

畫像處理技法을 利用한 變形率解析에 관한 研究

(第1報 마이크로 컴퓨터를 利用한 畫像處理技法의 開發)

白 仁 煥* · 辛 玫 教**

A Study on the Strain Analysis by Image Processing Technique

(Part 1: Development of Image Processing Technique with Microcomputer)

I. H. Paik, M. K. Shin

Abstract

The image processing system consisted of the microcomputer IBM PC-XT and the graphic board (16 gray level and 640×400 pixels resolution) has been proposed, and the image processing softwares programmed in the BASIC and in the assembler language have been developed.

The programs are consisted of the main menu and the sub menu, that have contained the subroutine for the capture of image data, the determination of region, the histogram, the change of value, the montage, the skeleton, the mask, the moving, the zoom, the disk access and the print.

For the application, the photoelastic fringe data have been captured and analyzed. It was seen that the programs are available for the image processing.

I. 序 論

최근 전자 공학의 발전과 더불어 컴퓨터의 보급이 가속화됨에 따라서 컴퓨터에 의한 화상 처리 시스템의 활용이 크게 늘어나고 있다. 즉 감시, 검사, 인식, 선별, 운동해석, 계측 등을 화상 처리함으로써 위험한 작업의 무인화 및 성력화에 도움을 주고 있다. 그리고 사람의 눈으로 하던 작업을 컴퓨터로 하여금 처리하게 하고, 사람 손으로 불가능하던 처리법 예컨대 CT단층촬영과 같이 종래에는 볼 수 없던 것을 가시화하기도 하며, 주사 모아레법과 같이 특수한 계

법측도 화상 처리법을 이용함으로써 가능하게 되었다.

마이크로 컴퓨터를 이용한 화상 처리시스템에는 전용 프로세서를 내장한 화상 처리 전용기가 있는데 처리 속도는 빠르지만 고가이고 범용성이 없다. 또 컴퓨터 메모리와는 별도로 외부에 화상처리 유닛을 장착한 화상 처리 시스템¹⁻⁴⁾이나, 컴퓨터 내부의 메모리를 화상메모리로 이용하는 화상 처리 시스템⁵⁾ 등이 있으며, 화상처리 프로그램으로는 C언어 또는 어셈블리어언어로 된 것이 개발되고 있다.

본 연구에서는 16 bit 512 KB 메모리를 가진

* 正會員, 부산대학교 공과대학

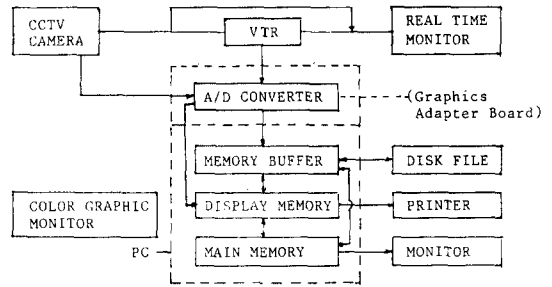
** 正會員, 한국해양대학

IBM PC-XT를 기본으로 하고 화상 메모리 보오드(16 Gray Level 640×400 Pixels Resolution)를 추가하여 시스템을 구성하였으며, 처리 속도를 높이기 위하여 어셈블리 언어로 서브루틴을 작성하여 대화형 프로그램으로 만들었으며, 그 프로그램의 활용 예로서 광탄성 실험에서 얻어진 화상데이터를 처리하는 데 적용하여 보았다. 즉 본보에서는 화상처리 시스템의 구성, 프로그램의 구성 및 알고리즘, 활용예를 검토한다.

2. 시스템의 構成

본 연구에서 구성한 화상 처리 시스템의 블록선도와 사용기기의 제원은 Fig. 2.1과 같다. 화상을 취득하기 위한 카메라는 폐쇄회로 카메라(Closed Circuit Video Camera: CCTV Camera)로서 EIA RS-170형을 사용하였다. 카메라에서 취득한 화상을 모니터(REAL TIME Monitor)로 상태를 확인하면서 VTR에 녹화시킨다. A/D변환기로는 Chorus Data System Inc. 에서 고안한 그래픽스 어댑터(Graphics Adapter Board)를 이용하였으며¹⁰⁾, 카메라 또는 VTR로부터 받은 아날로그 영상신호를 디지털 신호로 바꾸어 FIFO 메모리 버퍼에 일시적으로 저장하고 DMA(Direct Memory Access) 또는 PIO(Programmed Input/Output) 방식에 의하여 마이크로 컴퓨터의 메인 메모리(Main Memory)나 화상메모리(Display Memory)에 전송한다¹⁰⁾. 이때 화면의 프레임 및 농도 등급(Gray Level)은 소프트웨어로 제어된다¹⁰⁾. 화상 메모리에 화상 자료가 들어오면 그래픽 모니터(Color Graphic Monitor)에 디지털 화상이 나타난다. 본 연구에서 사용한 그래픽스 어댑터는 640×512화소(Pixel)를 64등급(6bit/pixel)까지 디지털화 할 수 있으나, 마이크로 컴퓨터 메모리의 제한과 모니터의 해상도(Resolution)의 제한 때문에, 640×400화소의 해상도와 16농도 등급(4bit/pixel)으로 화상 데이터를 취득한다.

마이크로 컴퓨터는 IBM PC-XT이며 메모리를 확장하여 512K Byte RAM으로 되어 있다. 화상 메모리는 A000:xxxx 번지로 할당되어 있으며



(a) Schematic Diagram

- CCTV CAMERA (RCA RS-170)
- VTR(RCA TC1214)
- REAL TIME MONITOR(SONY Trinitron)
- MICRO COMPUTER(IBM XT: 512KB RAM)
- MONOCHROME MONITOR(HERCULES)
- COLOR MONITOR(for TECMAR GRAPHIC MASTER: 16 GRAY LEVEL, 640×400 RESOLUTION)
- DOT MATRIX PRINTER(EPSON FX-80+)

(b) Specifications of Equipments

Fig. 2.1 Schematic diagram and Equipment of Image Processing System

메인 메모리 4000:xxxx-5000:xxxx번지를 화상 데이터 저장 영역으로 잡아둔다. 640×400 화소를 4bit/pixel(2 Pixels/Byte)로 저장하기 때문에 128KByte의 메모리가 필요하다.

프로그램의 메뉴 표시 등 프로그램의 작동에 필요한 자료의 표시를 위하여 모노크롬 모니터를 이용하였으며 프린트는 도트매트릭스 프린터를 이용하였다.

한편 화상 모니터는 Tecmar Graphic Board와 함께 사용하는 칼러 모니터를 이용하였으며, 화상 메모리(Display Memory)에 화상 데이터를 전송할 때는 03DAH 번지를 Port로 하여 1, 2, 5, 6, 9, 10...번째 행의 경우는 1FH를 OUT시키고, 3, 4, 7, 8, 11, 12...번째 행의 경우는 5FH를 OUT시키고 데이터를 보낸다.

3. 畫像 處理 소프트웨어

3.1 화상 처리 프로그램의 개요

화상 처리에서는 많은 양의 데이터를 취급하

고 처리해야 하기 때문에 처리 속도를 빠르게 할 필요가 있다. 그래서 고속을 요하는 처리의 서브루틴은 모두 어셈블리 언어로 작성하였으며 MENU부분은 BASIC으로 작성하여 DOS 상에서 LINK시켜 하나의 프로그램을 만들었다. 또 활용의 편의를 위하여 배치 파일(Batch File)로 만들어 PIFI (Program of Image Processing for IBM PC-XT)로 명명하였다.

본 프로그램은 메뉴(MENU)를 통한 대화형으로 짜여 있으며 프로그램을 실행시키면 메뉴로서 처리하고자 하는 내용이 화면에 표시되며, 원하는 내용의 번호를 입력하면 서브 메뉴(SUB MENU)가 나타나고 해당 서브 메뉴를 택하면 BASIC 프로그램에서 어셈블리 서브루틴(Assembly Subroutine)을 호출하여 원하는 처리를 수행하게 된다. 처리가 끝나면 서브 메뉴 또는 메인 메뉴로 돌아오므로 다른 처리를 수행할 수 있다. 메인 메뉴와 프로그램 내용은 Fig. 3.1에 제시하였다.

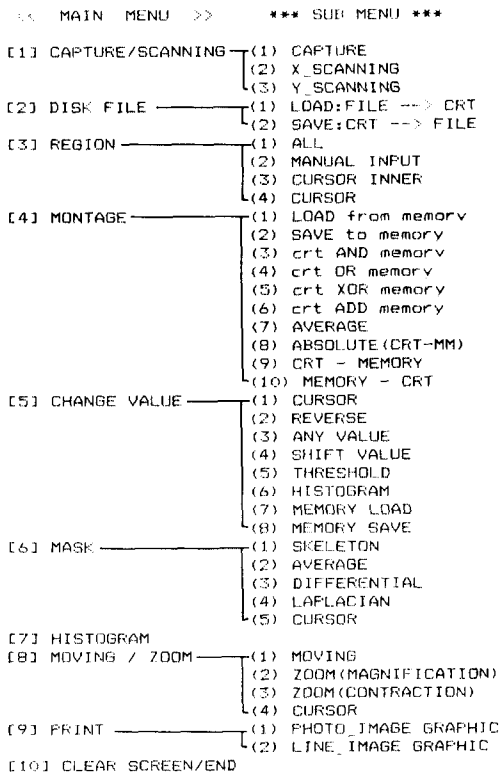


Fig. 3.1 List of Main MENU and Sub MENU.

각 서브루틴의 처리 시간은 처리 영역의 크기에 따라서 다소 차이는 있으나 한 화면 전체를 처리하는데 1초 이내인 것이 많으며, 프린트와 세션화를 제외하면 대부분 수초 이내에 처리가 완료된다. 중요한 화상 처리의 알고리즘과 프로그램의 구성에 대하여 기술한다^{5, 14, 15}.

3.2 화상의 인식과 취득(CAPTURE)⁶⁻¹⁰⁾

카메라(CCTV Camera)는 PC와 연결되어 있어서 영상신호를 계속 보내고 있는데, 화상 취득 프로그램을 가동시키면 카메라에서 출력된 아날로그 영상 신호는 PC의 입력 단자(Input Port) 310 H번지로 입력되어 A/D 콘버터에서 6Bit로서 64농도 등급의 디지털 신호로 바뀌고 화상 모니터에 전송되어 화상으로 나타난다.

백색 등급(White Level)과 흑색 등급(Black Level)의 설정, 화상 프레임(Left Margin, Top Margin, Horizontal Pixels, No. of Lines)의 구성, 스로틀 카운트(Throttle Count)등은 프로그램에서 파라미터로서 입력시킨다.

백색 등급은 A/D콘버터에서 기준 전압의 최대치로 하며, 입력된 영상 신호가 이 값보다 크거나 같으면 백색으로 변환된다. 또 A/D콘버터는 64농도 등급으로 디지털화시키므로 본 프로그램에서는 0~3을 흑색(Black, 0), 4~7은 청색(Blue, 1)···, 60~63은 백색(White, 15)으로 변환시켜, 16농도 등급의 슈도칼러(Pseudo Color)로서 모니터에 출력시킨다.

화상 프레임은 640×400화소를 최대치로 되게 결정한다.

EIA RS-170 카메라에서는 교호 주사 방식(Interlaced Scanning)으로서 비디오 신호는 매 초당 30프레임을 형성한다. 또 본 시스템에 이용된 디지털이저는 비디오 데이터 1라인을 디지털 신호로 바꾸는데 약 51.2μsec정도 소요하며 메모리에 디지털 데이터를 전송하는 시간, 비디오 신호와 컴퓨터 메모리 사이의 동조에 포함되는 지연 시간 등을 고려하면 640×400×16 해상도의 경우, 한 라인을 완전히 디지털 신호로 바꾸고 전송시키는데 약 320μsec 걸린다. 따라서

한 라인의 메모리 전송이 끝날 때까지 다른 라인의 취득(Acquisition)이 지연되어야 한다. 이 경우 320 μ sec 메모리 전송 간격이 겹치지 않고 다음 취득이 가능한 가장 가까운 라인은 11번째 라인이다. 이것이 최소의 스토틀 카운트이며 이것보다 큰 값은 허용되고 253라인까지는 지연되어도 좋다. 또 스토틀 카운트가 낮으면 화면이 찌그러지므로 본 시스템에는 200을 쓴다.

한편 이상에서 얻어진 16농도 등급의 디지털 신호는 PIPO 메모리 버퍼에 일시적으로 저장되고 DMA 방식이나 PIO 방식에 의하여 PC의 메인 메모리나 화상 메모리에 전송된다. PIO는 각 화소를 읽어 들이고 기록할 때마다 CPU의 통제를 받기 때문에 DMA에 비하여 많은 시간이 걸린다. DMA는 CPU의 통제를 받지 않고 DMA 컨트롤러에 의하여 자동적으로 전송한다. 본 연구에서는 속도가 빠른 DMA 방식에 의하여 화상 메모리(A000:0000H~A000:F9FFH)로 전송하여 화상을 얻는다. 이 경우 한 화면을 얻는데 약 2초가 소비된다.

3.3 영역 지정

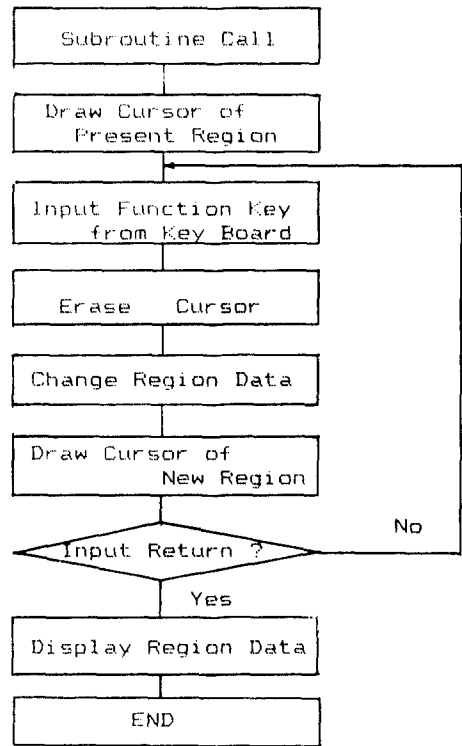
해석하고자 하는 영역의 범위를 지정하여 그 지정된 영역만을 처리함으로써 처리 시간을 절약하고 부분 처리를 가능하게 하는 것으로 모든 처리의 초기화 과정에 해당된다. 영역은 정방형으로 이루어지며 좌상점의 좌표와 우하점의 좌표(또는 수평 및 수직 화소수)로써 지정하며, 해당 변수의 번지에 좌표값을 넣어 주는 것이다. 전 영역(ALL), 좌표치를 키보드로 넣어 주는 방식(MANUAL INPUT), 커서를 나타내고 특수 키를 조작시켜 영역을 시각적으로 변경시키는 방식(CURSOR INNER) 등이 있으며, 커서를 화면에 표시하여 지정된 영역을 확인하는 방식(CURSOR)도 이 메뉴에 포함시키고 있다.

ALL은 변수의 초기화에 의하여 이루어지며, MANUAL INPUT은 변수의 번지에 좌표치만 넣어주면 되므로 BASIC프로그램에서 POKE함수를 이용하였으며, 지정된 영역의 좌표를 읽는 것도 PEEK함수로 읽어서 화면에 표시하게 하

였다.

또 커서의 표시는 영역의 가장자리에 해당되는 직선상의 원화상을 반전(REVERSE)시킴으로써 나타낼 수 있다.

CURSOR INNER에 의한 영역 지정은 먼저 커서를 나타내고, 키보드에서 *i, j, k, m*을 입력시킴으로써 좌표치의 증감과 동시에 커서가 이동되도록 어셈블리 서브루틴을 작성하였으며, 시각적으로 확인된 영역이 적당할 때 리턴 키를 눌러 영역을 확정하면 좌표치를 모니터에 표시하고 메뉴로 복귀한다. 커서 이동의 플로우 차아트와 입력 키의 기능을 Fig. 3.2에 나타내었다.



- i*: Upper Cursor, Up-Ward
- I*: Upper Cursor, Down-ward
- j*: Left Cursor, Left-ward
- J*: Left Cursor, Right-ward
- k*: Right Cursor, Left-ward
- K*: Right Cursor, Right-ward
- m*: Lower Cursor, Up-ward
- M*: Lower Cursor, Down-ward

Fig. 3.2 Flow Chart and Function Key for Moving Cursor

3.4 화상의 합성(MONTAGE)

모니터에 나타난 화상 즉 화상 메모리의 화상 데이터와 메인 메모리에 저장된 화상 데이터 사이에 대응되는 화소끼리 연산처리를 행하여 새로운 화상으로서 모니터에 출력시키는 과정이다. 화상 합성의 연산 처리는 두 메모리의 대응하는 좌표점의 화상농도를 읽고 논리 연산(AND, OR, XOR) 또는 산술연산(ADD, AVERAGE, CRT sub MM, MM sub CRT 등)을 행하여 화상 메모리의 해당 빈자에 넣는 것이다. 이 처리를 이용하면 움직이는 화상 또는 시간에 따라서 변화하는 화상의 특성을 해석할 수 있다.

이 처리를 위하여 화상 메모리에서 메인 메모리로 화상을 보내거나 (SAVE to memory), 메인 메모리에서 화상 메모리로 화상을 전송(LOAD from memory)할 필요가 있다. 화상의 합성에 관한 서브루틴의 프로그램 순서는 다음과 같다.

- (1) 합성할 화상을 하나는 메인 메모리에, 다른 하나는 화상메모리에 저장한다.
- (2) 처리할 영역을 설정한다.
- (3) 대응되는 빈지의 화소를 하나씩 따로 읽어낸다.
- (4) 읽어낸 화소의 농도치를 연산 처리한다.
- (5) 처리된 화소 데이터를 화상 메모리의 동일 빈지에 넣는다.
- (6) (3)~(5)을 영역 내의 모든 화소에 대하여 반복 적용한다.
- (7) 끝나면 메뉴로 돌아간다.

3.5 화상 농도의 변환

농도를 변환시키는 처리는 화상 메모리에 있는 각각의 화소 자체를 처리하는 것이다. 먼저 메인 메모리에 저장(MEMORY SAVE)시켜 두고 농도 분포를 확인한다(HISTOGRAM). 하나의 화소를 메모리 버퍼에 로드시켜 반전(REVERSE) 추출(ANY VALUE), 농도 증감(SHIFT VALUE), 2치화(THRESHOLD) 등의 조건에 따라서 처리한 후 원래 메모리 빈지에 저장하며 처리

영역 내의 모든 화소에 대한 처리가 끝날 때 까지 반복한다. 이 때 하나의 메모리 내에 2개의 화소가 들어있으므로 분리시키고 각각에 대하여 처리하고 다시 묶어서 메모리 속에 넣어야 한다.

REVERSE : 화상의 농도치를 I 라고 할 때 그것을 발견시킨 값(15-I)를 데이터로 한다.

ANY VALUE : 어떤 농도치를 입력하면 그 농도치의 화소만 그냥 되돌리고 다른 농도치의 화소는 모두 0으로 하여 지워 버린다.

SHIFT VALUE : 어떤 농도치의 화소에 대하여 주어진 수 만큼 농도치를 증감한다. 이 처리에 의하여 화상의 콘트라스트가 개선될 수 있다.

THRESHOLD : 어떤 기준치보다 작은 농도치의 화소를 0으로 하고 그 이상이면 15로 하여 흑백 2가지 값만으로 나타낸다.

한편 여기서 이용되는 농도 분포 서브루틴(HISTOGRAM)은 각 화소의 농도를 인식하고 각 농도의 값을 가진 화소수를 헤아리는 과정과 농도 분포를 모니터에 나타내는 과정이 있다. 헤아리는 과정은 어셈블리 서브루틴(HISTO)으로 작성하였으며, 그 도수를 각 농도 변수의 메모리에 저장하여 둔다. BASIC 프로그램에서 이것을 PEEK함수로 읽어내고 총합과 백분율을 구하여 막대 그림으로 표시한다.

3.6 마스크 처리(MASK)^{7,8)}

일반적으로 마스크 처리는 주변의 화소를 고려하여 처리하는 것이며 주목하고 있는 화소와 그 주변의 몇 개 화소 데이터에 가중치를 곱하여 더하는 연산인 것이다. 이 가중치를 적당히 선택함으로써 노이즈를 없애기도 하고 농도 경계를 검출할 수도 있다. 이 때문에 마스크를 공간 필터라고 한다. 주목 화소와 근방의 8개 화소를 합한 9개 화소 즉 3×3화소에 대하여 처리하는 것이 보통이며 Fig. 3.3과 같이(i, j) 점의 화소 농도치를 A(i, j), 가중치를 W(i, j)라고 할 때 처리 후의 농도치 A'(i, j)는 다음 식으로 표시된다.

$$A'(i, j) = \sum_{k, l=-1}^1 W(i+k, j+l) \times A(i+k, j+l) \tag{3.1}$$

본 연구에서는 3×3화소에 대하여 마스크 처리를 행하여 평활화(AVERAGE), 1차미분(DIFFERENTIAL) 수평 방향, 수직 방향, 2차 미분(LAPLACIAN), 세선화(SKELETON)등의 서브루틴을 어셈블러로 만들어 화상 처리에 이용한다.

$A(i-1, j-1)$	$A(i, j-1)$	$A(i+1, j-1)$
$A(i-1, j)$	$A(i, j)$	$A(i+1, j)$
$A(i-1, j+1)$	$A(i, j+1)$	$A(i+1, j+1)$

(a) Value of Pixels

$W(i-1, j-1)$	$W(i, j-1)$	$W(i+1, j-1)$
$W(i-1, j)$	$W(i, j)$	$W(i+1, j)$
$W(i-1, j+1)$	$W(i, j+1)$	$W(i+1, j+1)$

(b) Weights for Pixels

Fig. 3.3 Mask of 3×3 Pixels

1	1	1
1	2	1
1	1	1

(a) Average

-1	-1	-1
-1	8	-1
-1	-1	-1

(b) Laplacian

1	0	-1
1	0	-1
1	0	-1

(c) Horizontal Differential

1	1	1
0	0	0
-1	-1	-1

(d) Vertical Differential

Fig. 3.4 Weights in 3×3 Mask

3.6.1 평활화(Mask average)

평활화는 Fig. 3.4(a)와 같이 주변의 8화소에 가중치 1을, 주목 화소에 가중치 2를 주어서 각 화소의 농도치와 곱하여 총합을 구하고 10으로 나눈 값을 주목 화소의 값으로 한다. 10으로 나눌 때 나머지는 버린다.

$$A'(i, j) = \left[\sum_{k, l=-1}^1 A(i+k, j+l) + A(i, j) \right] / 10 \tag{3.2}$$

이 처리는 주목 화소 보다 주위의 화소가 강조되므로, 주변에 비하여 아주 다른 특별한 화소점 (보통 이것을 노이즈라 한다)을 주변 화소들과 비슷하게 만드는 역할을 하게 된다. 전기적 센서로 부터 발생하는 노이즈는 일반적으로 랜덤(Random)하게 나타나며, 이러한 형태의 노이즈는 보통의 영상에서 볼 수 있는 완만한 변화들과는 대조적으로 화소와 화소 사이에 급격한 변화를 일으킨다. 이러한 극단적인 변화 즉 노이즈를 없애기 위하여 국부 평균 (Local average)이 이용되며 이것이 마스크 평균 처리이다.

3.6.2 미분(Mask differential)

화상의 농도 변화율을 구하기 위하여 수평 방향 또는 수직 방향으로 미분을 행하는 것이다. 1차 미분의 근사는 다음 식으로 주어진다.

$$\frac{\partial f(x, y)}{\partial x} \cong f(x+1, y) - f(x, y) \tag{3.3}$$

$$\frac{\partial f(x, y)}{\partial y} \cong f(x, y+1) - f(x, y) \tag{3.4}$$

이상의 근사를 3×3화소에 적용시키는 필터(Filter)로 다음 식이 이용된다.

$$\frac{\partial f(x, y)}{\partial x} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \tag{3.5}$$

$$\frac{\partial f(x, y)}{\partial y} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \tag{3.6}$$

따라서 수평 방향 미분은 Fig. 3.4(c)와 같은 가중치를 주어 더한 값의 절대치로 되며, 수직 방향 미분은 Fig. 3.4(d)와 같은 가중치를 주어 더한 값의 절대치로 된다.

한편 경계 강화(Edge Enhancement)를 하기 위한 필터링 기법의 하나로써 1차 미분을 이용할 때 다음과 같은 근사식을 쓴다.

$$\nabla f(x, y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j \tag{3.7}$$

식 (3.3)과 (3.4)에서 구한 x 방향 미분과 y 방향 미분 값을 이 식에 대입하면 근사치는 다음 식으로 표시된다.

$$A(i, j) = \left| \sum_{k=-1}^1 \{A(i-1, i+k) - A(i+1, j+k)\} \right| + \left| \sum_{l=-1}^1 \{A(i+l, j-1) - A(i+l, j+1)\} \right| \quad (3.8)$$

즉 x 방향 미분과 y 방향 미분의 절대치를 합한 것이 공간 미분으로 된다.

3.6.3 2차 미분(Laplacian)

경계를 강화시키는 필터링 기법의 하나로써 2차 미분의 근사식을 이용한다⁷⁾.

$$\begin{aligned} \nabla^2 f(x, y) = & \frac{1}{2} \left[\frac{\partial^2 f(x, y-1)}{\partial x^2} + 4 \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y+1)}{\partial x^2} \right] \\ & + \frac{1}{2} \left[\frac{\partial^2 f(x-1, y)}{\partial y^2} + 4 \frac{\partial^2 f(x, y)}{\partial y^2} + \frac{\partial^2 f(x+1, y)}{\partial y^2} \right] \end{aligned} \quad (3.9)$$

$$\frac{\partial^2 f(x, y)}{\partial x^2} = \frac{\partial f(x + \frac{1}{2}, y)}{\partial x} - \frac{\partial f(x - \frac{1}{2}, y)}{\partial x} \quad (3.10)$$

$$\begin{aligned} \nabla^2 f(x, y) = & f(x-1, y-1) + f(x, y-1) \\ & + f(x+1, y-1) + f(x-1, y) \\ & + 8f(x, y) + f(x+1, y) \\ & + f(x-1, y+1) + f(x, y+1) \\ & + f(x+1, y+1) \end{aligned} \quad (3.11)$$

즉 Fig. 3.4(b)와 같이 주목 화소에 가중치 8을, 주변의 화소들에 가중치(-1)을 곱하여 합한 것의 절대치를 주목 화소의 농도 값으로 하면 된다.

3.6.4 마스크 처리 알고리즘

마스크 처리는 주변의 화상 데이터와 함께 처리하기 때문에, 한 화소의 처리 결과는 바로 그 화소의 화상 데이터로 돌려놓을 수 없다. 뿐만 아니라 원래의 화상에서 두개의 화소가 동일 번

지에 들어 있으므로 분리시킨 후에 처리하여야 하며 처리된 결과는 두개씩 쌍으로 만든 후 메모리에 넣어야 되는 문제점이 있다. 그래서 화소를 분리하여 처리할 수 있도록 준비하고 처리된 화소를 임시로 저장할 수 있는 버퍼를 메인 메모리중에 설정할 필요가 있다.

본 연구에서는 Fig. 3.5와 같이 3000 : 0000H로부터 3000 : 0C7FH까지를 버퍼로 이용한다. 또 이 그림에서 데이터의 흐름을 표시하였으며, 프로그램의 구성은 다음 순서에 따라 이루어 진다

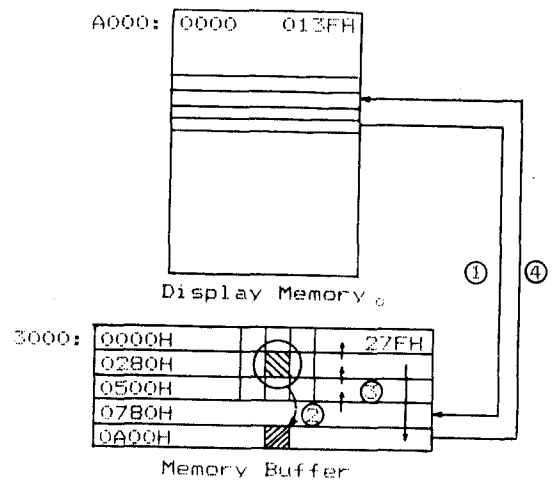


Fig. 3.5 Flow of Data in Mask

- (1) 화상 데이터를 메인 메모리에 저장해 두고 처리 영역을 정한다.
- (2) 제1행의 화소를 분리하여 3000 : 0780H번지 부터 옮겨 놓는다(①).
- (3) 버퍼 내에서 데이터를 ③과 같이 순차로 옮긴다. ①과 ③을 반복하여 3행의 데이터를 모두 옮긴다.
- (4) 3행의 데이터로 마스크 처리를 행하여 결과를 3000 : 0A00H번지 부터 넣어 둔다(②).
- (5) 이 결과에서 두 화소씩 짝을 지워 화상 메모리의 제2행에 옮겨 놓는다(④).
- (6) 제4행을 버퍼에 옮기고(①), 버퍼 내에서 전송하고(③), 마스크 처리하고(②), 화상 메모리에 전송한다(④).
- (7) 이상의 처리를 마지막 행까지 반복하고

끝에는 3000 : 0500H번지의 내용을 화상 메모리에 처리되지 않은 상태로 되돌린다.

3.6.5 세선화(Skeleton)

세선화는 2치화 된 흑백 화상의 넓은 선을 경계로부터 1화소씩 소거하여, 두께가 1화소로 이루어진 도형 즉 중심선을 구하는 것이다. 이때 한개의 화소(주목 화소)를 소거할 것인가를 결정하는 데는 주위의 화소가 어떠한가를 따져야 된다. 주위의 화소는 주목 화소의 상하 좌우 4개의 화소를 취하는 4근방법과, 이들 4개에 대각선 방향의 4개 화소를 더하여 8개의 화소를 취하는 8근방법이 있다. 이에 따라 여러가지 세선화 방법이 제안되어 있으나 여기서는 8근방의 연결로 세선화를 행하는 Hilditch의 방법⁷⁾을 개량하여 사용하였다.

프로그램의 알고리즘은 앞절에서 설명한 것과 같으나 Fig. 3.5의 ①에서 원 화상을 메모리 버퍼에 옮길 때, 선도형의 흑색(0)을 1로 하고, 배경을 이루는 백색(15)을 0으로 바꾼다.

이것으로 연결 조건을 조사하여 소거할 화소는 0으로 바꾸어 놓는다. 한 행의 화소에 대하여 조사가 끝나서 화상 메모리에 되돌려 넣을 때도(Fig. 35의 ④) 1은 0으로, 0은 15로 바꾸어서 넣는다.

세선화 처리를 위하여 먼저 연결수를 정의한다. 즉 주목 화소의 8근방 화소에 Fig. 3.6과 같이 번호를 붙이고 각각의 농도치를 A_i 라고 할때 연결수 C 는 다음 식으로 표시된다.

$$C = \sum_{i=1}^4 (1 - A_{2i-1}) \cdot \text{Max}(A_{2i}, A_{2i-1}) \quad (3.12)$$

여기서 $A_0 = A_1$ 이며

$\text{Max}(A_{2i}, A_{2i-1})$ 은 A_{2i}, A_{2i-1} 중 큰 값이다.

4	3	2
5		1
6	7	8

Fig. 3.6 Arrangement of Neighbor Pixels for Connectivity Evaluation

세선화 처리는 다음 조건을 모두 만족하는 점을 소거한다.

- 조건 1 : 현재의 농도치가 1이고 도형에 속해 있을 것
- 조건 2 : 8근방 중에서 2개 이상이 1일것(끝점의 보전)
- 조건 3 : 처리 후의 8근방 중에서 적어도 1개는 1일 것(고립점의 보전)
- 조건 4 : 연결수가 1일 것
- 조건 5 : 이미 주사된 4개의 연결점의 제거에 의하여 연결수가 변화하지 않을 것

이상의 조건을 조사할 때 각 화소마다 연결수를 계산하고 비교 처리하는 것은 시간이 많이 걸리므로, 3×3 매트릭스 내의 가능한 데이터 구성에 대하여 미리 계산된 결과를 테이블로 만들어 데이터 세그먼트에 넣어두고 이용한다. 이때 각 화소는 0 또는 1의 값으로 되어 있으므로 8근방이 취할 수 있는 패턴의 수는 $2^8 = 256$ 가지이다.

또 세선화 처리는 화면을 1회 주사할 때마다 번두리의 1화소씩 소거시키므로 굵은 선을 모두 1화소로 된 선으로 바꾸기 위해서는 화면의 주사를 반복해야 한다. 따라서 처리 시간이 많이 걸리며 처리 도중에 리턴키를 누르면 처리가 중단되고, 더이상 처리할 것이 없으면 끝날 수 있도록 프로그램을 작성하였다.

3.7 화상의 이동

화면의 이동과 확대, 축소는 화상 메모리의 데이터를 메인 메모리의 2000H 및 3000H번지에 옮겨 놓고 화상 메모리를 지운다음, 요구된 위치에 요구된 방법으로 다시 옮겨주는 것이다.

(1) 화면 이동(Moving)

이동시키려는 화면 영역을 설정하고 메모리의 2000H 및 3000H번지에 이 영역의 화상 데이터를 저장해 두고 화상 메모리를 지운다. 옮기고자 하는 화면 위치의 좌상점의 좌표를 입력시키고 메인 메모리에 저장해 두었던 데이터를 그 위치의 화상 메모리로 옮겨 넣으면 화면이 이동되어 나타난다.

(2) 확대(Zoom-magnification)

먼저 확대하고자 하는 영역을 설정해야 하는데 2배로 확대되기 때문에 전 화면의 1/4 영역(320×200)만 포함시킨다. 그 보다 클 때는 영역의 좌상점으로 부터 그 범위분만 유효하고 나머지 부분은 무시되어 버린다. 메인 메모리로 옮길 때 수평 방향으로 확대시킨다. 즉 한 번지에 들어있는 두 개의 화소를 분리하고 각각을 쌍으로 복제하여 각각 하나의 메모리 번지에 넣는다. 처리할 영역의 전 화상이 옮겨지면 화상 메모리를 지운다.

메인 메모리에 저장된 데이터를 화상 메모리로 옮길 때 수직 방향으로 확대시킨다. 즉 메인 메모리의 한 행을 화상 메모리의 두 행에 반복하여 넣으면 된다.

(3) 축소(Zoom-contraction)

주어진 화면을 1/2로 축소할 경우, 4개의 화소가 1개의 화소로 바뀌어 지므로 4개의 화소 중 대표 화소를 취하는 방법과 4개의 화소의 평균치를 취하는 방법이 있는데 여기서는 화상의 연속성을 살리기 위하여 평균치를 축소 화상의 1개 화소로 취하였다. 또 처리 영역을 취할 때 수평 방향으로 2의 배수, 수직 방향으로 2의 배수로 크기를 정하면 되지만 여기서는 화상의 구성을 고려하여 수평 방향 및 수직 방향의 영역 변수를 모두 4의 배수로 되게 하였다. 처리 순서는 다음과 같다.

2개의 행을 옮길 수 있는 버퍼를 메인 메모리에 설정하고 화상 메모리의 한 번지에 들어 있는 2개의 화소를 분할하여 평균을 취하고 메모리 버퍼에 넣는다. 이것을 2개의 행에 대하여 실시한 후, 버퍼의 상하행의 대응되는 화소끼리 평균치를 취하여 메인 메모리에 저장한다. 이렇게 하여 처리 영역의 모든 화상이 옮겨지면 처리된 화상 메모리를 지운다.

메인 메모리에 한 화소씩 저장된 데이터를 두 개씩 쌍으로 묶어서 화상 메모리에 전송하면 축소 화상이 나타나게 된다.

3.8 프린트(Print)

도트 프린터(Dot Printer)는 흑백 두 값 밖에

표시할 수 없기 때문에 몇 개의 도트를 조합시킴으로써 농담 화상을 표시할 수 있다. 본 프로그램에서는 16단계의 농도를 표현하는데 4×4=16 도트를 이용한다. 또 하나의 화소가 4bit로 이루어져 있으므로 16도트를 나타내기 위해서 농도치를 표시하는 값을 그대로 4회 반복하여 출력하는 방법을 쓸 수도 있으나 본 프로그램에서는 Fig. 3.7과 같은 도트 패턴을 각 농도치에 대응시켜 출력시키고 프린트하는 방법을 취하였다(PRNTV). 또 선으로 이루어진 화상을 이 방법으로 프린트하면 선이 끊겨 퍼지므로 16도트 중에서 중앙의 4개 도트만 프린트 되도록 프로그램을 짰다(PRNTVA). 프린터는 Epxon FX-80+를 이용하였으며 4배밀도(240 Dot/in)로 프린트한다.

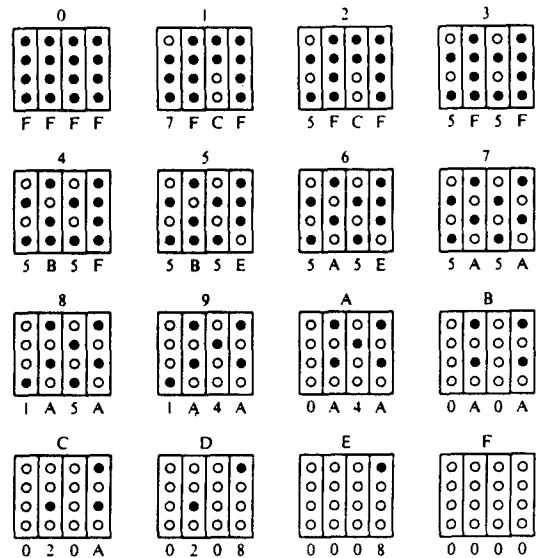


Fig. 3.7 Dot Pattern in Print

3.9 선 주사(Line Scanning)—썩음질(Thinning out)

이것은 얻어진 화상을 처리하고 해석하기 위한 일반 프로그램이 아니고 화상을 얻을 때 카메라 주사선을 제어하거나 얻어진 디지털 화상의 행 데이터를 적절히 처리하여 특수한 화상을 얻는 기법이다. 특히 여기서는 임의의 갯수의 행(Row) 또는 열(Column)의 데이터를 썩음질

(Thinning out)하여 원래 화상에 특수한 의미를 부여한 화상을 얻는 것이다. 하나의 행 데이터를 축소시킬 한다는 것은 화상을 잡을 때 해당 주사선을 축소시킬 한다는 것과 같으며 축소시킬 때 내고 그 자리는 비워 둘 수 없기 때문에 축소시킬 행의 바로 앞의 행 데이터를 그 자리에 채워 넣는 것이다(X-Scanning). 행 뿐만 아니고 열 데이터도 축소시킬 수 있다(Y-Scanning). 따라서 이 프로그램의 구성은 살리고자 하는 행 또는 열의 데이터를 메모리 버퍼에 옮겨 두고 축소시킬려는 행 또는 열에 축소시킬 수 만큼 반복하여 옮겨 넣는 것이다.

3.10 디스크 조작(Disk Processing: DISK FILE)

화상처리가 진행중이거나 완료되었을 때 화상 데이터를 파일로 만들어 디스크에 보관하며(SAVE), 저장된 파일 중에서 필요한 파일을 화상 메모리로 전송하여(LOAD) 화상처리에 이용한다. 그런데 한 화면의 화상 데이터는 640×400화소로서, 2 화소를 1바이트에 묶어서 나타내지만 한 화면 전체를 저장하는 데는 128KB의 메모리가 필요하게 되며, 하나의 디스켓에 2 화면 밖에 저장할 수 없으며 처리 시간도 많이 걸린다. 그래서 필요한 영역을 정하여 그 영역의 화상 데이터만을 저장할 필요가 있으며 또 때에 따라서는 저장된 파일 중에서 필요한 부분의 데이터만을 읽어내어 처리할 경우도 있다. 따라서 본 프로그램에서는 FCB(File Control Block)의 랜덤 액세스 방식(Random read and write)을 채택하였다.

SAVE : 파일명을 입력하여 파일을 열고(Open) 영역 데이터(좌표 값)를 파일에 저장하고 나서 화상 데이터 1행씩 데이터 버퍼(DTA)에 보내어 파일에 기록되게 하고 모두 저장되면 파일을 닫는다(Close)

LOAD : 파일명을 입력하여 디스크로부터 파일을 찾아서(Search) 영역 데이터를 읽어내어 화면의 해당 영역을 설정한 후 파일의 데이터를 순차로 읽어 화상메모리에 전송하면 화상이 재생된다.

4. 應 用 例

본 연구에서 얻어진 프로그램은 화상 처리를 필요로 하는 여러가지 실험과 연구에 이용될 것으로 기대된다. 즉 광탄성법에 의한 응력 해석, 주사 모아레법에 의한 변형률 해석, 홀로그래피 간섭법에서 얻은 데이터의 처리, 모아레 간섭법에 의한 미소 변형률 분포의 측정, 화상으로 얻어질 수 있는 대상물의 치수 및 밀도 측정, 현미경 사진의 특성 해석 등 여러 분야에 적용할 수 있다. 여기서는 광탄성 실험 결과의 처리에 적용한 예를 제시한다.

광탄성 실험에 의한 응력 해석은 재료 내부의 응력 분포를 직관적으로 볼 수 있는 장점이 있으나 실험에서 얻어진 화상으로 부터 응력 분포를 계산하는 과정이 아주 번거롭고 많은 수고를 필요로 하는 문제점이 있다. 즉 광탄성 실험에서 등색선(Isochromatic Fringe)과 등경선(Isoclinic Fringe)을 사진으로 촬영하여 기록하고, 손작업에 의하여 필요한 정보를 얻어내어 계산하게 되는데, 많은 시간이 걸리며 해석자에 따라서 오차를 수반하는 문제점이 있다.

최근 컴퓨터의 발달과 더불어 자동해석 시스템의 구성에 관한 연구가 이루어 지고 있다. 즉 Muller¹⁶⁾ 등은 자동 광탄성 실험장치와 미니 컴퓨터를 이용한 응력 해석 방법에 대하여 연구하였으며, Carter¹⁷⁾, Seguchi¹⁸⁾ 등은 컴퓨터를 이용한 광탄성 응력 해석에 관하여 연구하였다. 그러나 이들 연구 결과를 보완하여 보다 정밀하고 신속하게 데이터를 처리할 수 있는 시스템과 소프트웨어의 개발이 요망된다. 그 중의 일부본인 광탄성 실험 데이터의 처리에 본 연구에서 개발한 프로그램을 적용해 보았다.

4.1 등색선의 처리

등색선(Isochromatic Fringe) 처리에 본 프로그램을 적용하는 과정을 Fig. 4.1에 플로우 차트로 나타내었다. 즉 카메라로 화상을 잡고 아날로그 화상을 모니터로 확인한 후, 백색 등급(White Level)과 흑색 등급(Black Level)의 적당한 값을 택하여 최적의 디지털 화상을 얻는

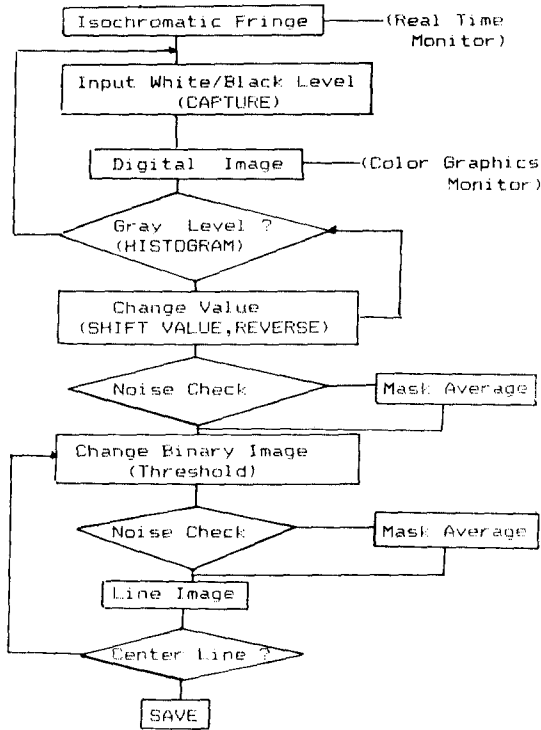


Fig. 4.1 Flow Chart of Processing for Isochromatic Fringe

다. 이때 농도분포(Histogram)를 조사하여 화상 처리에 적당한 농도분포가 되도록 하여야 한다. 얻어진 화상에서 농도변환(shift value 등) 기법으로 콘트라스트를 조절한 후, 평활화 처리(Mask Average)에 의하여 노이즈를 제거하고 적당한 한계치(Threshold Value)를 주어서 2치화 처리(Threshold)를 하면 흑백 화상이 얻어진다. 2치화된 화상에서 프린지의 주변에 노이즈가 있으면 세선화 과정에서 중심선 이외의 많은 불필요한 선 데이터 노이즈가 남게 되므로, 평활화 2치화에 의하여 제거하고 세선화 처리를 행한다. 이렇게 하여 얻어진 선으로 된 화상이 등색선의 대표치이며 그 좌표치와 등색선 차수가 응력 계산에 이용된다.

4.2 등경선의 처리

Fig. 4.2와 같이 등경선이 없는 2치화 화상과 등경선을 포함한 2치화 화상을 얻어서 각각 메

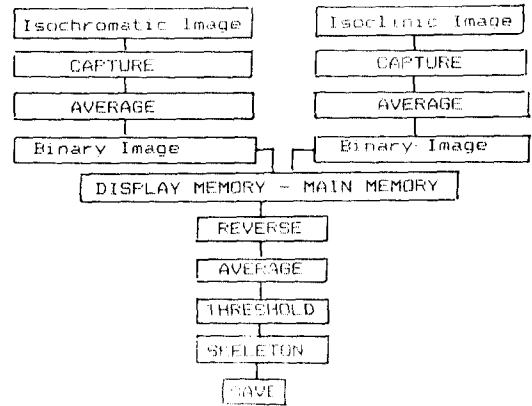


Fig. 4.2 Flow Chart of Processing for Isoclinic Fringe

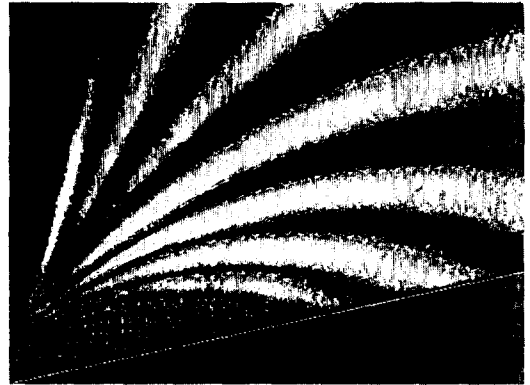
인 메모리(Main Memory)와 화상 메모리(Display Memory)에 저장한 후, 그 차를 연산하여 등경선을 구한다. 이 화상을 평활화, 세선화 과정을 거쳐서 등경선의 선 데이터를 얻는다.

4.3 처리 결과

Fig. 4.3은 광탄성 실험 결과의 사진이다. 예 폭시 수지로 만든 바이트로서 납을 절삭하는 2차원 절삭 과정의 광탄성 실험 결과이며 (a)는 등색선(Isochromatic Fringe), (b)는 등경선(Isoclinic Fringe) 사진이다. 패쇄회로 카메라로 절삭 중의 화상을 잡고 VTR에 녹화시킨후, 재생시키면서 상의 변화를 관찰하고 필요한 부분을 디지털 화상으로 잡아서 해석하는 것이다. Fig. 4.3(a)로부터 흑백 등급(Black/White Level) 2와 10으로 디지털화 하여 본 프로그램의 화상 프린트 모드로 프린트한 결과가 Fig. 4.4(a)이다. 이 화상에는 많은 노이즈가 포함되어 있으므로 마스크 평활화에 의하여 노이즈를 제거하고, 한계치(Threshold Value) 3으로 2치화 처리를 하여 반전(Reverse)시킨 화상이 Fig. 4.4(b)이다. 이것을 세선화하여 등색선 무늬의 중심선을 구한 것이 Fig. 4.5(a)이다. 같은 기법으로 등경선에 대하여 처리한 결과가 Fig. 4.5(b)이다.



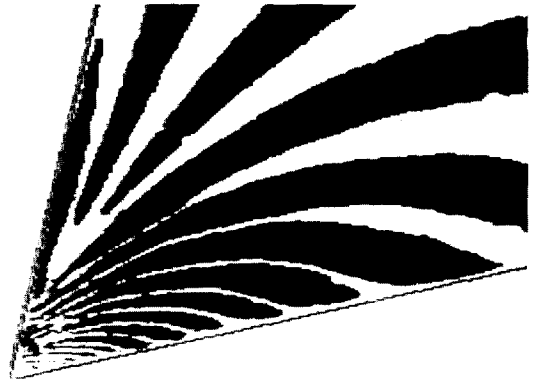
(a) Isochromatic Fringe



(a) Original Image



(b) Isoclinic Fringe



(b) Reversed Binary Image

Fig. 4.3 Photoelastic Fringes

Fig. 4.4 Digital Image of Isochromatic Fringe

이상의 처리 과정에서 원 화상의 부분적인 명암 차이와 노이즈의 밀도 차이 때문에 전체를 동시에 처리하기 곤란한 경우가 있으므로 몇개의 영역으로 나누어서 각 부분을 따로 처리하여 합성하면 훨씬 좋은 결과를 얻을 수 있다. 즉 Fig. 4.4(a)에서 인선 부분과 인선으로 부터 떨어진 부분을 별도로 처리하여 합성하는 것이다.

이상과 같이 광탄성 실험 결과에 본 연구에서 개발한 화상처리 기법을 적용하여 등색선과 등

경선의 중심선을 구한 결과 비교적 양호한 것을 얻었으며 이 결과를 이용하면 컴퓨터에 의한 응력해석이 가능할 것이다. 응력 해석 프로그램도 개발중에 있으므로 전 과정을 전산화하는 결과를 얻을 것으로 기대된다. 다만 광탄성 실험에서 인선 부분에는 좁은 영역에 많은 무늬가 집중되므로 레이저와 같은 간섭이 적은 광원을 쓰고 감도가 높은 카메라를 사용하는 것이 바람직하다.

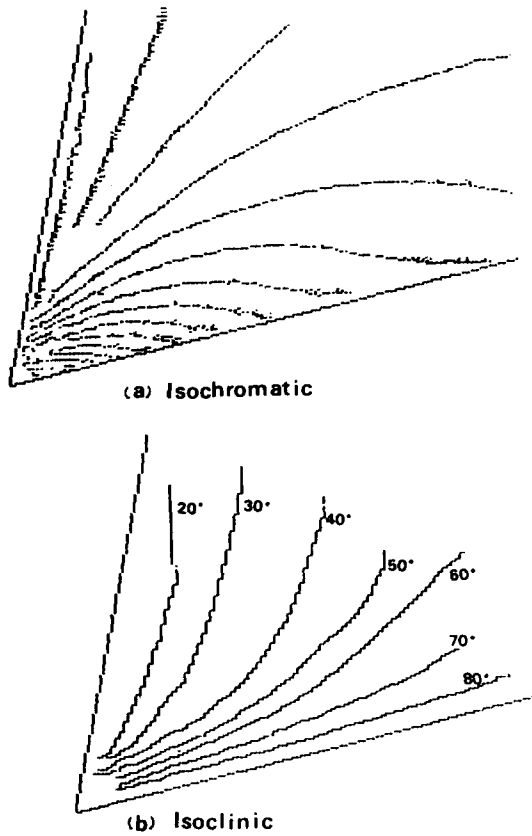


Fig. 4.5 Center Line of Photoelastic Fringes

5. 結 論

마이크로 컴퓨터 IBM PC-XT와 그래픽 보드를 기본으로 하여 화상 처리 시스템을 구성하였으며, 어셈블리 언어로 서브 프로그램을 작성하고 대화형으로 구성하여 사용하기 편리하고 처리 속도가 빠른 화상 처리 기법을 개발하였으며, 광탄성 실험에서 얻은 화상 데이터를 처리하는데 적용하여 효용성을 확인하였다.

본 연구에서 개발한 화상처리 프로그램은 화상의 취득, 영역의 설정, 농도 분포의 확인, 농도변환, 화상의 합성, 세선화, 마스크 처리에 의한 노이즈 제거, 화상의 이동 및 확대, 디스크 액세스, 프린트 등 화상처리에 필요한 대부분의 서브루틴을 포함하고 있으므로 응력 해석, 변형률 해석, 조직의 분포 해석, 치수 측정, 현

미경 사진의 특성해석 등 화상으로 얻어지는 실험 결과를 처리하는데 적용될 것으로 기대된다. 특히 주사 모아데법에 의한 변형률 해석에 적용한 결과는 제2보에서 제시하고자 한다.

參 考 文 獻

1. 森本 吉春, “パーソナルコンピュータを用いた畫像處理システムの構築”, PIXEL, NO. 35(1985)
2. 森本 吉春, “パソコンによる畫像處理”, PIXEL, NO. 15, PP. 137~143(1983)
3. 森本 吉春, 畫像處理, 東京, 增風館(1984)
4. 森本 吉春, 白石泰, “パーソナルコンピュータによるしま畫像解析”, 村料, V. 33 NO. 367, PP. 495~500(1984)
5. 森本 吉春, 加納 公生, “パソコンによる畫像處理とそのプログラミング”, PIXEL, NO. 36 (1985. 9)···NO. 43, PP. 111~118(1986. 4)
6. E. L. Hall, Computer Image Processing and Recognition, Academic Press(1979)
7. William K. Pratt, Digital Image Processing, John Wiley and Sons, Inc.(1978)
8. A. Sosenfeld, A. C. Kak, Digital Picture Processing, Academic Press(1982)
9. Chorus Data System, Inc., PC-EYE User's Guide(1984)
10. Chorus Data System, Inc., PC-EYE Technical Reference Manual(1984)
11. IBM Guide to Operations and BASIC, IBM Co. (1983)
12. Microsoft Corp., Disk Operating System, IBM Co.(1983)
13. R. Tocci, L. Laskowski, Microprocessors and Microcomputer, Prentice-Hall(1982)
14. Peter Abel, Assembler for the IBM PC and PC-XT, Reston Publishing Company, Inc. (1984)
15. Robert Jourdain, Programmer's Problem Solver for the IBM PC XT & AT, Brady Communications Company, Inc.(1986)
16. R. K. Muller and L. R. Saackel, “Complete Automatic Analysis of Photoelastic Fringe”, Experimental Mechanics, 1979. 7, pp. 245~

56 韓國船用機關學會誌, 第12卷 第3號, 1988

251

17. Alan L. Carter, "Obtaining Stress Maps from Photoelastic Data Using a Digital Computer", *Experimental Mechanics*, 1968. 3, pp.122~

127

18. Y. Seguchi, Y. Tomita and M. Watabe, "Computer Aided Frings Pattern Analyzer—A Case of Photoelastic Fringe", *Experimental Mechanics*, 1979. 10, pp. 362~370.