

# 초고속 정보 통신망을 이용한 원격 합주 교육 시스템

한 창 호<sup>†</sup> · 이 경 명<sup>††</sup> · 윤 광 섭<sup>†††</sup> ·  
류 기 홍<sup>††††</sup> · 모 종 식<sup>††</sup> · 김 유 성<sup>†</sup>

## 요 약

컴퓨터 및 미디 입력 장치의 관련 기술 발달로 인해 컴퓨터를 이용한 디지털 음악 시스템에 대한 활용 및 연구가 지속되고 있다. 또한, 컴퓨터 통신 기술의 발달로 대량의 멀티미디어 정보를 실시간으로 전송하는 것을 가능하도록 한 초고속 정보 통신망을 컴퓨터 음악과 접목시킨 시스템 개발이 가능하게 되었다. 본 논문에서는 합주 연주자들이 한 장소에 모이지 않고도 합주 연습 교육을 실시할 수 있도록 지원하기 위해 초고속 정보 통신망을 이용한 원격 합주 교육 시스템을 설계하고 개발하였다. 본 연구에서 개발한 원격 합주 교육 시스템은 연주자의 연주음을 MIDI 음으로 인식하고 이를 시스템이 관리하고 있는 원래의 MIDI 악보 데이터와 실시간으로 비교하여 음정 및 박자의 정확도를 평가하여 종합적으로 연주자에게 합주 분석 결과를 제공하는 합주 교육 시스템이다. 개발된 원격 합주 교육 시스템은 초고속 정보 통신망을 갖고 있는 학교 등에서 음악 교육을 위한 지원 시스템으로 사용될 수 있다.

## The Remote Concert Education System on High-Speed Communication Network

Chang-Ho Han<sup>†</sup> · Kyung-Myung Lee<sup>††</sup> · Kwang-Sub Yoon<sup>†††</sup> ·  
Ki-Hong Ryu<sup>††††</sup> · Jong-Sik Mo<sup>††</sup> · Yoo-Sung Kim<sup>†</sup>

## ABSTRACT

Continuing advance in computers and MIDI devices has accelerated research on the computerized music technology. Realization of high speed computer communication networks facilitated on-line computer music systems, which needs to send a volume of multimedia data. This paper presents the design and implementation of the Remote Concert Education System which helps users practice ensemble without gathering in a room. The system maintains the music database, identifies tones and measures of the melody played with different instruments, check the correctness on-line, and finally provides the analysed results of the ensemble. The developed system can be used as a supporting system for music education if high speed communication network is available.

\* 본 연구는 1997년도 정보통신부 초고속정보통신 응용기술개발사업의 지원으로 진행되었다.

† 종신회원 : 인하대학교 전자계산공학과 교수

†† 준 회 원 : 인하대학교 전자계산공학과

††† 정 회 원 : 인하대학교 전자계산공학과 교수

†††† 준 회 원 : 인하대학교 전자공학과

논문접수 : 1998년 12월 11일, 심사완료 : 1999년 3월 11일

### 1. 서 론

컴퓨터의 등장 이후, 미디 입력 장치를 이용한 컴퓨터 음악에 대한 관심이 증대되어 컴퓨터를 이용한 디지털 음악은 꾸준히 발전되어 왔다. 컴퓨터 통신망의 발달로 디지털 데이터의 전송 속도가 높아지면서 컴퓨팅 환경에 많은 변화를 가져왔다. 공간적인 이동 없이 통신을 이용한 원격 화상회의가 가능해졌으며, 또한 직접 병원에 가지않고도 의사가 화면을 통해 환자를 진료할 수 있게 하는 원격 진료가 가능해졌다. 이와 같이 통신망의 발달로 인해 초고속 정보 통신망의 장점을 이용한 많은 응용 프로그램들이 개발되고 있다. 이러한 변화는 컴퓨터 음악 분야에도 영향을 주어 초고속 정보 통신망을 컴퓨터 음악과 접목시키는 시도를 가능하게 했다. 특히 악기 연주 신호는 영상 신호 등에 비교하면 데이터의 양은 적으나 시간적 지연에는 더욱 민감하기 때문에 원격 음악 교육 시스템을 개발하기 위해서는 신속한 데이터 전달이 가능한 초고속 정보 통신망의 이용이 절대적으로 필요하게 된다.

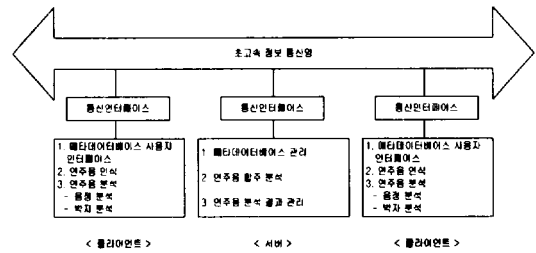
본 연구에서 개발한 원격 합주 교육 시스템은 지역적으로 떨어져 있는 여러 연주자들이 같은 장소에 모이지 않고도 초고속 정보 통신망을 이용하여 효율적으로 합주 연습을 수행할 수 있도록 지원하고자 개발한 교육용 소프트웨어이다. 본 연구에서 개발한 원격 합주 교육 시스템은 연주자의 연주음을 DSP보드를 이용하여 각 음의 기본 주파수와 음의 지속 시간을 알아낸다. 음의 기본 주파수가 음의 높낮이가 되고, 음의 지속 시간이 음의 박자가 된다. 이렇게 음의 높낮이와 박자 정보를 정확히 추출하여 서버의 데이터베이스에 있는 원곡 정보의 음의 높낮이와 박자를 비교 평가하여 평가 결과를 사용자에게 제공한다. 평가를 내릴 때는 각 음 단위로 평가를 내린다. 즉, 입력된 음과 서버의 원음의 시작점과 끝점을 비교 평가하여 그 음의 박자에 대한 평가를 내리고, 입력된 음과 서버의 원음과의 높낮이를 비교 평가하여 그 음의 높낮이에 대한 평가를 내린다. 그래서, 결국에는 이 두 요소, 즉 음의 박자와 높낮이에 대한 각각의 평가값이 그 음에 대한 평가가 된다. 이러한 각 음에 대한 평가를 종합한 값이 전체 연주의 평가가 된다. 그리고, 연주 평가 결과를 지속적으로 보존함으로써 연주자의 취약점 및 개선점을 지적할 수 있는 기능을 갖고 있다. 그래서 특별히 지도 교사의 도움 없이도 효율적으로 연습을 수행

하는데 활용이 가능하다.

본 논문의 구성을 살펴보면, 먼저 2장에서는 본 연구에서 제안한 시스템인 원격 합주 교육 시스템의 전체적인 클라이언트/서버의 구조와 각각의 기능에 대해 기술하고, 3장에서는 본 시스템에서 채택한 음악 정보를 표현하는 파일포맷으로 국제적으로 널리 사용되는 미디 포맷과 미디와 악보 파일을 검색하기 위해 서버에 구축한 메타 데이터베이스 스키마를 기술하고, 4장에서는 클라이언트에서 연주한 음을 DSP 보드를 통하여 인식하는 방법에 대해 기술하고, 5장에서는 클라이언트/서버 간의 통신 메시지 구조를 기술하고, 6장에서는 연주자가 연주한 음을 서버에 구축된 정확한 음악 정보와 비교, 분석과 그 분석 내용을 관리하는 부분에 대해 기술한다. 그리고, 마지막으로 7장에서 본 논문의 결론을 기술한다.

### 2. 원격 합주 교육 시스템의 구조 및 기능

(그림 1)은 원격 합주 교육 시스템의 전체적인 구성을 기능 모듈별로 나타낸 구성도이다. 원격 합주 교육 시스템은 하나의 서버와 초고속 정보 통신망으로 연결된 여러 개의 클라이언트로 구성된다.



(그림 1) 시스템 구성도

서버와 클라이언트를 기능 모듈별로 설명하면 다음과 같다.

- 우선, 서버는 다음과 같은 모듈로 구성되어 있다.
- 메타 데이터베이스 관리 모듈 : 서버에는 미디 음악 파일과 악보 파일을 저장하고 있는데 클라이언트의 연주자가 원하는 곡을 찾기 쉽게 다양한 질의를 사용할 수 있도록 서버는 미디 음악 파일의 많은 메타 정보(곡명, 작곡가명, 작사가명, 가사 등)를 저장 및 관리한다.
- 연주음 합주 분석 모듈 : 2인이 동시에 하나의 악보

상의 다른 성부를 연주할 때에 각 클라이언트에서의 분석 결과를 종합적으로 분석한다.

- 연주음 분석 결과 관리 모듈 : 각 클라이언트에서 연주 분석 결과를 추후의 연주와 비교 분석하기 위한 교육 자료로 사용하기 위해 서버에 데이터베이스화하여 관리한다.

클라이언트는 다음과 같은 모듈로 구성되어 있다.

- 메타 데이터베이스 사용자 인터페이스 : 관계형 데이터베이스 질의어인 SQL을 모르는 일반 사용자가 자신이 연주하고자 하는 곡을 찾기 위한 다양한 메타 정보 질의어를 쉽게 만들 수 있도록 사용자 위주의 인터페이스를 제공한다.
- 연주음 인식 모듈 : 악기의 실제 연주음으로부터 DSP 보드를 통해 음정과 박자 등을 인식하여 분석 모듈에 전달해 준다.
- 연주음 분석 모듈 : 서버에서 전달 받은 정확한 미디 파일과 연주음 인식 모듈에서 전달 받은 실제 악기로 연주한 연주음을 음정과 박자의 두 요소를 가지고 비교, 분석하는 모듈이다.

### 3. 미디와 메타 데이터베이스

#### 3.1 원격 합주 교육 시스템에서의 미디 활용

표준 미디 파일은 시퀀서(sequencer)가 기록하거나 연주하는 데이터를 저장하기 위해 디자인된 파일 포맷이다. 이 포맷은 표준 미디 메시지와 각각의 메시지에 대한 타임스탬프(메시지간의 시간을 나타내는 바이트들)를 저장한다. 미디 파일은 파일 포맷을 설명하는 헤더청크와 하나 이상의 트랙 청크를 포함한다. 청크는 관련된 바이트들의 그룹이다. 헤더 청크(MThd chunk)는 미디 파일 타입과 파일에 저장된 트랙수와 타임스탬프의 기본이 되는 4분음표당 펄스수와 같은 내용을 담고 있다. 트랙 청크(MTrk chunk)에는 시간을 나타내는 타임스탬프와 모든 미디 데이터와 비미디 데이터를 포함한다([1],[2]). 실제의 음악 정보는 트랙 청크에 들어 있다.

미디 파일은 여러 개의 트랙으로 구성되어 있다([4]). 보통 한 트랙은 하나의 채널의 미디 데이터이고 하나의 악기가 배당된다. 본 연구에서는 각 트랙은 한 사람이 연주할 수 있는 하나의 악보가 되기 위해서 한 트랙을 정확히 하나의 채널로 사용되는 타입 1을 사용한다. 미디 파일은 여러 개의 청크로 구성되어 있는데

각 청크는 여러 개의 미디 트랙들의 정보를 가지고 있다. 그렇기 때문에 하나의 트랙을 만들기 위해서 전체의 파일을 검색하면서 같은 트랙의 정보만을 모아야 한다. 미디 메시지도 실제 미디 파일에 저장된 메시지와 다른 형식을 가진다. 미디 메시지의 길이와 형식이 다르기 때문에 적절하게 변형시켜야 한다. 또한, 미디 파일에는 메타 정보도 포함되므로 이것은 실제 미디 장치에 보낼 때는 필요가 없기 때문에 무시해야 하고 장치에 명령을 하는 정보, 즉 템포라든지 볼륨 조절 등은 명령해 주어야 한다. 파일의 메시지를 변환시킬 때 중요한 것은 그것의 포맷 길이와 델타 타임이라는 시간을 변환시키는 것이다. 델타 타임은 트랙 메시지 중 전 메시지와 다음 메시지 사이의 시간 간격을 말하는데, 실제 연주를 하기 위해선 약간의 변환이 필요하다. 이렇게 변환시키면 미디 메시지를 리스트로 갖는 연주 가능한 트랙을 얻게 된다.

하나의 미디 파일에는 여러 개의 트랙이 있어서 미디 파일을 연주하기 위해 여러 트랙들을 동시에 하나의 시간에 맞추어 읽어 나가야 한다. 그래서 시간에 맞추어 미디 출력 장치에 미디 메시지를 출력하면 비로소 음이 출력된다. 미디 파일에 있는 시간에 대한 정보는 델타 타임밖에 없기 때문에 다중 트랙을 연주하는데 있어서 다음 음으로 어느 트랙의 음을 출력해야 하는지 판단하기가 곤란할 수도 있다. 그래서 한가지 방법으로 절대적인 시간으로 계산하여 트랙을 기록해 놓으면 내부 타이머의 시간으로 변환하면 쉽게 해결될 것이다. 즉, 한 트랙에서 정보를 읽을 때마다 델타 타임을 계속 더해져서 그것을 저장하면 쉽게 변환될 것이다.

본 연구에서는 DSP보드를 이용하여 마이크 입력으로부터 음을 추출하도록 하였다. 초기에 DSP보드에 프로그램을 적재시킨다. 그리고, I/O포트를 사용하여 적절한 명령을 주면 프로그램이 적재되어 입력이 있을 때마다 일정 메모리 영역에 정보를 처리하여 기록하게 된다. 클라이언트 프로그램에서는 계속적으로 입력을 기다린 다음 입력이 있으면 적절한 형태의 미디 포맷으로 변환시킨다. 변환된 데이터는 미디 메시지와 같은 데이터이다. 장치로부터 음의 시작을 알리는 정보, 음의 높낮이, 음의 끝을 알리는 정보 등의 세가지 정보가 입력된다. 미디 메시지로 변환시키기 위해서는 내부적인 타이머로부터 밀리초 단위로 측정하여 음의 시작과 음의 끝을 측정한다. 그리고 음의 높낮이에 일

정한 값을 더하여 미디 메시지 형태로 변환시킨다. 이렇게 변환된 미디 메시지는 NOTE ON과 NOTE OFF 정보이다.

(그림 2)는 하드웨어 장치로부터 입력된 정보를 미디 포맷으로 변환하는 과정을 설명하고 있다. (그림 2)의 변환 과정은 계속적으로 장치입력의 내용을 미디 메시지로 변환하는 기능으로 구성된다.

```

MidiInput()
{
    장치 초기화
    while ( 외부의 종료 신호가 있을 때까지 )
    {
        // 입력을 기다린다
        recording ( RECORDING );
        if ( NOTE ON Message 일 경우 )
        {
            NOTE ON에 대응되는 Midi Message로 변환;
        }
        else if ( NOTE OFF Message 일 경우 )
        {
            NOTE OFF에 대응되는 Midi Message로 변환;
        }
        Midi 장치로의 출력;
        Network로의 전송;
    }
}
    
```

(그림 2) 하드웨어 장치로부터 입력된 정보의 미디 포맷으로의 변환

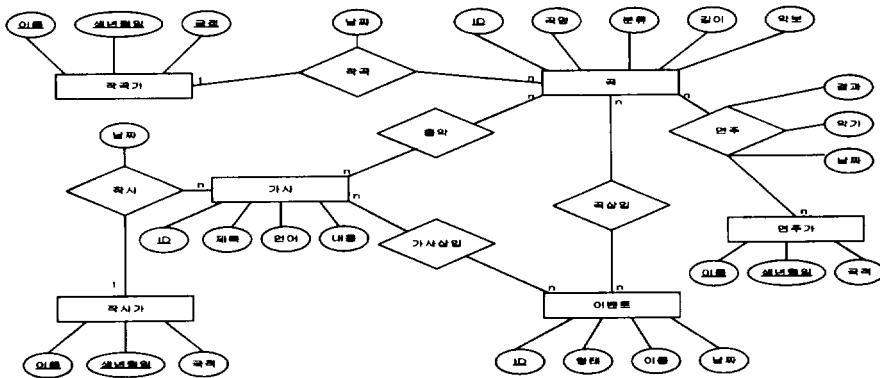
3.2 메타 데이터베이스

원격 합주 교육 시스템에 있어서 메타 데이터베이스 스키마의 E-R 다이어그램([6],[7])은 (그림 3)과 같다. 6개의 개체 집합과 6개의 관계 집합으로 구성된다.

각각에 대해 자세하게 설명하겠다.

3.2.1 개체 집합

- 작곡가 개체 집합 : 작곡가와 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 이름, 생년월일, 국적이며, 기본키로는 이름과 생년월일의 복합키가 사용된다. 동명이인이 있기 때문에 주요키로 이름과 생년월일의 복합키를 사용하였다.
- 곡 개체 집합 : 작곡가가 작곡하는 곡과 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 ID, 곡명, 분류, 길이, 악보이며, 기본키는 ID이다. 분류 속성은 가요, 동요, 클래식, 팝, 복음성가, 기타등으로 구분하였다. 길이의 단위는 곡의 마디이다. 악보 속성은 악보 파일의 위치를 나타낸다.
- 작사가 개체 집합 : 작사가와 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 이름, 생년월일, 국적이며, 기본키로는 이름과 생년월일의 복합키가 사용된다. 동명이인이 있기 때문에 기본키로는 이름과 생년월일의 복합키를 사용하였다.
- 가사 개체 집합 : 작사가가 작사한 가사와 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 ID, 제목, 언어, 내용이며, 기본키는 ID이다. 언어 속성은 쓰여진 언어를 말하며, 내용 속성은 실제 가사의 내용을 나타낸다.
- 이벤트 개체 집합 : 이벤트와 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 ID, 형태, 이름, 날짜이다. 기본키는 ID이다. 형태 속성은 이벤트의 형태를 나타낸다. 예를 들면, 영화, 연극, 광고 등이 된다. 이름은 그 이벤트의 이름이 되고, 날짜 속성



(그림 3) 메타 데이터베이스의 E-R 다이어그램

은 이벤트가 일어난 날짜를 나타낸다.

- 연주가 개체 집합 : 연주가와 관련된 정보를 저장하는 개체 집합이다. 구성된 속성은 이름, 생년월일, 국적이며, 기본키로는 이름과 생년월일의 복합키가 사용된다. 동명이인이 있기 때문에 기본키로 이름과 생년월일의 복합키를 사용하였다.

### 3.2.2 관계 집합

- 작곡 관계 집합 : 어떤 곡이라도 언제 누구에 의해 작곡되었을 것이다. 이와같은 정보를 나타내어 주기 위해서 작곡 관계 집합이 필요하다. 즉, 작곡가 개체 집합과 곡 개체 집합의 관계를 나타내는 관계 집합으로써, 날짜라는 하나의 속성을 가지고 있다. 그 의미는 "언제 어떤 작곡가가 어떤 곡을 작곡하였다"이다.
- 작사 관계 집합 : 가사도 동일하게 작사가에 의해 작사되었을 것이다. 이러한 관계 정보를 나타내어 주는 관계 집합이 필요하다. 작사 관계 집합이 이와 같은 역할을 할 것이다. 즉, 작사가 개체 집합과 가사 개체 집합의 관계를 나타내는 관계 집합으로써, 날짜라는 하나의 속성을 가지고 있다. 그 의미는 "언제 어떤 작사가가 어떤 가사를 작사하였다"이다.
- 가사삽입 관계 집합 : 어떤 가사는 어떤 이벤트, 즉, 영화나 연극, 광고 등에 삽입될 수도 있을 것이다. 이러한 관계를 나타내어 주는 관계 집합이 필요한데, 이와같은 역할을 가사삽입 관계 집합이 담당한다. 즉, 가사 개체 집합과 이벤트 개체 집합의 관계를 나타내는 관계 집합으로써, 어떤 가사가 어느 이벤트에 삽입되었는지를 나타내어 준다.
- 곡삽입 관계 집합 : 어떤 곡이 어떤 이벤트에 삽입될 수도 있는데, 이와같은 관계를 나타내어 주는 관계 집합이 필요한데, 곡삽입 관계 집합이 이를 담당한다. 즉, 곡 개체 집합과 이벤트 개체 집합의 관계를 나타내는 관계 집합으로써, 어떤 곡이 어느 이벤트에 삽입되었는지를 나타내어 준다.
- 연주 관계 집합 : 어떤 곡이라도 곡을 만든 이유는 연주를 하기 위해 만들 것이다. 곡이 연주되기 위해서는 연주가 필요한데, 이같이 연주가 개체 집합과 곡 개체 집합의 관계를 나타내어 주는 관계 집합이 필요한데, 연주 관계 집합이 이를 담당한다. 구성된 속성은 날짜와 악기, 연주 후의 결과물을 나타내는 결과로 구성된다.

- 음악 관계 집합 : 음악이라고 하는 것은 선율 정보인 곡과 가사가 합쳐진 것이라고 할 수 있다. 이와 같은 정보를 나타내는 것이 음악 관계 집합이다. 음악 관계 집합은 곡 개체 집합과 가사 개체 집합의 관계를 나타내는 관계 집합이다.

실제 구현에 있어서는 효율성면에 살펴보면 위에서 설명한 스키마는 약간의 변형이 필요할 것이다. 이러한 작업을 역정규화(denormalization) 과정이라 한다 ([6],[8],[9]). 본 연구에서는 역정규화 과정을 거쳐 실제 구현에서 필요한 테이블의 수를 (그림 4)와 같이 5개로 설계하였다. 즉, 작곡과 관련된 부분을 나타내는 작곡 Table, 작사와 관련된 부분을 나타내는 작사 Table, 이벤트와 관련된 부분을 나타내는 이벤트 Table, 연주와 관련된 부분을 나타내는 연주 Table, 마지막으로 음악과 관련된 부분을 나타내는 음악 Table 등 5개의 기본테이블로 구성된다.

작곡TABLE	ID	악보길이	장조	작곡가명	악보	박자	빠르기	날짜
작사TABLE	ID	언어	작사가명	날짜	가사			
연주TABLE	ID	연주시간	연주가명	미디어정보	연주악기	날짜		
이벤트TABLE	ID	TYPE	이름	날짜				
음악TABLE	ID	분류	곡명					

(그림 4) 원격 합주 교육 시스템의 메타데이터베이스 스키마

## 4. 연주음 인식

### 4.1 하드웨어 소개

초고속 정보 통신망을 이용하여 합주 연습시 연주자들이 한 장소에 모이지 않고도 연습을 효율적으로 진행할 수 있는 초고속 정보 통신망용 원격 합주 연습 교육 시스템을 구현하기 위해서는 마이크를 통해 연주자의 연주를 입력으로 받아 음의 시작을 알리는 정보, 음의 높낮이, 음의 끝을 알리는 정보 등의 세가지 정보를 추출하는 음정, 박자 인식 하드웨어가 필요하다.

지역적으로 떨어져 있는 여러 연주자들이 같은 장소에 모이지 않고도 효율적으로 합주 연습을 수행할 수 있도록 하기 위해서는 음정, 박자의 인식 속도가 빨라야 한다. 음정, 박자 인식 하드웨어는 고속의 신호

처리를 가능하게 하는 TMS320C30 DSP보드를 이용하여 구현되어 음정, 박자의 인식을 빠르게 수행한다([12]). TMS320C30 DSP보드는 TMS320C30 C소스 디버거를 사용하여 응용 프로그램을 실행하고 디버깅할 수 있도록 해주는 개발 도구이다([3],[13]). 마이크를 통해 시스템에 아날로그 입력되면, TMS320C30 DSP보드의 16비트 통신 포트를 통하여 호스트 PC로 전달될 수 있다. 또한, 음정 인식에 사용된 고속 푸리에 변환 알고리즘은 이산 푸리에 변환을 고속으로 계산하는 방법인데, 신호의 주파수 해석을 빠르게 수행할 수 있도록 한다([5],[14]).

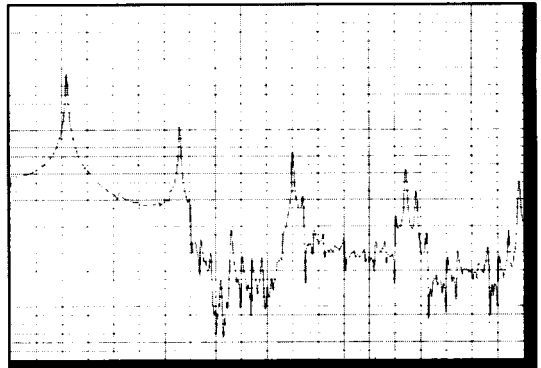
TMS320C30 DSP보드는 마이크를 통해 연주자의 연주를 직접 입력 받아 고속의 신호처리를 통해 음정, 박자 정보를 빠르게 인식할 수 있으므로, 원격 합주 연습 교육을 실시간으로 진행하는 것을 가능하게 한다.

4.2 음정 인식

각 음정은 고유주파수를 지니고 있다. 따라서 음정의 주파수를 파악하면 음정을 인식할 수 있다. 음정의 주파수를 파악하기 위한 방법으로 주파수 계수기를 통해 파형의 개수를 인식하는 방법이 있다. 마이크를 통해 전기적으로 변환된 아날로그 음정 신호의 펄스 개수를 주파수 계수기를 통해 1초동안 계수하면 그 값이 음정 신호의 고유주파수이다. 그러나, 음정 신호의 파형이 일정한 정형파의 형태를 가지지 못하므로 주파수를 정확하게 계수하기는 어렵다. 또한 음정 인식을 위해 음정 신호의 파형을 1초동안 계수한다면 음정 인식의 응답 시간이 1초가 된다. 이 응답 시간은 실시간 합주 평가에는 적합하지 못하다. 다른 문제는 1초 이하의 음 길이를 가지는 음정의 경우에는 계수된 값이 바로 그 음정 신호의 고유주파수가 되지 않는다는 것이다. 이런 경우에는 계수된 값에서 음정 신호의 고유주파수를 추출하기 위해서 추가적인 연산이 필요하게 된다. 이런 문제들 때문에 음정 펄스의 계수를 통한 고유 주파수 인식은 어렵다([10],[11]).

(그림 5)는 다장조 라(440Hz) 음정 신호의 주파수 스펙트럼이다. 음정 신호의 주파수 스펙트럼을 분석해보면 기본 주파수 스펙트럼 외에 고조파의 주파수 스펙트럼과 잡음의 주파수 스펙트럼이 같이 존재하는 것을 알 수 있다. 그러나 기본 주파수 스펙트럼에 에너지가 집중되어 있으므로 신호의 음정을 인식할 수 있다. 주파수 스펙트럼의 분석을 통한 음정 인식 방법은 음정

펄스 계수를 통한 음정 인식 방법에 비해 세 가지 장점을 지닌다. 첫째, 고속 디지털 신호 처리 칩을 이용하여 고속 푸리에 변환을 수행함으로써 음정 펄스 계수 방법에 비해 짧은 응답 시간 내에 음정을 인식하는 것이 가능하다. 즉 주파수 스펙트럼의 분석을 통한 음정 인식 방법은 본 연구에서 목표하는 실시간 합주 평가에 적합하다. 둘째, 주파수 스펙트럼의 분석을 통한 음정 인식 방법은 1초 이하의 음 길이를 가지는 음정의 경우에도 음정을 인식하는 것이 가능하다. 셋째, 음정 신호에 고조파와 잡음이 존재하는 경우에도 음정 인식이 가능하다. 본 연구에서는 고속 푸리에 변환을 통해서 주파수 스펙트럼 값을 얻은 후에 가장 큰 값을 찾아 음정을 인식하는 방법을 사용하였다([5],[14]).



(그림 5) 음정 신호의 주파수 스펙트럼

음의 높·낮이는 일정한 단일 시간 내의 주파수에 의해서 정해진다. 주파수가 많고 진동이 빠르면 높은 음이 되고, 주파수가 적고 진동이 느리면 낮은 음이 된다. 1초간의 진동수는 그 단위를 헤르츠(Hz)로 표시한다. 사람의 귀로 들을 수 있는 가청한계는 1초에 약 16~20,000Hz이며, 음악에서 쓰이는 범위는 88건의 피아노 음역은 약30~4,000Hz이고, 피이프오르간이나 피콜로 등은 이 범위를 넘는다. 음악에 쓰이는 음높이를 통일시키기 위해 선정된 진동수를 연주 음고라 한다. 음악을 연주할 때 모든 악기는 이 표준 진동수에 따라 음을 맞춘다. 라 음을 440Hz로 정하여 조율하여 쓴다. 우리나라를 비롯한 거의 모든 나라에서 이 연주 음고를 사용하고 있다.

평균율은 1옥타브를 12개의 평균한 음정으로 분할하고, 이를 반음으로 정하여 음계를 구성한 것이다. 12

평균율은 일정한 악기로 어떠한 조로도 바꿀 수 있는 이점 때문에 전 세계적으로 보급되었다. <표 1>에 평균율로 조율되었을 때의 진동수를 예로 든다.

<표 1> 평균율로 조율되었을 때의 진동수

옥타브	음정	주파수(Hz)
중앙 옥타브	도	261.6255653
	#도	277.182631
	레	293.6647679
	#레	311.1269837
	미	329.6275569
	파	349.2282314
	#파	369.9944227
	솔	391.995436
	#솔	415.3046976
	라	440
	#라	466.1637615
	시	493.8833013

평균율의 표준 주파수는 440Hz(다장조 라)이고, 반음이 올라갈 때마다 1.059463094배 만큼 주파수가 증가한다. 주파수 스펙트럼 값을 얻은 후에 가장 큰 값을 찾아 음정을 인식하기 위해 고속 푸리에 변환을 사

용한다. 마이크를 통해 들어오는 음을 8kHz로 표본화하고, 1024개 단위로 고속 푸리에 변환한다. 고속 푸리에 변환 출력은 0Hz에서 8kHz까지 등간격으로 존재하며, 각 주파수 스펙트럼에 해당되는 주파수는 아래 식 (1)과 같다.

$$\text{frequency of } X(k) = \frac{k \times \text{sampling rate}}{\text{FFTSIZE}} = \frac{8000 \times k}{1024} \quad (1)$$

위의 식에 의해 얻어지는 주파수 스펙트럼 중 가장 큰 값을 구한다. 가장 큰 주파수 스펙트럼에 해당되는 주파수가 음의 높이이다.

음정 인식의 과정은 다음과 같고, (그림 6)에 음정 인식 모듈을 나타내었다.

- ① 입력 신호와 정현파 표, 고속 푸리에 연산을 위한 배열을 초기화한다.
- ② 고속 푸리에 연산시 참조를 위한 정현파 표를 채운다.
- ③ 새로운 데이터가 들어올 버퍼를 준비한다.
- ④ 고속 푸리에 변환을 위한 입력 배열을 채운다.
- ⑤ 고속 푸리에 변환을 수행한다.

```

init_arrays(); /* 배열의 초기화 */
for ( i = 0 ; i < 512 ; i++ ) /* 정현파 표를 채운다 */
    SinTable[i]=sin(i*2*3.141592/1024);
for ( j = 0 ; j < 16 ; j++ )
    {
        wait_buffer(); /* 새로운 데이터를 위한 버퍼 준비 */
        for ( n = 0 ; n < 64 ; n++ ) /* 푸리에 변환을 위한 입력 준비 */
            FFTBufferIn[64*j+n]=(float)intermediate[n];
    }
fft_r(FFT_SIZE,LOG2_SIZE,FFTBufferIn,SinTable,BITREV); /* 고속 푸리에 변환 수행 */
FFTBufferIn[0]=FFTBufferIn[21]*FFTBufferIn[21]+FFTBufferIn[1003]*FFTBufferIn[1003];
...
FFTBufferIn[16]=FFTBufferIn[53]*FFTBufferIn[53]+FFTBufferIn[971]*FFTBufferIn[971];
FFTBufferIn[17]=FFTBufferIn[56]*FFTBufferIn[56]+FFTBufferIn[968]*FFTBufferIn[968];
FFTBufferIn[18]=FFTBufferIn[59]*FFTBufferIn[59]+FFTBufferIn[965]*FFTBufferIn[965];
...
FFTBufferIn[30]=FFTBufferIn[119]*FFTBufferIn[119]+FFTBufferIn[905]*FFTBufferIn[905];
FFTBufferIn[31]=FFTBufferIn[127]*FFTBufferIn[127]+FFTBufferIn[897]*FFTBufferIn[897];
/* 주파수 스펙트럼의 제곱값을 배열에 저장 */
temp=FFTBufferIn[0];
for ( n = 1 ; n < 32 ; n++ )
    {
        if ( temp < FFTBufferIn[n] )
            {
                temp = FFTBufferIn[n]; /* 가장 큰 값을 찾는다(음정 인식). */
                melody = n;
            }
    }
recording(intermediate, RECORDING); /* 인식된 정보를 호스트 PC로 전송 */
    
```

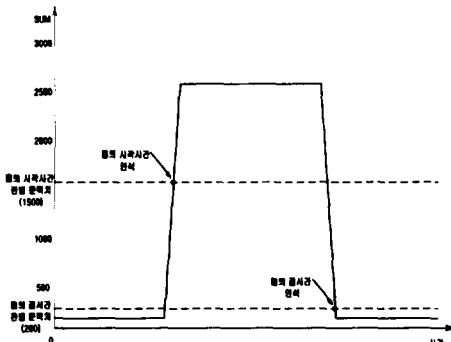
(그림 6) 음정 인식 모듈

- ⑥ 32개의 음정 각각에 해당하는 주파수 스펙트럼의 제곱값을 배열에 저장시킨다.
- ⑦ 32개의 주파수 스펙트럼의 크기를 비교하여 가장 큰 값을 찾는다(음정 인식).
- ⑧ 인식된 음정 정보를 호스트 PC로 전송한다.

4.3 박자 인식

음의 길이는 진동 시간의 길이에 따라 결정된다. 이것은 진동의 지속 시간과 일치하는데 진동 시간이 길면 음이 길고, 그 시간이 짧으면 음이 짧고, 진동이 그치면 음도 소멸한다. 음악에는 여러 가지 박자가 사용되는데, 이는 일정한 시간에 따라 생기는 질서 위에 세워진 박이라는 단위에 따라 만들어진다. 즉, 이 박이 필요에 따라 연결되고 분할됨으로써 음악이 구성된다. 박자를 인식하기 위해서는 음의 시작시간 및 끝시간을 인식할 필요가 있다.

박자를 인식하는 과정은 다음과 같다. 마이크를 통해 들어오는 음을 8kHz로 표본화하여 64개 단위로 입력받아 배열에 저장한다. 그 후, 64개의 표본의 절대값을 취해 더한다. 즉, 다음과 같다.  $SUM = sample[0] + sample[1] + \dots + sample[63]$ . 음이 존재할 때는 그 합이 50~150, 음이 존재하지 않을 때는 2000~3000사이의 값을 가진다. 이 정보를 이용하면 두 개의 문턱치를 설정할 수 있다. 음의 시작시간 판별 문턱치는 1500, 음의 끝시간 판별 문턱치는 200으로 결정하는 것이 타당하다. 64개 표본의 부분합들을 시간의 흐름에 따라 계속 감시한다. 만약 SUM이 계속 증가하는 추세에 있다면, 1500보다 커지는 순간 음의 시작을 인식할 수 있다. 그리고 만약 SUM이 계속 감소하는 추세에 있다면, 200보다 작아지는 순간 음의 끝을 인식할 수 있다. 이를 그림으로 보이면 (그림 7)과 같다. (그림 8)에 음의 시작



(그림 7) 음의 시작 시간 및 끝 시간 인식

시간 및 끝시간 인식 모듈을 나타내었다.

```

note=0; /* 박자 정보의 초기값 설정 */
sum=0; /* 부분합의 초기값 설정 */
for ( n = 0 ; n < 64 ; n++ )
{
    sum=sum+abs(intermediate[n]); /* 절대값을 취한 64개 표본들의 부분합 계산 */
}
if ( note == 0 )
{
    if ( sum > 1500 )
        note=1; /* 음의 시작시간 인식 */
}
recording(intermediate, RECORDING); /* 인식된 정보를 호스트 PC로 전송 */
else if ( note == 1 )
{
    if ( sum < 200 )
        note=0; /* 음의 끝시간 인식 */
}
recording(intermediate, RECORDING); /* 인식된 정보를 호스트 PC로 전송 */
    
```

(그림 8) 박자 인식 모듈

5. 통신

본 연구의 시스템은 서버와 클라이언트간 상호 통신 수단으로 통신 메시지를 통하여 소켓 방식으로 이루어진다. 설계되어진 통신 메시지의 구조는 (그림 9)와 같다. 메시지의 구조를 살펴보면, 4개의 요소로 구성되어 있는데, m\_nMsgType은 메시지의 타입을 나타내고, m\_nUsedType은 메시지의 사용되는 용도를 나타낸다. 메시지 형식에 대한 자세한 내용은 <표 2>에 나타나 있다. 통신 메시지는 서버에서 처리하는 컨트롤 메시지와 클라이언트에서 처리하는 컨트롤 메시지 등이 포함된다.

```

Class Cmessage : public Cobject
{
    // Cmessage Class는 여러 가지 형식의 메시지를 포함한다.
    UINT m_nMsgType; // 메시지의 형식
    UINT m_nUsedType; // 메시지의 사용되는 용도
    BOOL m_bClose; // 프로그램의 종료 여부
    CmessageBody* m_pBody; // 메시지의 실제 정보
}
    
```

(그림 9) 통신 메시지 구조



〈표 2〉 메시지 형식

메시지 형식 이름	설 명
CDWordMsgDouble	Word type의 정보를 포함하는 메시지
CstringMsg	CString type의 정보를 포함하는 메시지
CstrListMsg	CString의 List type의 정보를 포함하는 메시지
CsearchMsg	데이터베이스 검색 정보를 포함하는 메시지
CSResultListMsg	합주 및 독주의 평가결과 정보를 포함하는 메시지
CfileMsg	CFile type의 정보를 포함하는 메시지
CinfoctMsg	Client status 정보를 포함하는 메시지
CanalysisListMsg	합주 및 독주의 연주정보를 포함하는 메시지
CsongsListMsg	곡 리스트 정보를 포함하는 메시지
CresultInfoMsg	종합 평가 또는 개인 정보를 포함하는 메시지

## 6. 연주 결과 분석 및 관리

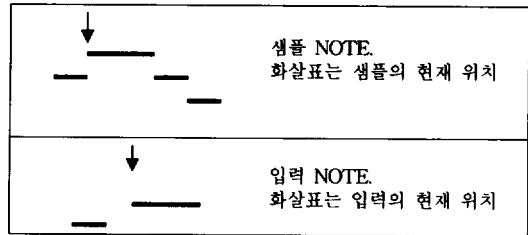
### 6.1 연주 결과 분석

클라이언트에서 연주한 음에 대한 온라인 평가는 클라이언트에서 실시된다. 그리고 합주에 대한 종합적인 평가는 서버에서 행하여진다. 종합적인 평가는 근본적으로 클라이언트에서 실시한 온라인 평가를 바탕으로 하여 평가되어진다. 클라이언트에서 실시하는 온라인 평가 과정에 대해서 더 상세히 기술하면 다음과 같다.

평가는 미디 메시지들을 분석하는 데부터 시작된다. 평가요소는 박자, 음정 등의 두가지인데 입력으로 들어온 값은 NOTE ON과 NOTE OFF에 해당하는 메시지지만 발생하기 때문에 실제 비교하려는 미디 데이터(미디 파일)에서 NOTE ON과 NOTE OFF 정보만 필요로 한다. 미디 파일에는 다른 많은 정보들이 있기 때문에 우리는 NOTE ON과 NOTE OFF정보만 구분하여 하나의 구조를 만든다. 음(NOTE)이란 단위인데, 한 음의 시작, 길이, 음의 높낮이 정보만 갖는다. 그리고 그것들을 시간에 맞추어 리스트로 구성한다. 비교하기 위해서 입력 장치로부터 들어오는 메시지들은 모두 NOTE로 변환된다. 한 음이 올바르게 연주되는지 알기 위해서 단지 비교 샘플(미디 파일을 NOTE 리스트로 변환한 것)의 현재 연주되어야 할 NOTE와 비교해 보면 쉽게 결과를 얻을 수 있다. NOTE들은 선으로 표현할 수 있으며, 세로로 음의 높낮이를 나타내고 가로로 음의 길이와 시작위치를 나타낸다.

입력 장치로부터 변환되어 들어오는 NOTE와 샘플

에서의 현재 음과 차례로 비교하는 과정은 (그림 10)과 같다.



(그림 10) NOTE의 비교

샘플 NOTE 리스트들의 NOTE중 현재 가리키는 NOTE와 입력으로 들어오는 NOTE가 일치할 때 우리는 박자와 음정이 일치하다고 말할 수 있을 것이다. 그리고 높이의 위치가 틀리면 음정이 틀렸다고 말할 수 있고, 길이가 틀리거나 시작 점이 틀리면 박자가 틀렸다고 말할 수 있다. 비교할 때, 매 밀리초 간격으로 비교하기에는 많은 부하가 따른다. 그래서 새로운 NOTE가 발생할 때 마다 비교를 하고 샘플 NOTE를 가리키는 위치를 진행시킨다.

새로운 NOTE가 발생할 때 다음과 같은 경우가 있다.

- 현재 샘플에서의 NOTE가 입력의 NOTE와 일치한다.( 경우 1 )
- 현재 샘플에서의 NOTE보다 입력의 NOTE가 먼저 발생한다.( 경우 2 )
- 현재 샘플에서의 NOTE보다 입력의 NOTE가 나중에 발생한다.( 경우 3 )

샘플의 NOTE 리스트를 진행시키기 위해서 위의 경우 중 (경우1)과 (경우2)에선 샘플의 가리키는 NOTE를 진행시켜 다시 비교를 시작하게 된다. (그림 11)의 예를 통하여 자세히 비교분석과정을 살펴볼 수 있다.

예를 들어 (3. 2)에서 입력이 현재 샘플이 가리키는 NOTE에 비해 더 늦게 발생했기 때문에 다음의 NOTE와 비교하도록 샘플의 NOTE를 진행시킨다. (2. 1)과 (2. 2)의 경우에 있어선 다음의 입력을 기다릴 수 있다. 이런 과정을 통하여 입력 음이 실제 미디 샘플과 비교해서 얼마나 옳고 그른지 체크할 수 있다. 결과적으로 모든 음에 대해서 사용자가 연주한 음이 옳고 그른지 체크한다. 체크한 값은 온라인으로 보여주

며 곡이 끝나면 체크한 값들을 모아서 서버에 넘겨 종합 분석을 할 수 있게 한다.

클라이언트에서의 평가는 얼마나 전체적으로 박자와 음이 잘 연주되었는지 백분율로 나타내고 합주일 경우, 각 클라이언트에서의 평가가 서버에 넘겨져 합주가 얼마나 잘 이루어졌는지 평가하고 개선 사항을 화면에 출력한다.

(1. 1)			
경우 1.			
샘플:	=====		
입력:	=====		
(2. 1)                      (2. 2)                      (2. 3)			
경우 2.			
샘플:	=====	=====	=====
입력:	=====	=====	=====
(3. 1)                      (3. 2)                      (3. 3)			
경우 3.			
샘플:	=====	=====	=====
입력:	=====	=====	=====

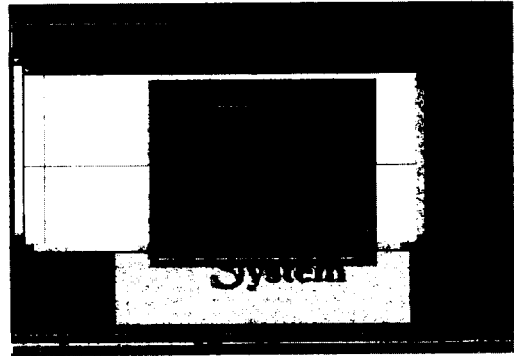
(그림 11) NOTE 비교를 위한 경우의 수

### 6.2 연주 결과 관리

연주 평가 결과는 서버의 데이터베이스에 저장되어 관리된다. 연주 평가 결과를 하나의 테이블, 즉 RESULT 테이블에서 관리한다. RESULT 테이블은 ID, 날짜, 내용 속성들로 구성되는데, ID는 가변 문자열로 선언되었으며, 25자까지 입력 가능하다. 사용자가 원격 합주 교육 시스템에 접속 시에 사용한 사용자ID를 저장한다. 그리고 날짜 속성에는 연주한 날짜를 저장하며, 내용 속성에는 연주 후 시스템에서 평가한 내용을 저장한다.

(그림 12)는 연주 후 자신과 상대방의 연주에 대한 박자와 음정에 대한 평가와 합주에 대한 결과를 보여주는 창이다. 이는 서버에 있는 정확한 음을 박자와 음정면에서 연주한 음과 비교하여 음단위로 평가를 내린 값을 종합하여 백분율로 나타낸 것이다. 이를 서버에 있는 데이터베이스에 저장할 수 있게 하였으며, 지금까지의 서버에 저장된 연주자의 과거 연주 평가 데이터를 볼 수 있게 하여 자신의 학습의 발전 사항을 한눈에 파악할 수 있도록 하였다. 단, 서버에서 검색하는 키워드는 사용자가 시스템에 접속할 때 사용한 사용자ID를 사용한다. 그러므로, 연주자가 과거의 자신의 데이터를 보기 위해서는 과거의 접속한 사용자ID와 같

은 사용자ID를 사용하여 접속하여야 한다.



(그림 12) 연주 평가 결과창

## 7. 결 론

본 연구에서 개발한 원격 합주 교육 시스템은 하나의 서버와 초고속 통신망으로 연결된 여러 개의 클라이언트로 구성되는데 연주자가 합주 또는 독주 연습 시 기존의 서버 데이터베이스에 저장되어 있는 미디어 파일 정보와 실제 연주자가 연주한 음악 정보를 비교 분석하여 음정, 박자 등에 대하여 온라인으로 지적 사항을 알려주며 연주 후에는 종합 분석 결과를 알려주어 연습을 효율적으로 진행할 수 있도록 한다. 그리고 각각이 연주자들의 연주 결과를 서버 데이터베이스에 저장, 관리하여 추후에 연주자에게 제공하여 좀더 효과적인 학습을 할 수 있도록 하였다.

서버는 연주곡의 미디어 파일 및 악보 파일을 저장, 관리하고 있으며, 이를 사용자에게 온라인으로 전달하고 연주자가 곡 선택 시에 정확한 연주곡명 외에도 작곡가명, 작사가명, 연주곡의 분류 및 가사의 일부에 대한 정보로도 자신이 원하는 연주곡을 찾을 수 있도록 지원하기 위해서 메타 데이터베이스를 저장, 관리한다 ([15]). 또한, 서버는 연주자의 연주 결과를 초고속 정보 통신망을 통하여 전달 받아 이를 서버에 저장된 미디어 파일과 비교하여 음정분석, 박자분석을 실시하여 이를 종합적으로 분석하여 연주자에게 결과를 제공하여 연주자가 자신의 연주에 있어서 수정하여야 할 부분들을 알게 한다. 클라이언트는 연주자의 연주를 입력 받아 이진 신호로 변환 및 분석하여 서버에 전달하며 서버에서 보내어진 연주 분석 결과를 연주자에게 보여준다. 또한 서버의 데이터베이스에 접근할 수 있

는 인터페이스도 제공한다. 향후 연구 방향으로는 본 시스템의 평가 과정에 있어 좀더 인공 지능적인 요소가 첨가되어야 한다는 것이다.

### 참 고 문 헌

- [1] 김종화, "MIDI 총론", 도서출판 세운, 1994.
- [2] 김평암, 최재남, "미디어와 케이크 워크", 성안당, 1995.
- [3] 박귀태, 이상락, "C 언어로 쉽게 쓰는 TMS320C31", 대영사, 1994.
- [4] 양정우, "미디어와 사운드 레코딩의 세계", 크라운 출판사, 1997.
- [5] 이문호, 염재훈, "C 언어를 이용한 디지털 신호처리 및 필터설계", 대영사, 1995.
- [6] 이석호, "데이터베이스 시스템", 정익사, 1995.
- [7] 지정규, 오해석, "선율을 이용한 음악정보 검색 시스템의 설계 및 구현", 한국정보처리학회 논문지 제5권 제1호, 1998.
- [8] Abraham Silberschatz, Henry F. Korth & S. Sudarshan, "Database System Concepts," The McGraw-Hill Companies, Inc. 1997.
- [9] C. J. Date. "An Introduction to Database Systems," Addison-Wesley Publishing Company, November, 1994.
- [10] Lawrence Rabiner, Biing-Hwang Juang, "Fundamental of Speech Recognition," Prentice-Hall, 1993.
- [11] L. R. Rabiner, R. W. Schafer, "Digital Processing of Speech Signals," Prentice-Hall, 1978.
- [12] TEXAS INSTRUMENTS, "Digital Signal Processing Applications with the TMS320C30 Evaluation Module," 1991.
- [13] TEXAS INSTRUMENTS, "TMS320C3x C Source Debugger User's Guide," 1993.
- [14] TEXAS INSTRUMENTS, "TMS320C3x User's Guide," 1994.
- [15] W. Bruce Croft, "Information Retrieval System : Theory and Implementation," Kluwer Academic Publishers, 1997.

### 한 창 호

e-mail : chhan@inha.ac.kr  
 1980년 성균관대학교 전자공학과 (공학사)  
 1982년 서울대학교 대학원 전자공학과(공학석사)  
 1991년 The Univ. of Texas at Austin(공학박사)

1983년~1986년 한국전자통신연구소 연구원  
 1991년~1992년 Cadence Design Systems, Inc., Senior Member of Technical Staff  
 1992년~현재 인하대학교 전자계산공학과 부교수  
 관심분야 : 컴퓨터 음악, 설계 자동화, 컴퓨터 그래픽스

### 이 경 명

e-mail : lee.km@zf-group.co.kr  
 1996년 인하대학교 전산공학과(공학사)  
 1999년 인하대학교 대학원 전산공학과(공학석사)  
 1999년~현재 ZF KOREA Co., Ltd. 근무

관심분야 : 컴퓨터 그래픽스, 설계 자동화

### 윤 광 섭

e-mail : ksyoon@inha.ac.kr  
 1981년 인하대학교 전자공학과(공학사)  
 1983년 미국 조지아 공대 전자공학과(공학석사)  
 1990년 미국 조지아 공대 전자공학과(공학박사)

1988년~1992년 미국 Silicon System Inc., 선임 연구원  
 1992년~현재 인하대학교 전자공학과 부교수  
 관심분야 : 집적회로 설계, 반도체소자 모델링

**류 기 흥**

e-mail : g9711117@inhavision.inha.ac.kr

1997년 인하대학교 전자공학과(공학사)

1999년 인하대학교 대학원 전자공학과(공학석사)

1999년~현재 삼성전자 근무

관심분야 : 신호처리회로설계, DSP 코딩

**모 종 식**

e-mail : g1981365@inhavision.inha.ac.kr

1998년 인하대학교 전산공학과(공학사)

1998년~현재 인하대학교 대학원 전산공학과 석사과정

관심분야 : 멀티미디어데이터베이스, 화상 및 음성 처리

**김 유 성**

e-mail : yskim@dragon.inha.ac.kr

1986년 인하대학교 전자계산학과(이학사)

1988년 한국과학기술원 전산학과(공학석사)

1992년 한국과학기술원 전산학과(공학박사)

1990년~1992년 삼성전자 컴퓨터부문 주임 연구원

1996년~1997년 미국, 피듀대학교 전산학과 방문연구원

1992년~현재 인하대학교 전산공학과 조교수

1998년~현재 인하대학교 전산정보실 시스템 부장

관심분야 : 멀티미디어 데이터베이스, 정보검색, 이동 데이터베이스, 트랜잭션 관리