논문 2010-47CI-5-6

MP3 장치용 플래시 메모리의 오류 검출을 위한 음원 비교 기법

(An Audio Comparison Technique for Verifying Flash Memories Mounted on MP3 Devices)

김 광 중*, 박 창 현**

(Kwang-Jung Kim and Chang-Hyeon Park)

요 약

휴대용 정보기기와 엔터테인먼트기기 등의 사용이 대중화 되면서 플래쉬 메모리의 수요도 급격히 증가하였다. 일반적으로 플래시 메모리는 장착되는 장치에 따라 다양한 형태의 오류 패턴을 가지며, 메모리 생산자들은 최종적인 생산과정에서 실제 장착되는 기기와 동일한 환경에서 전기적/물리적 테스트를 수행한다. 이 과정을 메모리의 응용기기 실장 테스트라고 하며, 여기에서 사용되는 장비를 메모리 실장기라 한다. 현재 여러 가지 종류의 실장기들이 제작되어 메모리 생산 환경에서 사용되고 있으나 대부분이 검수자의 청각이나 시각 등의 감각에 의존하여 메모리의 오류를 판단하고 있다. MP3 실장기의 경우 음원의 재생 기능을 이용하여 메모리 오류를 판단하는데 적절한 자동 검수 기법이 존재하지 않아 검수자가 실장기에서 재생되는 음원을 직접 듣고 오류를 판단한다. 이런 과정은 실장환경의 자동화에 있어 큰 걸림돌이 되고 있으며 인력 활용 측면에서도 비효율적이다. 본 논문에서는 MP3 장치용 플래시 메모리의 효과적인 오류 검증을 위한 음원 비교 기법을 제안한다. 제안하는 방법은 원본 파일과 MP3 장치에서 재생되는 샘플값의 분산을 활용함으로써 메모리 오류 발생 여부를 판단한다.

Abstract

Being popularized the use of portable entertainment/information devices, the demand on flash memory has been also increased radically. In general, flash memory reveals various error patterns by the devices it is mounted, and thus the memory makers are trying to minimize error ratio in the final process through not only the electric test but also the data integrity test under the same condition as real application devices. This process is called an application-level memory test. Though currently various flash memory testing devices have been used in the production lines, most of the works related to memory test depend on the sensual abilities of human testers. In case of testing the flash memory for MP3 devices, the human testers are checking if the memory has some errors by hearing the audio played on the memory testing device. The memory testing process like this has become a bottleneck in the flash memory production line. In this paper, we propose an audio comparison technique to support the efficient flash memory test for MP3 devices. The technique proposed in this paper compares the variance change rate between the source binary file and the decoded analog signal and checks automatically if the memory errors are occurred or not.

Keywords: Application-level Memory Tester, Audio Comparison, MP3 Device, Flash memory Testing

I. 서 론

휴대용 정보기기와 엔터테인먼트 기기 등의 사용이 대중화 되면서 플래시 메모리의 수요가 급격하게 증가하였으며, 그에 따라 메모리 생산업체에서 대규모의 생산 및 출하가 이뤄지게 되었다. 메모리는 사용되는 장

* 학생회원, ** 정회원, 영남대학교 컴퓨터공학과
(Department of Computer Engineering, Yeungnam University)
※ 이 연구는 2009년도 영남대학교 학술연구조성비에 의한 것임.
접수일자: 2010년3월3일, 수정완료일: 2010년8월31일

치에 특성에 따라서 오류 발생 유형이 다양한데, 메모리 생산업체에서는 제품 출하 시 수행하는 전기적/물리적인 불량 파악 테스트(electric test) 뿐만 아니라, 메모리가 실제로 탑재되는 응용 기기와 동일한 환경 및 조건하에서 데이터 무결성을 테스트하여, 제품의 불량률을 최소화시키기 위해서 노력한다. 이런 과정을 응용기기 실장테스트 (Application-level Memory Test)라고 하며, 이 과정에 사용되는 장비 및 설비를 실장기 (Application-level Memory Tester)라고 한다. 최근 플래시 메모리의 용량이 커지고, 생산량이 급증하면서 메모리 테스트의 자동화가 생산라인에서 중요한 사항으로 고려되고 있다.

본 논문에서는 MP3 장치에 탑재되는 플래시 메모리를 자동으로 테스트하기 위한 음원 비교 기법에 대해서 기술한다. 현재까지 여러 종류의 실장기들이 제작되어 생산업체에서 사용되고 있는데, 대부분 플래시 메모리에 저장되어 있는 바이너리 데이터를 단순 비교하는 방법을 사용하지만, 원본 데이터와 비교 대상 데이터의 1:1 비교 방법은 본 논문에서 다루는 MP3 장치 실장기에는 적합하지 않다. MP3 장치는 메모리에 저장되어 있는 데이터는 이상이 없더라도, 파일이 재생되거나 장치 내에서 삭제 및 수정되는 동안에 메모리에서 오류가 발생할 수 있기 때문이다. 따라서 본 논문에서는 MP3 장치의 고유 기능인 음원 재생 기능을 이용하여 원본 파일과 재생되는 음원을 녹음한 파일의 샘플값을 비교하는 기법에 대해서 설명한다.

기존의 음원 비교에 대한 연구는 음성인식이나 멜로디에 국한되어 왔다. 음성인식의 경우 다양한 표본의 학습과 정규화를 통해서 구현되며[1~2], 멜로디에 관한 연구는 간단하게 표현될 수 있는 음의 특성[5]을 활용한다. 그러나 이러한 기법들은 멜로디를 추출하기 어렵고 다양한 표본을 준비 할 수 없으며, 음원의 재생시간이 긴 MP3 장치 실장 환경에서는 적절하지 못한 방법이다.

산업현장에서 본 논문과 유사한 내용을 다룬 기술적인 시도가 있었지만, 같은 음원 파일이라도 재생되는 장치에 따라 약간의 주파수 차이가 존재 할 수 있으며, 비교 대상의 녹음 과정에서 필연적으로 잡음이 섞이기 때문에 좋은 결과를 가져오지 못하였다. 본 논문에서 제시하는 MP3 실장기의 목표는 음원을 재생하기 전단계의 바이너리 데이터를 확인하는 것이 아니라, MP3 장치가 안정적으로 데이터를 읽어서 재생하는 것을 검증하는 것이다. MP3 장치에 장착되어 있는 NAND 플래시 메모리는 데이터를 블록단위로 저장하며, 특정 메모리 셀에서 오류가 발생할 경우 그 오류는 좁게는 해당 셀을 포함하고 있는 페이지에서 넓게는 해당 블록에 영향을 미치게 되며, 해당 오류는 음원의 여러 구간에 걸쳐서 원본과 상이한 샘플값을 갖게 된다. 본 논문에서 제안하는 기법은 이런 영향을 받는 블록에서 발생하는 묵음 또는 강한 노이즈를 판별하기 위하여 샘플값의 분산 변화율을 비교한다.

본 논문은 Ⅱ장에서 그 동안 수행 된 음원 비교에 대한 관련연구를 살펴보고, 단순 데이터 비교의 문제점을 언급한다. Ⅲ장에서는 본 논문에서 제안한 음원비교를 통한 메모리 오류 검출 기법에 대해 기술하고 Ⅳ장에서 실험결과를 보이며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. 실장테스트의 필요성

플래시 메모리는 그의 물리적 특징으로 인하여 다양한 형태의 오류를 나타낼 수 있으며, 기존에 사용되던 자기 기반의 저장장치와는 다른 오류 검출 방법이 요구된다. 플래시 메모리에는 데이터를 저장 할 수 있는 셀이 2차원 또는 3차원으로 배열되어 있으며, 최근 플래시 메모리의 집적도가 높아지면서 특정 셀이 주위 셀의 간섭에 의한 오류가 많이 발생하고 있다[3~4].

플래시 메모리에서 발생할 수 있는 오류의 대표적인 형태는 셀 간섭에 의한 오류, 컨트롤러 이상 동작에 기인한 오류, 전력 문제로 인한 오류, DRAM으로의 버퍼링 과정에서 발생할 수 있는 오류로 나눌 수 있는데, 앞의 3종류의 오류는 플래시에 저장되어 있는 데이터와 원본 데이터를 1:1로 비교함으로써 찾아 낼 수 있다. 하지만 MP3 장치 등과 같이 장치 내부에서 DRAM으로 적재하는 과정을 가진 장치에서는 데이터를 직접적으로 비교하는 방법을 사용할 수 없다.

이러한 이유로 실제 플래시 메모리가 장착되는 환경과 유사한 환경을 구축하고, 해당 장치를 직접 동작시킴으로써 플래시 메모리의 안정적인 데이터 버퍼링을 확인해야만 한다. 특히 MP3 장치 같은 경우, 플래시 메모리의 용량이 커지면서 이를 검증하기 위하여 실장기에서 재생되는 음원을 검수자가 시작부터 마지막까지 들어본 후 오류 여부를 판단하기 때문에, 검수 과정의 완전한 자동화가 이뤄지지 않는 실정이다. 따라서 이런 검수과정의 자동화를 위해서 원본 음원 재생되는 음원

을 비교하는 자동화된 견고한 기법이 절실히 요구된다.

2. 음원 비교

음악 비교의 대표적인 예는 MIDI(Music Instrument Digital Interface)[7] 파일과 마이크로폰을 통하여 입력받은 사람의 허밍에서 음높이의 변화(Pitch Contour)를 감지하여 UDR(Up, Down, Repaet) 스트링으로 표현하는 방법을 사용하는 기법이다[8]. 이러한 음악 비교 기법은 대부분 MIDI 형태의 파일에서 이뤄진다. 그 이유는 MIDI 파일의 경우 특정 악기 채널을 제거하면 중요 멜로디를 쉽게 추출할 수 있기 때문이다. 하지만 본 논문에서 대상으로 하는 MP3 파일이나 웨이브 파일에서는 멜로디를 추출하는 방법은 구현하기가 매우 힘들고, 몇몇 기 구현된 사례[9~11]에서도 그 결과물에 대한 신뢰성이나 활용도가 떨어진다. 따라서 실제 음의 샘플들로 구성된 파일들의 정확한 비교를 위해서 멜로디 추출 및 비교를 통한 기법을 이용하는 것은 정확도를 요구하는 실장 환경에 적합하지 않다.

이 외의 기법으로 녹음되는 음원에서 필연적으로 추가되는 잡음을 제거하여 비교를 수행하는 기법[12~13]에 대한 연구도 진행되었다. 대부분의 잡음 제거 기법이 음성 녹음 환경에 대해서 국한되어 있으며, 대표적인 방법으로 잡음의 함수를 구하여 음원에서 잡음을 제거하는 기법이 활용된다. 하지만 원본 파일에도 잡음이 섞일 수도 있으며, 완벽한 잡음 함수를 구하는 것은 불가능하다.

따라서 실장 환경의 자동화를 위한 음원 비교 기법은

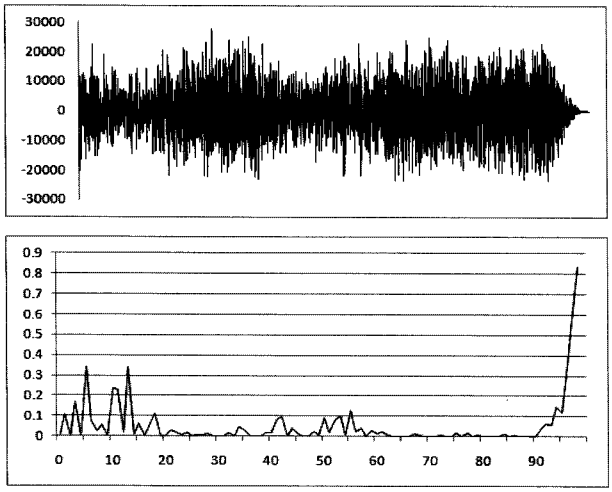


그림 1. 볼륨 수준 70%의 음원 파형과 분산변화율
Fig. 1. Frequency form and a variance change rate at 70% volume level.

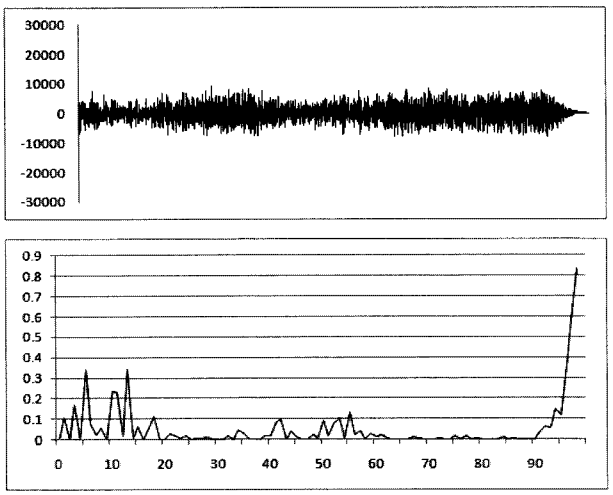


그림 2. 볼륨 수준 30%의 음원 파형과 분산변화율
Fig. 2. Frequency form and a variance change rate at 30% volume level.

멜로디를 추출하지 못하는 상황에서도 적용 될 수 있어야 하며, 잡음에 견고하여야 한다.

III. 음 원 비 교

본 장에서는 본 논문에서 제안하는 기법에 대해서 기술한다. 본 논문에서 제안하는 플래시 메모리 테스트를 위한 음원 비교 기법의 기본적인 아이디어는 음원 파일을 여러 구간으로 나누었을 때 구간별 샘플값 분산의 변화율은 어느 정도의 잡음이 첨가되더라도 큰 변화가 없다는 것이다. 잡음은 디지털 신호 처리 과정에서 포함되는 잡음과 MP3 장치 또는 워크스테이션의 고유의 잡음이 될 수 있다. 그러나 이러한 잡음은 음원이 녹음되는 시간동안 일정하며, 샘플값 분산의 변화에는 큰 영향을 미치지 않는다.

1. 샘플값의 분산이 가지는 의미

분산이란 도메인 내의 샘플들이 얼마나 균일한가를 나타내는 수학적 수치이며, 도메인 내의 몇몇 샘플들이 특별한 값(예:impulse)을 가지고 있다 하더라도 결과에는 큰 영향을 주지 못한다. 그리고 분산은 샘플들의 평균 크기에 민감하지 않다. 이런 분산의 두 가지 큰 특징은 잡음이 균일하게 포함될 가능성이 있고, 정확한 볼륨 수준을 설정하기 어려운 실장환경에서 유용하게 활용될 수 있다. 본 논문에서 제안하는 기법은 음원을 일정한 구간으로 분할하여 각 구간에 포함된 샘플값들의 분산을 계산한 뒤, 구간별 분산의 변화율로 음원에 특

표 1. 변수의 의미
Table 1. Description on variables.

변수	의미
V_k	k번째 구간의 분산
C_k	k 구간과 k-1 구간 사이의 분산 변화율
N_k	k번째 구간의 총 샘플 갯수
S_i	구간의 i번째 샘플
A_k	k 번째 구간의 샘플값의 평균

징을 부여한다. 식 1은 k번째 구간의 분산을 나타내고 있으며, 식 2는 k번째의 분산 변화율을 보이고 있다. 표 1은 식 1, 식 2에 사용된 변수의 의미를 나타낸다.

$$V_k = \frac{\sum_{i=1}^{N_k}(S_i - A_k)^2}{N_k} \quad (1)$$

$$C_k = 1 - \left(\frac{V_k}{V_{k-1}}\right)^2 \quad (2)$$

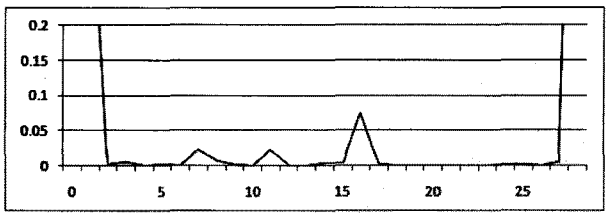
그림 1과 그림 2는 볼륨 수준이 같지 않은 두 음원 파일의 샘플값의 분산 변화율을 보이고 있다. 그림 1과 그림 2에 보이듯이 분산 변화율은 샘플값들의 평균에 영향을 받지 않는다.

2. 구간 분할

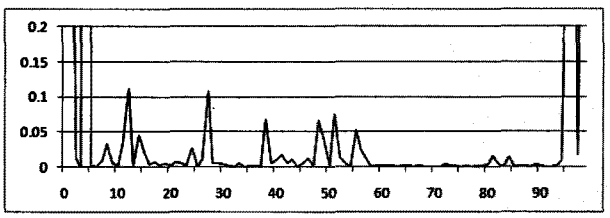
샘플값들의 분산을 구하기 위해서 전체 음악을 일정한 간격의 구간으로 분할한다. 분할된 구간의 수는 음원 비교에 소요되는 시간과 정확도에 중요한 영향을 미친다. 일반적으로 많은 구간으로 분할할 경우 샘플 간 비교에 대한 정확도는 높아지지만, 그에 소요되는 시간이 증가한다. 정확도가 증가한다고 해서 너무 많은 구간으로 분할하게 되면 결론적으로 각각의 샘플을 비교하는 것과 같기 때문에 분산을 이용한다는 것이 무의미해 질 수 있다.

그림 3은 원본 파일과 MP3 장치에서 재생된 음원을 녹음한 파일을 30, 100, 1,000 구간으로 분할하였을 때의 구간별 분산 변화율을 나타내고 있다.

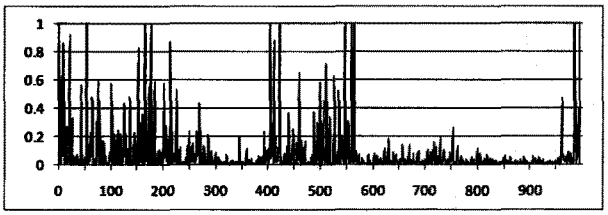
너무 적은 구간으로 분할할 경우 음원을 특정지을 수 없고, 너무 많은 구간으로 분할 할 경우 결과 비교에 소비되는 시간이 증가하며, 비교에 대한 정확도도 떨어진다는 것을 확인 할 수 있다. 본 논문에서는 여러 가지 경우의 구간 분할에 대한 실험을 수행한 후, 시간과 정확성의 효율성을 가장 잘 나타내는 100구간으로 분할하여 제안 기법에 적용한다.



(a) 30구간 분할(Blocks devided by 30)



(b) 100구간 분할(Blocks devided by 100)



(c) 1,000구간 분할(Blocks devided by 1,000)

그림 3. 동일 음원을 30, 100, 1,000 구간 분할 했을 때의 분산 변화율
Fig. 3. Variance change rates of a same audio when it is devided by 30, 100 and 1,000 blocks.

3. 음원비교 시작점 동기화

원본 음원 파일과 실장기에서 재생되는 음원의 비교를 위해서는 원본 파일과 비교 대상 파일의 시작점을 일치시켜야 한다. 본 논문에서 제안하는 방법은 시작 시점을 임의로 선정 후 단위 시간 별로 시작 시점을 변경시키면서 분산 변화율을 측정하여 두 음원의 분산 변화율이 가장 일치하는 곳을 시작점으로 결정한다. 그림 4는 두 음원의 시작점을 일치시키는 기법의 알고리즘을 보이고 있다.

이 과정을 통해 얻어지는 두 음원의 분산 변화율 비교 결과의 모습은 그림 5와 같다.

그림 5는 0.1초 단위로 구간별 분산 변화율 비교를 반복하여 측정한 일치 비율을 나타내고 있다. 그림 5에서 3.7초 지점에서 일치정도가 가장 높음을 알 수 있고, 이 결과는 MP3 실장기에서 재생되는 음원을 녹음한 파일의 비교 시작점을 3.7초로 설정했을 때, 일치율이 가장 높음을 의미한다. 또한 이 결과는 녹음이 시작되고 약 3.7초 뒤에 재생이 시작되었다는 것을 나타내고 있

[알고리즘 1]
원본 파일과 녹음된 파일의 시작점을 동기화하는 알고리즘

```
MinDifference = MAX_NUMBER

/* 시작점에서 특정 시간까지 일정 시간 간격으로 비교 */
FOR i = StartingSample TO Seconds STEP compareUnit
  CurrentDifference = 0.0
  Record.CRVariance[i]          =
          getCRVariance(Origin, i, SamplePerBlock);

/* 시간내의 분할된 구간의 분산 변화율의의 일치도를 계산
*/
  FOR j = 0 TO SamplePerBlock
    CurrentDifference = CurrentDifference +
    POW(Origin.CRVariance[j]  - Record.CRVariance[[j],
2);
  END FOR

/* 일치도가 가장 좋을 경우 해당 i를 시작점으로 설정 */
  IF CurrentDifference < MinDifference THEN
    MinDifference = CurrentDifference
    MinPoint = i
  END IF
END FOR
```

변수
Origin : 원본 파일
Record : 실장기에서 재생되는 신호를 녹음한 파일
StartingSample : 음원 파일의 시작점
SamplesPerSec : 비교 단위
SamplesPerBlock : 시작점을 이동시키는 시간 단위
CurrentDifference : 두 음원의 분산 변화율 차이
MinDifference : 분산 변화율의 차이의 최소값
MinPoint : 최소의 분산변화율 차이를 보이는 시작점
A.CRVariance[x] : A음원의 x번째 블록의 분산 변화율

함수
getCRVariance(A, x, n) : n을 시작점으로 가지는 A음원의 n 번째 분산 변화율을 얻어오는 함수

그림 4. 시작점 동기화 알고리즘
Fig. 4. Algorithm of synchronization starting point.

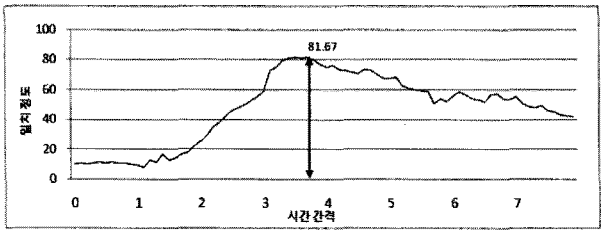


그림 5. 시작점에 따른 음원 일치도
Fig. 5. Audio sync rate through starting point.

으며, 이와 같은 결과가 나타난다면, 원본은 0초부터, 비교 대상 파일은 3.7초부터 구간을 분할하면 된다.

4. 메모리 오류 판별

구간별 음압 분산 변화율은 두 개의 음원 파일의 동일한 구간에 대해 분산의 증가와 감소를 나타낸다. 경우에 따라 증가와 감소가 서로 반대로 나타나는 현상이 발생 할 수 있지만, 그 차이는 크지 않다. 그리고 메모리 오류 발생시 MP3 장치에서 묵음이나, 강한 잡음으로 음원이 출력될 경우 두 파일의 구간별 분산 변화율은 큰 차이를 나타낸다. 오류가 발생한 블록에서는 묵음일 경우 분산은 0이 되며, 강한 잡음일 경우 그 값이 매우 커진다. 따라서 이전 블록의 분산과 비교한 결과값인 변화율은 크게 증가하며, 메모리 오류 구간동안은 이전 블록과 분산차이가 크게 없음으로 분산 변화율이 0에 가까워진다. MP3 장치에서 메모리 오류 구간을 건너뛰기, 해당 음원의 종료 등 다른 방법으로 처리 할 경우에도 원본이 가지고 있던 신호가 정상적으로 출력되지 않음은 분명하기 때문에 원본 파일과 비교 했을 때 상이한 분산 변화율을 가지게 된다. 메모리 오류는 다음의 두 가지 조건으로 판단한다.

조건 1 : 원본파일의 분산 변화율이 일정 수치 이상이면서 비교 대상 파일의 분산 변화율이 0에 가깝게 몇 구간 유지되고 두 음원에 대해서 동일한 현상이 발생하지 않으면 메모리 오류가 발생했다고 판단한다.

조건 2 : 두 파일의 k 번째 구간의 분산 변화율을 O_k, R_k로 정하고, 만약 O_k, 또는 R_k가 1을 초과 하면서 한 값이 다른 값보다 2배 이상으로 커지고 두 음원에서 동일한 현상이 발생하지 않으면 메모리 오류가 발생했다고 판단한다.

※ 음원의 종류에 따라 조건 1, 조건 2에 해당하는 경우가 발생할 수 있는데, 두 음원에서 동일한 현상이 발생하지 않는 경우 오류가 발생한 것으로 판단한다.

여기서 변화율 초과값 1과 차이값 2는 다양한 음원 파일에 대해서 수행된 실험을 바탕으로 얻어진 경험적 수치이다. 위의 조건에 해당하지 않는 경우는 다음 식 3을 통해서 메모리 오류를 판단한다.

여기서 O_k은 원본 파일의 k번째 구간의 분산 변화율, R_k은 MP3 장치에서 출력된 음원을 녹음한 파일의 k번째 구간의 분산 변화율을 나타낸다. 식 3은 음원의 일치율을 나타는 식으로써, 만약 식 3의 결과가 특정 값보다 작을 경우 해당 음원에서 오류가 발생한 것으로 판단한다.

$$\sum_{n=0}^{\#of\ Blocks} \frac{MAX(O_k, R_k) - |O_k - R_k|}{MAX(O_k, R_k)} \quad (3)$$

Ⅳ. 구 현 및 실 험

본 장에서는 본 논문에서 제안한 기법에 대한 실험과 실장 시스템의 구조 및 그것을 바탕으로 제작된 실장기에 대해서 기술한다. 그림 6은 실장 시스템의 구조를 보이고 있다.

본 논문에서 제안한 기법을 바탕으로 구현한 실장기는 총 4대의 MP3 장치를 동시에 구동하며, 각 MP3 장치는 장착되어 있는 플래시 메모리를 탈/부착 할 수 있도록 메모리 장착부분을 소켓으로 대체하였다. 하나의 워크스테이션에서 녹음할 수 있는 스트림은 하나로 한정되기 때문에, 각각의 MP3에서 음원이 출력되는 부분은 오디오 버퍼를 별도로 구성하였다.

그리고 MP3 장치의 제어를 위하여 워크스테이션의 병렬포트를 활용하였다.

그림 7은 위의 구성에 따라 실제 제작된 실장기를 보이고 있다. 실장기 내부에 워크스테이션이 내장되어 있

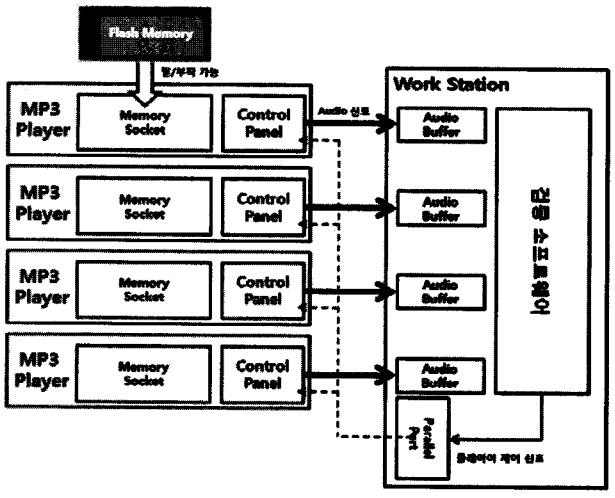


그림 6. MP3 장치 실장 시스템의 구성도
Fig. 6. The brock diagram of MP3 device application-level tester.

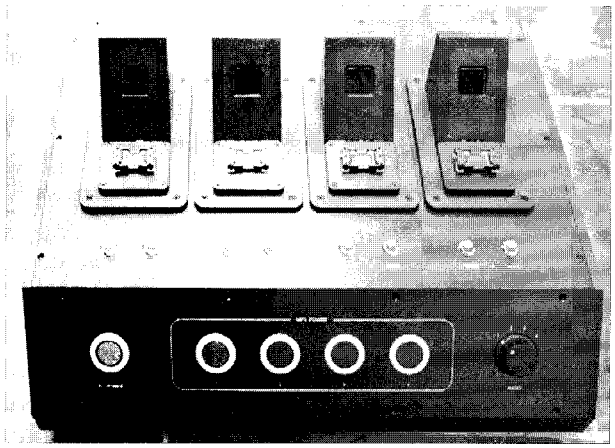


그림 7. 제작된 MP3 장치 실장기
Fig. 7. The implemented MP3 device application-level tester.

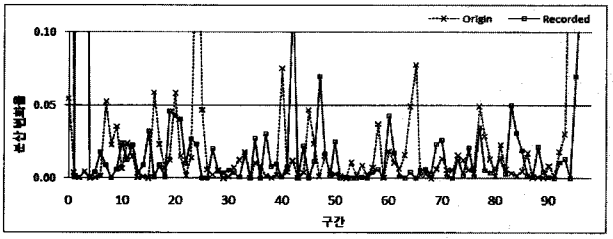


그림 8. 시작점을 일치시키기 전의 두 음원의 분산 변화율
Fig. 8. Variance change rates of two audios before starting point synchronization.

으며, 실장기 후면에 모니터 및 키보드 등 인터페이스를 위한 포트를 포함하고 있다.

본 논문의 실험은 식 3에 의해 계산된 음원 일치도가 70% 이하일 경우 메모리의 오류가 발생했다고 가정하고 수행되었다.

첫 번째 실험은 Ⅲ장 3절에서 기술한 시작점을 일치시키는 기법에 대한 결과를 확인하기 위하여 수행되었다. 실험에 사용된 음원은 3분 10초의 재생 시간을 가지고 있으며, 음원을 100구간으로 분할하였다. 그림 8은 시작점을 일치시키기 전의 두 음원의 구간별 분산 변화율을 보이고 있다. 식 3을 통해 계산한 두 음원의 일치도는 0.0%이다. 이는 두 음원이 다른 음원임을 의미한다.

그림 9는 Ⅲ장 3절에서 제안한 기법을 적용한 후 시작점을 일치시킨 뒤의 분산 변화율을 보이고 있다. 실험 후 두음원의 일치도는 79.03%이다. 이 실험 결과에서는 앞에서 지정한 음원 일치도인 70% 보다 높기 때문에 메모리에 오류가 발생하지 않은 것으로 간주 할 수 있다.

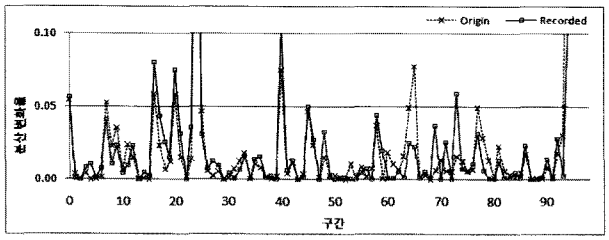


그림 9. 시작점을 일치시킨 후의 분산 변화율
Fig. 9. Variance change rates of two audios after starting point synchronization.

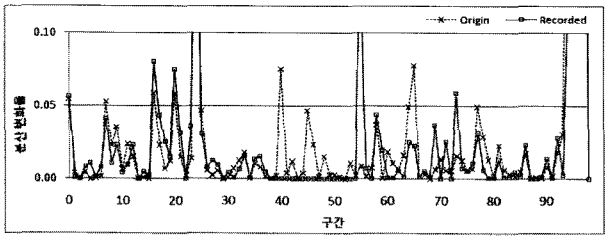


그림 10. 오류를 묵음으로 처리했을 때의 분산 변화율
Fig. 10. Variance change rate on the situation that MP3 device treats error to mute.

MP3 장치는 종류에 따라 메모리에서 발생하는 오류를 처리하는 방법이 다양하다. 메모리를 처리하는 방법은 강한 잡음, 묵음, 오류구간 무시, 다음곡 재생 등 4가지로 구분할 수 있다. 두 번째 실험은 이런 다양한 처리에 따른 상황을 재현하여 수행되었다. 메모리 오류의 발생 재현은 하나의 블록이 오류에 영향을 받는다고 가정했을 때, 일반적으로 MP3에 사용되는 플래시 메모리의 블록크기가 512Kbyte일 경우 실험에 사용된 MP3 음원에서 영향 받는 구간은 약 1/10 정도 된다. 따라서 그에 해당 하는 구간을 묵음, 강한 잡음으로 처리하였다. 그림 10은 오류 구간을 묵음으로 처리 했을 때의 결과를 보이고 있다. 37번째 구간부터 녹음된 음원의 분산 변화율이 0으로 55번째 구간까지 유지된다. 이 후 그래프상에서 56번째 구간에서 분산 변화율이 급격하게 증가하는데 0으로 유지되던 변화율이 56번째에서 변화하는 원인으로 작용하였다. 이는 3.4절에서 기술한 "조건 1" 해당되기 때문에 메모리 오류가 발생한 것으로 판단할 수 있다. 식 3에 의한 일치도는 17.75%이다.

다음 실험은 오류 구간을 강한 잡음으로 처리 했을 때를 재현하여 수행되었다. 그림 11에서 그의 결과를 보이고 있다. 58번째 구간에서 녹음된 파일의 분산 변화율이 급격히 증가하고 변화율이 0으로 약 12구간 유지된 후 다시 변화율이 급격히 증가한다. 이 현상은 58번째 분산값이 59번째 분산값 0으로 변화하면서 (식 2)에 의해 변화율이 급격하게 증가하였고, 0으로 유지되는 구간은 잡음 구간의 분산값이 일정하기 때문에 발생한다. 이는 "조건 1"에 해당하므로 메모리 오류가 발생한 것으로 판단된다. 일치도는 31.47%이다.

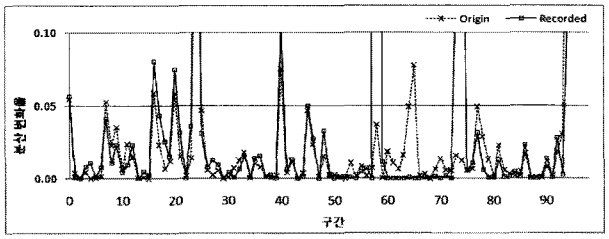


그림 11. 오류를 강한 잡음으로 처리했을 때의 분산 변화율
Fig. 11. Variance change rate on the situation that MP3 device treats error to noise.

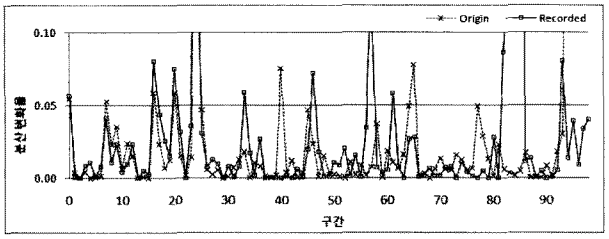


그림 12. 오류 구간을 무시했을 때의 분산 변화율
Fig. 12. Variance change rate on the situation that MP3 device ignores the errors.

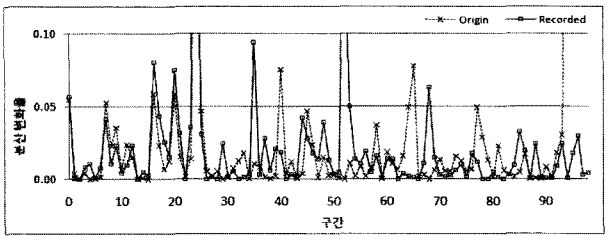


그림 13. 오류 발생시 다음 음원을 재생했을 때의 분산 변화율
Fig. 13. Variance change rate on the situation that MP3 device skips error and plays next audio file.

다음 실험은 MP3 장치가 오류구간을 무시했을 때를 재연하여 수행되었다. 그림 12를 확인했을 때 40번째 구간에서 두 음원의 분산 변화율이 큰 차이를 보이고, 그 이후로 두 음원이 일치하지 않음을 알 수 있다. 일치도는 5.56%이다.

이번 실험은 메모리에 오류가 발생했을 때 재생하던 음원을 중지시키고 재생목록에 있는 다음 음원의 재생이 시작되었을 때를 재현하였다. 그림 13에서 그 상황에 대한 실험결과를 보인다. 29번째 구간에서 눈에 띄는 차이가 발생하다 그 이후로 두 음원이 일치하지 않는 것을 확인 할 수 있다. 이 실험의 일치도는 17.86%이다.

앞서 기술된 4개의 실험은 음원이 정상적으로 재생되다가 오류가 발생했을 때를 재현하였다. 이 외에도 메모리 오류가 발생했을 때, 해당 파일에 대해서 재생을 못하는 경우를 고려 해 볼 수 있지만, 이런 경우는 시작점을 일치시키는 과정에서 오류를 찾아 낼 수 있기 때문에 실험에서 제외하였다.

4가지 실험에서 모두 실험 전에 지정한 임계치 70%에 도달하지 못했기 때문에 정상적으로 메모리에서 오류가 발생한 상황을 판별 할 수 있었다.

V. 결 론

본 논문에서는 MP3 장치 실장 검증 과정에서 자동화를 위한 음원 비교 기법에 대해서 기술하였다. 본 논문에서 제안하는 기법은 음원을 여러 구간으로 분할하고 구간별 샘플값의 분산을 구한 후 구간별 분산 변화율로 음원을 특징지었다. 실험 결과 메모리 오류에 대해 MP3 장치가 처리하는 다양한 방법에 대해서 메모리 오류를 검출 할 수 있었다.

차후 연구로는 샘플값의 분산을 이용한다는 측면에서 MP3 외에도 휴대전화나 멀티미디어 장치 실장기에서 영상의 샘플값을 활용한 비교 기법을 고려하고 있다.

참 고 문 헌

[1] E. Eide and H. Gish, "A Parametric Approach to Vocal-Tract-Length Normalization" *Proc. of Acoustics, Speech, and Signal Processing,* vol. 1 pp. 346-348, May 1996

[2 L. Lee and R. Rose, "A Frequency Warping Approach to Speaker Normalization" *IEEE Transactions on Speech and Audio Processing,* vol. 6, pp. 49-60, Jan 1998.

[3] "Wikipedia Flash Memory - Nand" from http://en.wikipedia.org/wiki/Flash_memory#NAND_flash

[4] J. D. Lee, S. H. Hur, and J. D. Choi, "Effects of the floating-gate interferences on NAND Flash memory cell operation" *IEEE Electron Device Letters,* vol. 23, pp. 264-266, May 2002

[5] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems." In DAFX-05:*Proceedings of the 8th Int. Conference on Digital Audio Effects*, pp. 153 - 160, 2005.

[6] Mongeau Marcel and David Sankoff, "Comparison of musical sequences", *Computers and The Humanities* vol. 24, No.3, pp. 161-175, 1990

[7] "Wikipedia MIDI" from http://en.wikipedia.org/wiki/MIDI

[8] A. Ghias, et al., "Query by humming - musical information retrieval in an audio database" *In ACM Multimedia 95 - Electronic Proceedings*, pp. 231-236, USA, 1995.

[9] "Wav to midi conversion software - Solo Explorer" from http://www.recognisoft.com/

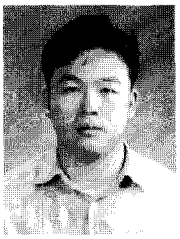
[10] "AmazingMIDI. WAV to MIDI converter for music transcription" from http://www.pluto.dti.ne.jp/~araki/ amazingmidi/

[11] "IntelliScore Polyphonic. WAV to MIDI Converter" from http://www.intelliscore.net/

[12] 한진우, 김범상, 서창우, 이기용, "음질개선을 위한 적응평균방법을 가진 스펙트럼 차감법" *정보통신연구 진흥원 학술기사*, 2000년도 추계종합학술발표회

[13] 김진영, 엄기완, 최홍섭, "Wiener Filtering을 이용한 잡음환경에서의 음성인식" *한국음성학회, 음성과학* 제 1권, 1997. 4, pp. 277 - 283

저 자 소 개



김 광 중(학생회원)

2006년 영남대학교 컴퓨터공학과 공학사

2008년 영남대학교 컴퓨터공학과 공학석사

2008년~영남대학교 컴퓨터공학과 박사과정

<주관심분야 : 임베디드 시스템, 멀티미디어 신호처리, 지능형 자동차>

박 창 현(정회원)-교신저자

1986년 경북대학교 전자공학과 전산학전공 공학사

1988년 서울대학교 전산과학과 인공지능전공 이학석사

1992년 서울대학교 전산과학과 인공지능전공 이학박사

1992년~1993년 서울대학교 컴퓨터신기술공동연구소 특별연구원

1993년~현재 영남대학교 컴퓨터공학과 교수

1998년~1999년 Univ. of Maryland, UMIACS 연구소 방문연구원

2009년~2010년 Univ. of Washington, Dept. of Electrical Eng., 방문연구원

<주관심분야 : 인공지능, 임베디드 시스템, 지능시스템, 지능형 자동화>