

효과적인 초음파검사를 위한 동화상 검색시스템 설계 및 구현

Design and Implementation of Video Retrievaling System for Effective Ultrasonograph

오 태 석*, 오 무 송**
(Teh Sok Oh*, Moo Song Oh**)

요 약

초음파 진단장치는 X선 촬영과 달리 인체에 해를 주지 않고 계속적으로 장시간 관찰할 수 있고 실시간으로 영상을 볼 수 있으며, 또 타장비에 비해 가격이 저렴하고 소형이라는 장점이 있다. 현재는 이 영상들을 대용량 저장매체에 저장되어 컴퓨터를 통해 재생하여 볼 수 있게 되었다. 본 논문에서는 이러한 막대한 양의 영상데이터를 검색하기 위한 새로운 검색 방법을 제시한다. 제안하는 검색방법은 검색을 원하는 정지화상의 컬러이미지를 제시하면 시스템이 이를 자동으로 분석하여 이미지 데이터베이스에 저장된 유사한 이미지데이터들과 관련된 정보들을 정의결과로 나타내어 쉽게 검색하고자 한다. 이를 위하여 사용자가 제시한 정지화상을 Bitmap으로 구성하고, Bitmap전체의 비디오 메모리에서 검색할 부분 영역을 검색대상으로 설정한다. 이 값을 key값으로 우선적으로 여기에 원하는 유사미를 설정한 후 전체 동화상의 각 프레임에서 추출한 비디오 메모리 데이터와 검색 화면의 비디오 메모리를 Pixel별로 비교하여, 사용자가 원하는 영상데이터의 위치 point 값과 유사비율값을 보관한다. point값으로 보관된 것을 유사비율에 따라 우선 순위를 정하여 데이터베이스에 보관하고 이 보관된 후보 이미지들을 순위별로 화면에 나타내어 사용자가 원하는 이미지데이터를 쉽고 빠르게 검색할 수 있었다.

ABSTRACT

Ultrasonography is more useful than X-ray because it dose not harm to humanbody and look at the image unceasingly. And it also smaller and cheaper than other equipments. Nowadays, this images can be stored in very large storage memory unit and seen through computer replay. For this, the static image suggested by users is structured as Bitmap and the retrieval object is selected from this entire Bitmap. And the comparing rate is set as key-value and then memory data extracted from each frame of entire video is compared to video memory of retrieved image in each pixel. From this, the position value of image data users desire as point value and the comparing rate are stored. Storage as pont value is ordered according to comparing rate and stored in database. And then the candidate images represented according to the above rate. This proposed system can retrieve the desired image easily and rapidly.

I. 서 론

인류는 이제까지 질병과 싸우며 이의 퇴치를 위해 많은 노력을 해오고 있으며, 장기의 이상유무를 판별하고 인체 내부구조를 피부 절개 없이 내부를 들여다 보기위해 많은 노력을 기울여왔다. 인체영상을 가시적으로 영상화하려는 노력은 X선에 의해 임상의학 영상진단법이 시작된 이래 자기공명 영상진단장치, 초음파 진단장치, 감마선의 양전자를 이용한 핵의학 영상진단장치등 침단 의료 영상시스템으로 발전되어왔다. 이들 시스템중 비교적 간단한 X선 촬영은 비용이 싸다는 장점이 있지만 촬영과

현상에 걸리는 시간과 그 필름을 찾기 위해 분류하고 방대한 양의 필름을 보관하기 위해 넓은 장소가 필요하고 분실의 문제점이 나타나고 있다.[1]

초음파 진단장치는 X선과 달리 인체에 해를 주지 않고 계속적으로 장시간 관찰하여 실 시간적으로 영상을 볼 수 있고, 또 타장비에 비해 저렴하고 소형이라는 장점이 있다. 이러한 점에 비하여 영상 해상도(Resolution)를 올리려면 높은 주파수를 사용해야 하는데 고주파일수록 인체내에서 감쇠가 심하므로 깊은 부위에서 해상도가 높고 좋은 영상을 얻기 힘든 단점이 있다.[2]

이에 본 논문은 초음파 진단장치에 컴퓨터를 연결하여 그 영상을 AVI(Audio Video Interface)에 저장해서 진료가 원하는 영상을 선택, 저장하면 지나가 버린 영상을 다시 검색하여 원하는 영상을 찾아 그 화면부터 다시 재생해 볼 수 있도록 하고, 초음파에서 얻은 감쇠가 심한

* 목포과학대학 전자과

** 조선대학교 컴퓨터공학과

접수일자 : 1998년 7월 22일

해상도가 낮은 영상을 전 처리하여 관찰자의 필요에 맞게 영상처리해서 보다 더 신명하고 확실한 영상을 얻기 위해 Adjust (Level, Brightness, Contrast)와 Filter (Noise, Sharpen)와 Map (Inverter, Equalize)등 여러 가지 기법을 사용하였다.

본 논문에서 제안하는 검색방법은 사용자가 검색을 원하는 정지화상의 이미지데이터를 Bitmap으로 구성하고, Bitmap 전체의 비디오 메모리를 전부 비교하기에는 시간이 많이 소요되므로 전체 화면을 3차 형식으로 검색할 부분 영역을 검색대상으로 설정하고, 이 값을 key값으로 우선적으로 유사비를 설정한 후 전체동화상의 각 프레임에서 추출한 비디오 메모리와 검색 화면의 비디오 메모리를 Pixel별로 비교하여, 사용자가 원하는 영상데이터의 위치를 프레임 point 값과 유사비율을 보관한다. 프레임 point값으로 보관된 것을 유사비율에 따라 sort하고, 우선순위를 정하여 데이터베이스에 보관하고 이 보관된 이미지서후보가 많이 선택된 경우는 유사비가 높은 프레임 point 값 화면에 나타내어 사용자가 원하는 이미지데이터를 선택하여 쉽고 빠르게 검색하도록 시스템을 설계 및 구현하였다.

본 논문의 구성은 2장에서 검색에 사용하는 이미지 파일의 구성에 대하여 간략하게 소개하고, 3장에서는 동영상 검색 방법으로 Visual C++의 검색엔진 프로그램의 구성과 처리내용을 설명, 4장에서는 검색 실험결과와 구현해서 얻은 화면을 바탕으로 설명하였고, 5장에서 결론을 맺는다.

II. 이미지 파일의 구조

영상데이터를 검색하기 위해서 검색을 원하는 정지화상의 이미지데이터의 컬러이미지 또는 시각적인 모델을 제시하여야 하는데 이 제시모델을 본 논문에서는 BMP파일을 사용하였다. BMP파일은 윈도우에서 어느 컴퓨터에 서든지 화면에 출력시킬 수 있는 장치 독립적인 비트맵 파일이기 때문에 사용하기가 편리하다. BMP파일의 구조는 처음의 두 바이트는 BM이란 BMP파일이라는 것을 알리는 문자가 있으며 다음에 DWORD형으로 BMP 실질 이미지 데이터가 있는 파일의 포인트를 가리킨다. 그 다음에는 BMP의 실질적인 헤더정보가 있는데 biSize는 현재 헤더의 크기를 말해주고, 다음으로 가로와 세로의 크기, 플랜의 값이 설정된다. BitCount는 한 픽셀의 정보가 들어있는 비트 수를 말한다.

BMP 파일 형태의 두 번째 블록에 있는 데이터들은 구조체라고 볼 수 있는데 이 구조체의 biXPelsPerMeter와 biYPelsPerMeter 두개의 요소들 가지고 이미지의 크기를 결정한다. 다음 biClrImportant는 현재 이미지에서 가장 많이 사용하거나 중요하게 사용하는 색의 값을 지정하고, 팔레트가 다른 두개의 이미지를 교차하였을 경우 그림이 깨어지는 것을 방지하기 위해 정보로 가지고 있어야한다. BITMAPINFO의 구조체는 다음과 같다.

```
typedef struct tagBITMAPINFO
{
    BITMAPINFOHEADER bmiHeader ;
    RGBQUAD bmiColors[1]; //팔레트 테이블
} BITMAPINFO ;
```

간단 BITMAPINFO는 현재 BMP의 모든 정보를 다 가지고있다. 여기에 이 정보를 가지고 동화상 검색을 하는데 이용하고자 한다.

CMyDib 클래스 생성시에는 생성자에 파일을 주어서 생성과 동시에 이미지를 메모리에 로드하게 다음과 같이 만들었다.[3]

```
CMyDib :: CMyDib (LPSTR fileStart_Position ; )
{
    Istrcpy ((LPSTR) m_strFileName, (LPSTR) filename) ;
    szDib = NULL ;
    lpDib = NULL ;
    m_palDIB = NULL ;
    szPal = NULL ;
    if (!bLoadBmpFile( ))
        AfxMessageBox(m_strMessageTemp) ;
}
```

III. 동영상 검색 방법

1. 동영상 검색 프로우차트

위와 같은 명령으로 동영상의 한 프레임씩 메모리에 불러 비트맵으로 정의된 HANDLE szPal, HANDLE szDib, LPSTR lpDib 등, 각 파라미터 정보들을 메모리에 저장하여 동화상의 각 프레임의 정보들과 비교, 검색 질의할 통하여 검색하고자 하는데 이 프로우차트는 그림 1과 같다.

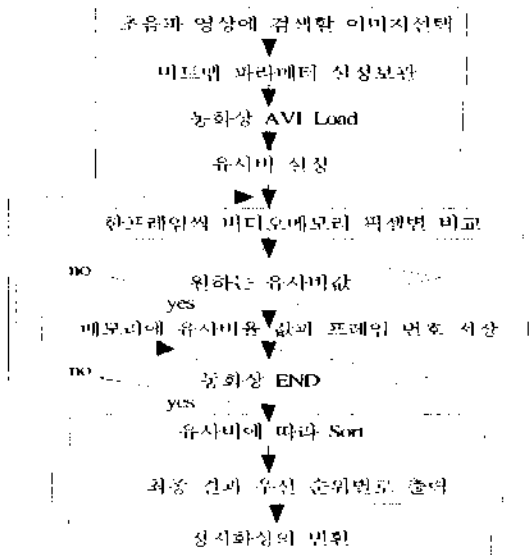


그림 1. 동화상검색 프로우차트
Fig. 1 Flow chart of video retrieval.

2. 동영상 비교검색 프로그램

위에서 정의된 내용에 의하여 구현된 비교, 검색 프로그램은 다음과 같다.[4][5]

```
typedef struct {
    DWORD PixelID;          // 픽셀당 비트수
    DWORD MovieExtentX;    // 프레임의 X값
    DWORD MovieExtentY;    // 프레임의 Y값
    DWORD Start_Position;  // 비교 시작점
} Comp_Pixel_Data;

typedef struct {
    DWORD Frame_Position; // 해당 프레임 번호
    DWORD Comp_Value;     // 프레임의 비교 값
} Comp_Frame_Data;

AviFrameComp :: AviFrameComp()
// AviFrameComp Class 생성자
{
    Comp_Frame_Data Comp_Data_Map[mplInfo.dw-
        TotalFrames]; // 프레임 비교자료 Map Create
    Comp_Pixel_Data Comp_Position_Start;
    // 프레임 비교 시작점 Struct 선언
    Map_Count = 0;
    // 비교 결과 Map Counter Clear
}

void LoadingInformation(MMPID idMovie)
{
    MMPMOVIEINFO mpInfo;
    mpInfo.chFullMacName[0] = 0;
    mmpGetMovieInfo(idMovie, &mpInfo);

    Comp_Position_Start.PixelID = mpInfo.dwPixelDepth;
    // 취득된 픽셀당 비트수를 저장
    Comp_Position_Start.MovieExtentX =
        mpInfo.dwMovieExtentX;
    // 취득된 프레임의 x축 픽셀수 저장

    Comp_Position_Start.MovieExtentY =
        mpInfo.dwMovieExtentY;
    // 취득된 프레임의 Y축 픽셀수 저장
}

AviFrameComp::CompStram(HPALETTE hhClipPalerte)
    HPALETTE hpalette;
    DWORD CompValue_V1, CompValue_V2,
        CompValue_V3;
    hpalette = mpGetPalettehandle(idMovie);
    if(hpalette == NULL )
    // AVI 파일의 해당 프레임의 핸들을 가져온다.
    {
        MessageBox(hWnd, 'Could Not Access Palette
            Information !.', szAppname, MB_OK);
    }
}

```

```
// 프레임의 핸들을 가져오지 못하면 Error Message를
// 보여준다
return;
}
Comp_Position_Start.Start_Position = 0;
// 두 화상의 비교 시작점을 처음 시작점인 0으로 설정
// 한다.
// PixelComp함수에 Load된 프레임의 핸들과 비교 대
// 상인 화상의 핸들, 비교 시작 지점의 Y축 값을 보낸다.
// Y축 값 * X축 값 * 해당 픽셀당 비트수 = 검색 시
// 작 위치값
CompValue_V1 = PixelComp(hpalette, hhClipPalerte,
    Comp_Position_Start );
Comp_Position_Start.Start_Position =
    (int)(mpInfo.dwMovieExtentY / 2);
CompValue_V2 = PixelComp(hpalette, hhClipPalerte,
    Comp_Position_Start );
Comp_Position_Start.Start_Position =
    mpInfo.dwMovieExtentY ;
CompValue_V3 = PixelComp(hpalette, hhClipPalerte,
    Comp_Position_Start );

CompValue_Ave = (CompValue_V1 + CompValue_V2
    + CompValue_V3 ) / 3;
Comp_Data_Map[Map_Count].Frame_Position =
    mmpGetCurFrame(idMovie);
Comp_Data_Map[Map_Count++].Comp_Value =
    CompValue_Ave;
}

```

AviFrameComp Class의 Class함수인 AviFrame-Comp :: CompStram(HPALETTE hhClipPalerte)는 두 이미지간에 픽셀별 자료 비교에 따라 비교 결과를 Map에 저장하는 함수이며, 처리 내용은 다음과 같다.[6]

- 1) 검색을 위하여 Memory에 저장된 해당 프레임의 핸들을 취하고, 핸들을 가져 오는 중 문제가 발생하면 Error Message를 출력한 후 Return한다.
- 2) 화상 비교를 위하여 첫 번째로 시작점부터 X축의 수만큼의 비교하기 위하여 시작점을 0으로 설정하고
- 3) Pixel별 비교를 수행한다.
- 4) 두 번째 비교 과정을 위하여 프레임의 중간 값을 취하기 위하여 Y축의 값을 2로 나누어 정수화 시킨 후 Y축 위치를 설정하여
- 5) Pixel별 비교를 수행한다.
- 6) 마지막 비교 과정을 위하여 프레임의 마지막 Y축의 값을 설정
- 7) Pixel별 비교를 수행한다
- 8) 위 처리 과정중 3, 5, 7의 수행에 따라 구하여진 CompValue_V1, CompValue_V2, CompValue_V3에 값의 평균을 구한다.
- 9) 각각의 처리 과정을 통하여 구하여진 값을 해당 프레임의 번호와 함께 Map에 저장한다.

IV. 실험결과

이상에서 서술한 동화상 검색 시스템의 실험은 IBM-PC (펜티엄II 266MHz) Win95환경에서 검색엔진부분은 Microsoft Visual C++ 5.0을 사용하고 사용자 인터페이스 환경은 Visual Basic 5.0기을 이용하였다.

비디오 보드는 DVD용 멀티미디어 통합 VGA보드 Winx perfect IV를 사용하였고, 먼저 검색화상을 얻기 위해 소 크린 캡처 프로그램 AVIEdit를 사용해서 여러 개의 정지 화상을 얻어 본 논문에서 제안한 프로그램을 구동하였다.[8][9]

이에 의하여 구현된 메인 화면은 그림 2와 같다.

여기에서 첫 번째로 Menu를 선택하여 사용자가 검색을 원하는 정지화상(그림 3)과 같이화면에 띄워놓고 두 번째로 검색을 원하는 동영상을 골라 선택하여 그림 4와 같이 설정한다. 그리고 사용자가 원하는 유사비를 입력하여 원하는 프레임만 메모리에 저장하게 하였다.

여와 같은 이유는 전체 프레임의 유사비를 전부 메모리 Dimension에 보관하면 비효율적이기 때문에 사용자가 원하는 유사비를 입력시켜 원하는 유사비 이상의 프레임만 메모리에 저장하게 하였다. 기본 설정이 끝나면 화면 우측 하단의 검색 시작 버튼을 눌러 검색을 시작한다. 동화상 실행 중에 검색을 위하여 Memory에 저장된 해당 프레임의 핸들을 취하고, 검색 첫 번째로 시작점부터 X축의 수만큼의 비교를 위하여 시작점을 0으로 설정한 후 Pixel별 비교를 수행한다.

두 번째 비교 과정을 위하여 프레임의 중간 값을 취하기 위하여 Y축의 값을 2로 나누어 정수화 시킨 후 설정 후 Pixel별 비교를 수행하고, 마지막 비교 과정을 위하여 프레임의 마지막 Y축의 값을 설정해서 Pixel별 비교를 수

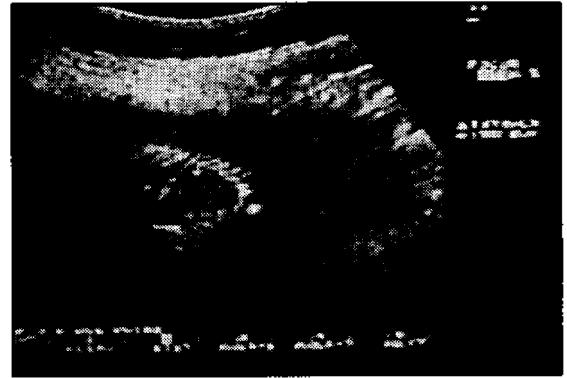


그림 3. 정지화상 파일 Load 화면
Fig 3 Load screen of picture file



그림 4. 동영상 파일 Load 화면
Fig 4 Load screen of video file.

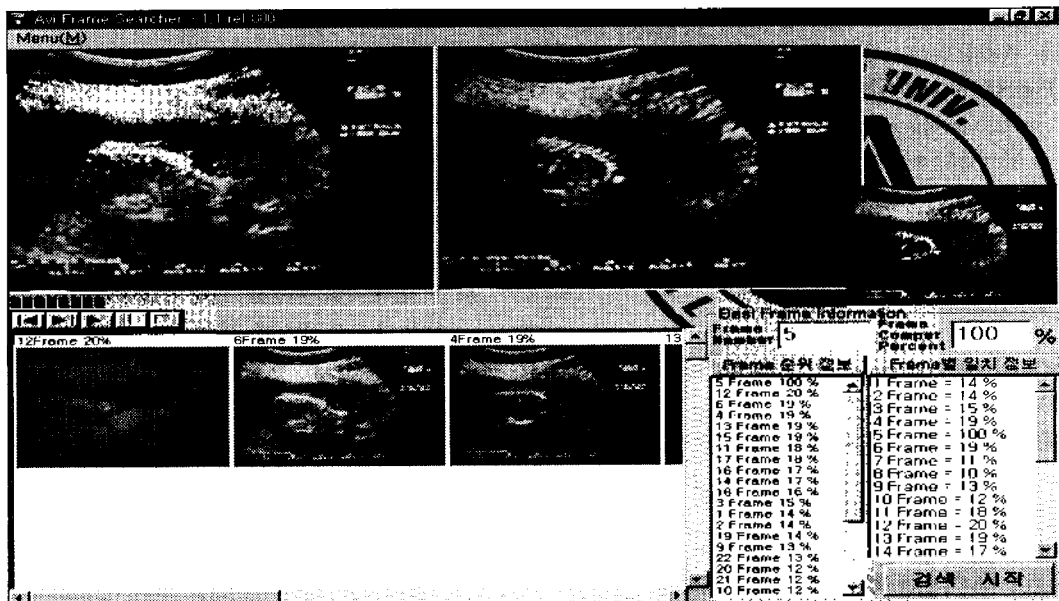


그림 2. 동영상 검색 프로그램 메인 화면
Fig. 2 Main screen of video retrieval program.

행한다. 위 처리 과정중 수행에 따라 구하여진 CompValue_V1, CompValue_V2, CompValue_V3에서 구해진 Pixel별 비교값 평판을 구한 다음, 각각의 처리과정을 통하여 구하여진 값을 해당 프레임의 번호와 함께 Map에 저장한다. 이 Map에 저장된 파라메터를 화면에 동시에 그림 5와 같이 화면 우측에 디스플레이 하여 좌측에 유사비에 따라 sort된 결과가 표시되고 우측에 각프레임 유사비가 순서대로 나타나게 해서 사용자가 알기 쉽도록 하였다.

Best Frame Information	
Frame Number	5
Frame Compar Percent	100 %
Frame 순위 정보	Frame별 일치 정보
5 Frame 100 %	1 Frame = 14 %
12 Frame 20 %	2 Frame = 14 %
6 Frame 19 %	3 Frame = 15 %
4 Frame 19 %	4 Frame = 19 %
13 Frame 19 %	5 Frame = 100 %
15 Frame 19 %	6 Frame = 19 %
11 Frame 18 %	7 Frame = 11 %
17 Frame 18 %	8 Frame = 10 %
16 Frame 17 %	9 Frame = 13 %
14 Frame 17 %	10 Frame = 12 %
18 Frame 16 %	3 Frame = 15 %
3 Frame 15 %	1 Frame = 14 %
1 Frame 14 %	2 Frame = 14 %
2 Frame 14 %	19 Frame = 14 %
19 Frame 14 %	9 Frame = 13 %
9 Frame 13 %	22 Frame = 13 %
22 Frame 13 %	20 Frame = 12 %
20 Frame 12 %	21 Frame = 12 %
21 Frame 12 %	10 Frame = 12 %
10 Frame 12 %	
검색 시작	

그림 5. 프레임별 번호와 유사비 화면
Fig. 5 Frames NO & comparing rate screen.

동화상 검색이 끝나면 map에 저장된 유사비에 따라 가장 유사비가 높은 프레임 번호부터 화면 아래에 프레임 번호와 유사비가 출력되도록 그림 6과 같이 출력하였다.

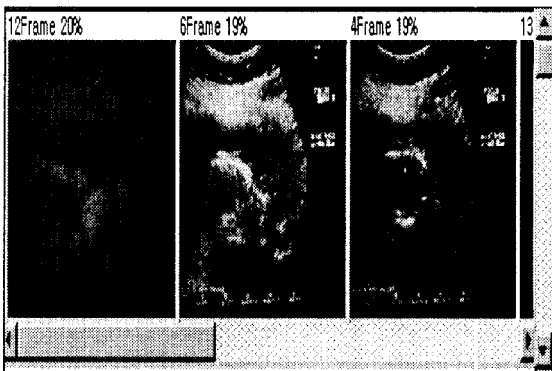


그림 6. 유사비율에 의한 검색된 화면
Fig. 6 Retrieved screen by comparing rate.

여기에 나타나는 화면중에 검색을 원하는 정지화상이 있으면 화면 중에 원하는 화상을 마우스로 선택하여, 그 프레임부터 다시 동화상을 Replay 시켜 이용할 수 있게 하

였다. 검색한 결과로 원하는 화상과 100%일치한 프레임은 검색하였고 연결되는 프레임에 화상이 조금만 바뀌어도 20%이하로 일치율을 보였고 검색위치가 바뀌면 2%이하로 나타났다. 여기에 선택된 초기화면은 그림 7의 좌측과 같은 이렇고 잡음이 많은 화면이 캡처되어 보기가 어려웠다.

첫 번째 작업으로 하단의 도구중 MAP을 설정하여 Inverter를 실행결과 그림 7의 우측과 같은 화면이 나와 설명하지는 않지만 밝은 화면이 구현되었다.

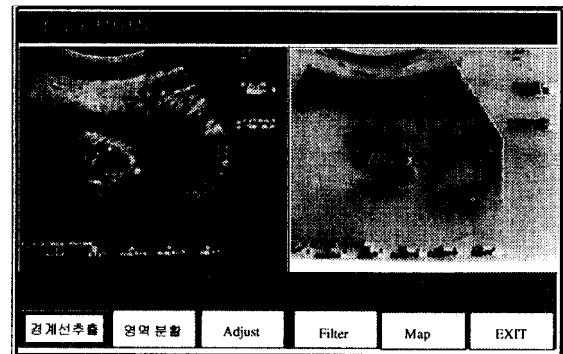


그림 7. 초음파 원 영상이 Inverter된 화면
Fig. 7 Inverted screen from original ultrasonogram.

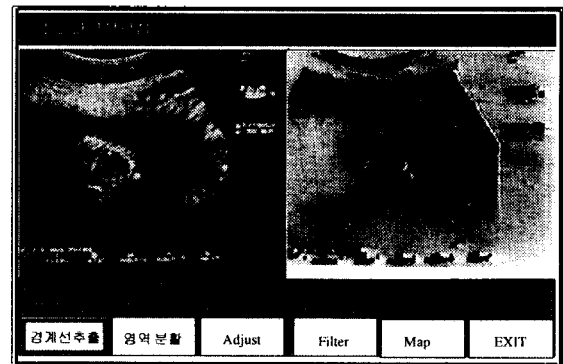


그림 8. Inverter된 화면을 Equalize 시킨 화면
Fig. 8 Equalize screen from inverted screen.

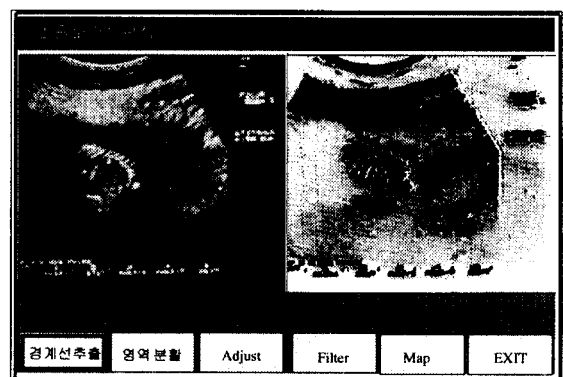


그림 9. Equalize된 화면을 Adjust 시킨 화면
Fig. 9 Adjust screen from equalized screen.

선명하지 않은 우측화면을 다시 Equalize하여 최적화 시킨 화면이 그림 8의 우측 화면으로 명암이 뚜렷하게 나타났다. Equalize된 화면을 다시 도구인 Filter와 Adjust를 사용하여 Shapen과 Noise, Brightness를 구현해서 명도의 대비 등으로 보다 명확한 화면 그림 9를 얻게되었다.

V. 결 론

본 논문은 초음파 진단장치에서 획득한 동영상에서 사용자가 원하는 화상이 그 초음파 화상 File에 존재하는지도 존재하면 그 위치가 어디에 있는지를 알아내고, 그 위치부터 시작하도록 설계하였다. 구현방법은 초음파 동화상을 AVI file로 변환 출력 시켜 동영상에서 사용자가 원하는 화상을 얻기 위하여 장치화상을 Bitmap으로 구성하고 각 프레임에 지정된 메모리 데이터와 픽셀별로 비교하여, 두 이미지간에 비교결과를 Map에 저장하였다. 이 저장된 결과를 비율별로 sort하여서 화면에 출력시켜 원하는 화상이 출력되어 있으면 그 화면을 선택해서 원하는 프레임 번호를 얻을 수 있어서 그 프레임부터 재생할 수 있었다.

이 실험 중에서 여러 초음파 동화상 File을 검색하였지만 대부분 제한된 검색에서 프레임 유사비(그림 6)에 나타난 것과 같이 화면이 조금만 바뀌어도 유사비는 20% 이하가 나와 원하는 정지화상이 검출되어 빠르게 검색할 수 있었다.

그러고 검출된 화면을 진단자가 쉽게 볼 수 있도록 여러 가지 영상처리를 하여 내부구조를 뚜렷하게 나타내었다. 이 구현방법은 정지화면의 3줄(맨 위의 한 라인, 중간라인, 맨 마지막라인)만 가지고 검색대상으로 삼아 그 안에서 조금만 대상을 찾는다면 이웃한 프레임과 유사비가 비슷하여 많은 프레임이 검출되었다. 이 점을 보완하기 위해 앞으로 검색대상을 X형태로 검색하면 검색시간은 약간 길겠지만 유사비가 많이 검출되는 것은 막을 것으로 생각된다.

참 고 문 헌

1. 이 승우, "초음파 진단장치", 전자공학회지, p267-281, 1996. 3.
2. 방사선과 교수협의회, "방사선 촬영학", 대학시립, 1994.
3. 이 상엽, "Visual C++ Programming Bible Ver 5.X", 영진출판사, 1997.
4. 이 정근, "Microsoft Multimedia Development Kit", 도서출판 세운, 1996.
5. Microsoft, "Microsoft Windows software Development Kit", Microsoft, 1997.
6. Ioannis Pitas, Digital Image Processing Algorithms, Prentice Hall, 1993.
7. 주 경민외 2인, "Visual Basic Programming Bible Ver 5", 영진출판사, pp.255-274, 1997.
8. Jon Peddic, "Multimedia Graphics Controllers", Windcrest, 1994.
9. James D. Murray & William Varryper, "Graphics file format second edition", O'Reilly & Associates Inc, 1996.

▲오 태 석(Teh Sok Oh)



1991년: 조선대학교 대학원 컴퓨터공학과(공학석사)
 1997년: 조선대학교 대학원 컴퓨터공학과 박사수료
 1992년~ 현재: 목포과학대학 신지과 조교수
 ※주관심분야: 마이크로프로세스, 멀티미디어 시스템

▲오 무 송(Moo Song Oh)



1965년: 조선대학교 공과대학 전기공학과 졸업(공학사)
 1969년: 조선대학교 대학원 전기공학과(공학석사)
 1965년~1982년: 조선대학교 공업전문대학 전기과 교수
 1983년~ 현재: 조선대학교 컴퓨터공학과 교수

※주관심분야: 멀티미디어 시스템, 영상처리