.3% samples are correctly categorized. (b) the Hierarchical Model [8]. 72.8% samples are correctly categorized. (c) the Space-Time Shape Model [6]. 97.5% samples are correctly categorized. For all the three confusion matrixes, horizontal lines are ground truth, and vertical columns are predicted labels. The intensity level of each block implies the confusion matrix element value.

Our future work will be concentrated on exploring the correlation between action learning and saliency based action segmentation, if these two parts can collaborate with each other, the whole system performance will be improved.

Acknowledgment

This project is partially supported by the Faculty Research Grant of Hong Kong Baptist University. The authors would like to thank the Computer Vision Lab in Weizmann Institute of Science for the contribution of the Weizmann human action dataset.

References

[1] M. Ahmad and S. W. Lee. Human action recognition using multi-view image sequences. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 523–528, 2006.

[2] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. *International Conference on Computer Vision*, pages 1–8, 2007.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1395–1402, 2005.

[4] A. Briassouli and N. Ahuja. Extraction and analysis of multiple periodic motions in video sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1244–1261, 2007.

[5] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Lecture Notes In Computer Science*, 904:23–37, 1995.

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2053, 2007.

[7] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on SMC*, 34(3):334–352, 2004.

[8] C. N. Juan and F.-F. Li. A hierarchical model of shape and appearance for human action classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[9] I. Laptev. On space-time interest points. *International Journal on Computer Vision*, 64(2-3):107–123, 2005.

[10] C. Liu and H. Wechsler. Robust coding schemes for indexing and retrieval from large facedatabases. *IEEE Transactions on Image Processing*, 9(1):132–137, 2000.

[11] C. Liu, P. C. Yuen, and G. P. Qiu. Object motion detection using information theoretic spatio-temporal saliency. *Submitted to IEEE Transactions on Circuits and Systems for Video Technology*, 2008.

[12] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[13] E. Shechtman and M. Irani. Space-time behavior based correlation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):2045–2056, 2007.

[14] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):893–908, 2008.

[15] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 984–989, 2005.

[16] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1530–1535, 2006.

[17] http://www.wisdom.weizmann.ac.il/vision/spacetimeactions.html.