산업용 비젼시스템을 위한 하드웨어 체인코더의 설계

°이 병 일*, 신 유 식*, 임 준 홍**, 변 증 남*

* 한국과학기술원, ** 한국항공대학

A Hardware Implementation of Chain-coding Algorithm for Industrial Vision Systems

°B.I.Rhee*, Y.S.Shin*, J.Lim**, Z.Bien*

* KAIST EE, ** Hankuk Aviation College

ABSTRACT

In an industrial vision system, a coding technique for binary image is essential to extract useful informations. To reduce the processing time, a hardware implementation of the chain coding algorithm is attemped. For that purpose, the chain coding algorithm is modified so that it is more suitable for a hardware implementation. A hardwired chain coder is also developed and tested with developed vision system. The result shows that the processing time is greatly reduced and that the developed vision system is maybe feasible for real-time applications.

1. 서 론

산업현장의 자동화에는 각종 센서가 사용되고 있지만 그중 특히 사람의 시각기능을 대신할 수 있는 Vision System은 그 사용용도의 다양성으로 인해 더욱 각광을 받고있다[1,2]. 특히 산업용 Vision System은 생산라인상의 각종 물체를 식별하고 제품의 양,불량 여부를 판정하며 위치를 검출하게 하는등 다양한 면에 이용되어 질 수 있다[2]. 그러나, 산업현장에서 가장 문제가 되는것은 생산성 이며[3], 이 생산성을 결정짓는 요소에는 여러가지가 있지만 그 중 가장 큰 비중을 차지하는것이 그 소요시간이다. 따라서 생산성을 향상시키려면 그 처리시간을 줄이는것이 가장 효과적이다.

산업용 비젼 시스템에 보다 빠른 처리시간을 가지게 하여 생산성 향상에 기여 하고자 하는 노력은 여러가지로 시도되어 왔다.[4,5,6,7] 산업 현장이라는 제한 조건은 인공 광원이나 기타 환경을 적절히 조절하여 명암의 구별을 뚜렷이 할 수 있다는 가정을 세울 수 있도록 해준다. 따라서 Gray Level Image로 부터 Binary Image (2진 영상)를 얻을때 물체의 모양에 대한 정보가 손상되지 않는다고 가정하고, 이렇게 하여 만들어진 Binary Image에서 다시 물체에 대한 정보만을 뽑아내기 위한 Coding 방법으로 SRI Module[4]에서 사용한 Run-length Coding 과 Freeman이 제안한 Chain Coding[7] 이 주로 사용되고 있다.

본 논문에서는 Chain Coding Algorithm을 Hardware화에 알맞는 형태로 구성하여 수정 제안하였다. 이를 위해 경계점, 주변값, 추적규칙등을 새로이 정의하거나 정리하고, 이를 구현하여 Hardware Chain coder를 제작하였다. 이 Hardware Chain Coder는 본 논문에서 제안된 경계점및 추적규칙에 따른 chain code등을 Look up table형태로 구성하고 전체 동작을 Micro Program에 의해 동작되는 Controller로 제어하였다. 이렇게 제작된 Hardware는 개발된 Vision System에 적용하여 실험한 결과, 종전에 software에 의해 chain coding을 행하는 것에 비해 약 1/20 정도의 시간이 단축되어 실시간 응용에 이용가능성을 보였다[8].

2. Hardware 구성이 용이한 chain coding algorithm

적절한 역치 선정을 통해 얻어진 이진 영상으로 부터 유용한 정보들을 보다 간편히 얻어 내기 위하여는 또 다른 data compression 방법들이 사용 된다. 이 장에서는 이러한 data compression 방법중 chain coding 에 대하여 설명하고 이를 hardware로 구현하기 위한, 보다 간편한 algorithm을 제안하고자 한다.

본 algorithm은 이진 영상에 대하여 이진영상으로 표현되는 물체의 두께는 적어도 2 pixel 이상일것과, 이진 영상으로 표현된 물체는 전체가 영상 내부에 있을것을 전제로 한다. 첫째 제약조건은 산업용 Vision System의 경우 물체를 충분히 크고 뚜뚜하게 볼 수 있기때문에 그리 큰 문제점이 되지는 않는다. 둘째 제약조건은 물체의 인식을 위한 특징량을 구하기위해서는 반드시 있어야할 가정이다.

2.1 경계점및 주변값 (neighbor-value)

적절한 임계치 선택을 통해 얻어진 이진 영상을 chain coding 을 이용해 부호화 하고자 할때 먼저 구해야 하는것은 영상의 경계이다.

이웃 관계를 가지는 두 pixel 이 경계를 이룰 조건을 다음과 같이 정의하자.

정의 1) p(i,j), p(k,l)을 이웃 관계에 있는 두 pixel 값 (0 또는 1) 이라 할때,

$$| p(i,j) - p(k,l) | = 1$$

을 만족하면 p(i , j), p(k , l) 사이를 영상의 경계라 한다.

정의 1 로부터, 중심 pixel p(i,j) 에 대하여 8 - neighborhood 이웃관계를 가지는 pixel 들의 집합을

$$Pb(i,j) = \{ p(k,l) \mid i-1 \le k \le i+1,\ j-1 \le l \le j+1,\ k \ne i,\ l \ne j \}$$

라고 쓸 수 있고, 만약 p(k , l) ∈ Pb(i , j) 인 모든 pixel 에 대해 | p(i , j) - p(k , l) | = 0 이면, p(i,j) 는 경계점이 아니라고 말할 수 있다. 본 논문에서는 8 -neighborhood를 사용하였다.

이진 영상 내에서 물체의 경계가 있는 위치를 알기 위해서는 이진 영상 전부를 탐색할 수 밖에 없다. 이진 영상을 탐색하면서 탐색점이 위의 경계점의 정의에 부합되는가를 확인하면, 경계점의 위치를 구할 수 있다. <정의 1>과 더불어 주변값(Neighbor-value) 을 정의하자.

정 의 2) 경계점 p(i,j) 에 대한 주변값 M(i,j) 는 8 bit 의 이진수 이고 각bit를 b7,b6 ...b0 라하면, 각 bit는 다음과 같이 정의한다.

(b7 , b6 , b5 , b4 , b3 , b2 , b1 , b0)
= (p(i+1 , j+1) , p(i , j+1) ,
p(i-1 , j+1) , p(i-1 , j) ,
p(i-1 , j-1) , p(i , j-1) ,
p(i+1 , j-1) , p(i+1 , j))
단, p(i , j) = 1

2.2 시작점(start point)

이진 영상의 탐색 순서를 영상 좌표 (그림 2.1)의 (0,0)로 부터 x 축 방향으로 (255,255) 까지 탐색한다고 하면, p(i - 1,j) - p(i,j) = -1인 경계점중 가장 먼저 만나게 되는 경계점의 주변값은 그림2.2(a)과 같은 값을 가지게 되며, chain coding 을 위한 추적은 이 점으로 부터 시작하면 되므로 앞으로 이 논문에서는 시작점(start point)이라 부르기로 하겠다. 위의 값들은 영상 탐색의 순서 때문에 주변값의 bit 1 , 2 , 3 , 4 값이 모두 '0'인 경우의 수들이다. 그러나 '1' 에서 '0' 으로 변하는 경계점에서는 뒤에서 언급하는 chain coding 의 방향 추적 규칙 때문에 발생하는 무한 추적의 경우를 만들지않게 하기 위하여 다음과 같은 조건을 갖는 경계점을 시작점으로 정의한다. 1) 3 x 3 window 내의 상측및 좌측 pixel 값은 모두 '1' 일것. (bit 1 , 2 , 3 , 4 , 5) 2) p(i + 1 , j) = 1 인 경우는 p(i , j + 1) = 0 이어야 할것. 이상의 조건을 만족하는 pixel 의 주변값을 그림 2.2(b)에 표시 하였다.

2.3 Chain coding 을 위한 추적 규칙

시작점의 위치를 알고나면 그때 부터 chain coding 을 시작할 수 있는데, 이를 위해서는 다음 경계점을 찾아가는 추적 규칙이 필요하다. 먼저, 경계점들 로서 이루어진 폐곡선을, 시작점을 기준으로 하여 반 시계 방향(Counter Clock Wise) 으로 추적해 가는 방법과 시계 방향(Clock Wise) 으로 추적해 나아가는 방법이 있는데, 어느 방법이나 결과는 같기 때문에 이 논문에서는 반 시계 방향으로 추적하는 방법을 선택한다. 또, 각각의 추적법을 CCW추적, CW추적 이라 부르기로 하겠다. 또한, 다음의 경계점을 찾기 위해 경계점을 중심으로 한 단계 전의 경계점으로부터 좌측으로 돌면서(Rotate Left) 경계점을 찾는 방법과, 우측으로 돌면서(Rotate Right) 경계점을 찾는 방법을 들 수 있는데, 각각을 RL탐색, RR탐색 이라 부르기로 하겠다. 그림 2.3 에 CCW추적, CW추적 그리고 RL탐색및 RR탐색 방법을 보였다. 이상에서 얻거한 방법을 조합해 보면 총 4 가지의 mode 가 만들어 지는데 각각은 다음과 같다.

- CCW추적 - RL탐색
- CW추적 - RL탐색
- CCW추적 - RR탐색
- CW추적 - RR탐색

위의 4 가지 mode 중 본 논문에서는 CCW추적 방법만을 사용하므로 CCW추적 - RL탐색 mode 와 CCW추적 - RR탐색 mode 의 두가지로 제한 되는데, 이진 영상에서 물체가 '1' 로 표현되는 경우, 물체의 외부 경계는 CCW추적 - RR탐색 mode 를 사용하고, 내부경계는 CCW추적 - RL탐색 mode 를 사용한다.

한편, 내부경계의 시작점에서 CCW추적 방법이 적용되기 위해서는 최초의 탐색 방법이 RR탐색 이어야만 하며, 그 이후에는 정상적으로 RL탐색으로 전환되어져야 한다. 또, CCW추적 - RL탐색의 경우에 있어서, 시작점을 정하는 방법에 따라 그림 2.4 와 같이 시작점으로 돌아오지 않게되는 무한 추적의 경우가 발생한다.이 경우는 시작점을 잘 선정해 주어야하는데, 이러한 무한 추적을 없애기 위한 시작점의 조건은 다음과 같다. 1) p(i - 1 , j + 1) = 1 2) 만약 p(i + 1, j) = 0 이면, p(i, j + 1) 는 반드시 0 이어야 한다.
추적의 종단은 시작점으로 되돌아 올 때와 주변값이 0 이 되는 경우에 이루어 진다.

이상의 추적 규칙을 종합, 정리해보면, 물체가 '1' 로 표현될 경우 1) 물체의 외부 경계는 CCW추적 - RR탐색 mode로 추적한다. 2) 물체의 내부 경계는 시작점에서는 CCW추적 - RR 탐색 mode를 적용하고, CCW추적 - RL탐색 mode 로 전환하여 추적한다. 3) 추적의 종단은 추적점의 좌표값이 시작점의 좌표값과 일치할 경우와, 추적점의 주변값이 0 이 될 경우에 발생한다.

3. Chain Coding 의 Hardware 화 기법

3 . 1 Chain Coder의 구성

그림 3.1에 chain coder전체의 구성도를 보였다. Chain coder는 크게 neighbor-value coder, neighbor-vlaue image memory (NIM), chain table, chain list memory, neighbor-value image address generator및 micro programmed logic controller와 address decoder등 으로 나누어 볼 수 있다.

1 raster scan시간 동안 LUT에 의해 이진화된 영상이 vision bus를 통해 neighbor-value coder로 들어 오면 neighbor-value로 변환된 출력이 NIM에 저장된다. raster scan이 끝난 후 command에 의해 NIM을 탐색하여 시작점을 찾고, 시작점이 발견되면 NIM의 address는 chain table에 의해 결정된 chain code에 따라 변화되며 tracking을 계속하게되고, 얻어지는 chain code는 chain list memory에 저장된다. NIM의 탐색부터 chain code의 저장에 이르는 일련의 과정은 micro programed logic controller로 제어된다.

3.2 Chain coder의 state

chain coder의 상태는 총 4 가지로 구분 지을 수 있고, 각각은 다음과 같다.

a) Idle state (Idle)

b) Memory Encoding state (ME)

b) Memory Searching state (MS)

d) Contour Tracking state (CT)

이들 각 state는 micro programmed logic controller에 의하여 각각 제어되며 사용자의 명령에 의해 다른 state로 천이될 수 있다. 이들의 천이 관계와 사용자의 명령에 대한 상태도를 그림 3.2 에 보였다.

a) Idle State

이 state는 C P U 로 부터 명령을 받거나, C P U 가 chain coder내에 있는 각종 register및 memory를 access할 수 있도록 하는 대기 상태이다. 이 상태에서는 chain coder 내에 있는 모든 memory와 register들의 address및 data는 system의 address와 data bus에 접속된다. 따라서 C P U는 필요에 따라 이들을 accress할 수 있다.

C P U에서 accress하는 내용은 ,

1) neighbor-value coding table의 write

2) NIM 상에서 tracking 된 부분의 내용을 clear

3) chain code의 길이를 read

4) chain code의 시작점 좌표값을 read

5) chain coder status register read / write

6) neighbor-value coding 시작 명령

7) memory searching 명령

8) chain table의 write

9) chain code list의 read

b) Memory Encoding State

Idle state에서 C P U로 부터 neighbor-value coding 시작명령(MESC)이 내려지면 3개의 vision bus로 부터 입력되는 이진영상 data가 3 x 3 window및 neighbor-value coding table을 거쳐 NIM에 저장되는 neighbor-value coding state로 된다. NIM의 address는 raster scan방식으로 증가되며, 좌표값이 (255 , 255)가 되면 MEE signal을 발생 시켜 idle state로 되돌아 간다.

c) Memory Searching State

idle state에서 C P U로 부터 memory search 시작명령 (MSSC)이 내려지면 C P U에 의해 미리 지정된 window내를 raster scan하며 시작점을 찾게 된다. 미리 마련된 table과 같은 값을 가진 주변값 (neighbor-value)이 인지되면 그 점을 시작점으로 간주하고, START signal을 발생시켜 contour tracking state로 상태천이를 행한다. 시작점이 검출되지 않고 window의 끝에 도달하면 NIM address cmoparator에 의해 MSE signal이 발생되면서 idle state로 상태 천이가 일어난다. 한편, window는 불필요한 부분의 scan을 최소화 하기위해 사용자가 임의의 크기를 가진 사각형으로 지정할 수 있다. 또 시작점이 검출되고 contour tracking이 끝난뒤 다시 C P U 로 부터 시작명령이 내려지면 contour tracking이 끝난 그 지점 부터 다시 raster scan을 계속 하게 된다.

d) Contour Tracking State

memory searching state로 부터 START signal에 의해 contour tracking state로 상태 천이가 이루어 지면, NIM으로 부터 출력되는 주변값과 바로 전 step의 chain code, 그리고 시작점 발견시에 결정되는 mode select bit로 구성되는 chain table에서 다음의 chain code를 얻고, 이 chain code를 토대로 NIM address generator를 control하게된다. 한편, 얻어지는 chain code들은 순서적으로 chain list memory내에 저장되며, 이때 chain의 길이도 같이 증가된다. 또, contour tracking state로 상태 천이가 이루어질때 시작점의 address를 기억시켜, tracking이 되는 모든 pixel의 address와 비교하여 그값이 같아지게 되거나, 주변값이 '0'이 되게끔 address가 지정되면 END signal이 발생하며 idle state로 상태 천이를 일으켜 C P U 로 하여금 tracking된 부분의 주변값을 '0'으로 만들어 중복되게 tracking하는 요소를 없앤다.

기본적인 chain coding 순서는 그림 3.3 에 나타 내었다.

4. 실험결과및 검토

Chain coding time은 raster scan을 포함하는 시간으로써 software로 처리할 경우 많은 시간을 요하는 요소이다. 표 4.1과 그림 4.1에 chain code의 갯수와 Hardware chain coder를 이용한 chain coding time 측정치를 보였다. Software로 처리할때 걸리는 시간을 1 pixel 당 10㎲sec 라고할때, raster scan시간과 비교해보면 약 20배 이상의 speed향상을 이룰 수 있었다. 그림 4.1에서 보듯이 구조상 raster scaning time이 항상 존재하게되므로 25.8ms의 기본 chain coding time이 소요되며 code의 갯수가 증가함에 따라 그 소요시간은 선형적으로 증가한다. 따라서 code의 갯수가 많아질수록 속도 증가의 효과는 커진다.

현재 program의 개발 tool은 C - language를 이용했는데 program의 overhead를 줄이기위해 assembly language로 바꾸면 30% 이상의 processing time감소가 예상된다.

5. 결 론

산업 현장에서 유용하게 사용될 수 있는 Vision

System은 저가격 이면서 사용자의 다양한 요구에 대처할 수 있는 유연성과 신뢰성을 지닌 System이다. 본 논문에서는 이와같은 욕구를 충족시킬 수 있는 Vision System의 설계를 주 과제로하여, 이러한 일을 수행하는데에 있어서 기존의 System이 안고있는 processing time의 단축문제를 해결하기위해 이진 영상처리를 할때, Hardware화에 보다 적합한 Chain Coding 알고리즘을 제안하고 chain coder를 설계,제작하여 Vision System에 이용함으로써 알고리즘의 유용성을 증명하였고, 실험을 통해 chain coding시간이 software에 의한것 보다 약 20배 정도 빨라짐을 볼 수 있었다.

앞으로의 연구 과제는 제안된 chain coder에서의 제약조건을 제거하고 물체의 국부적인 특징을 이용해 인식을 할 수 있게하는 chain coder의 보완및 더많은 특징량을 찾아내어 더욱 많은 물체를 인식할 수 있도록 하는 새로운 알고리즘의 개발이 필요하다.

참고 문헌

1) Eugene Charniak, Drew McDermott "Artificial Intelligence" Addision Wesley 1985
2) 월간 자동화 기술 1985년 10월호
3) 월간 자동화 기술 1985년 3월호
4) Gerald J. Agin "Computer Vision Systems for Industrial Inspection and assembly " IEEE Computer May 1980
5) Alan Pugh " Robot Vision " IFS Ltd. U.K.
6) Carl Helmers "Robotics Age in the beginning" HAYDEN BOOK COMPANY Inc. p81 ~ 149.
7) Herbert Freeman "Computer Processing of Line-Drawing Images" Computing Surveys Vol. 6 No.1 March 1974.
8) 이 병 일 "물체인식을 위한 산업용 비전 시스템의 설계 ", 한국과학기술원 1987
9) 이 정 환 "Run-length code를 이용한 새로운 Chain coding 알고리즘에 관한 연구", 한국과학기술원 1986

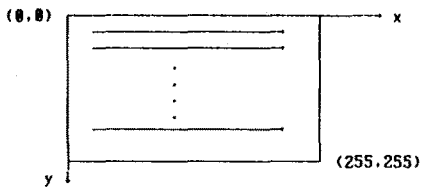
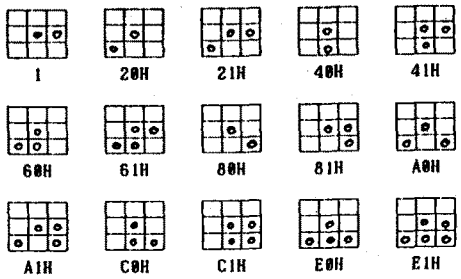
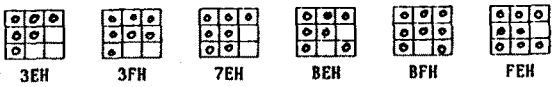


그림 2.1 영상 탐색 순서



(a) '0' → '1' 경계에서의 시작점



(b) '1' → '0' 경계에서의 시작점

그림 2.2 시작점의 종류

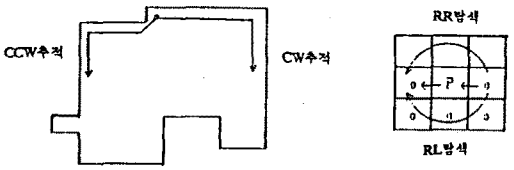


그림 2.3 CCW추적, CW추적, RL탐색, RR탐색

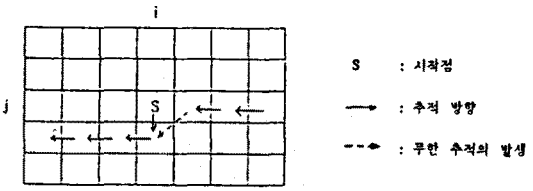


그림 2.4 무한 추적의 발생

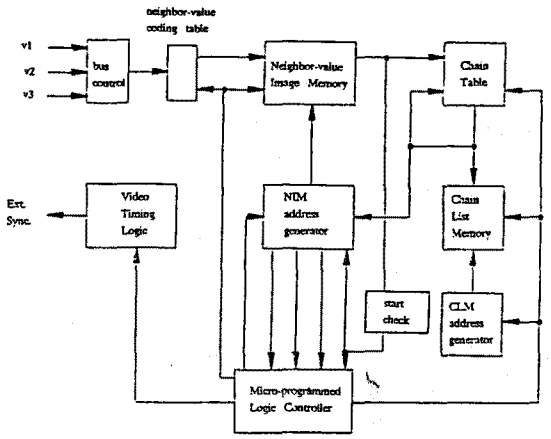


그림 3.1 Chain Coder의 전체 구성

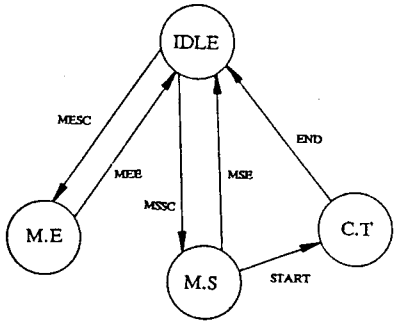


그림 3.2 Chain Coder의 상태 천이도

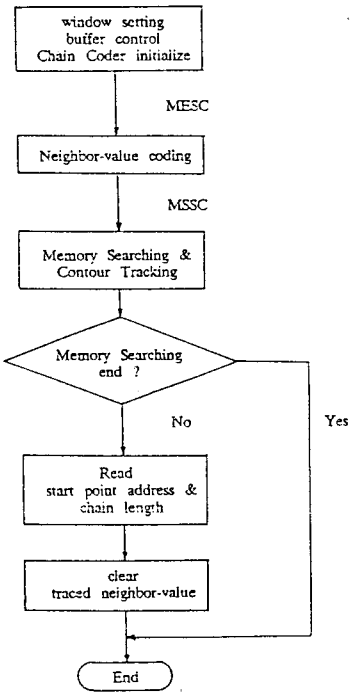


그림 3.3 Chain Coder를 이용한 기본적인 Chain Coding 순서

chain coding number[개]	time [mSec]
90	29.2
200	32.9
316	38.3
381	41.4
506	47.1
602	51.4
733	56.6
1738	120.5

표 4.1 chain code 갯수와 chain coding time

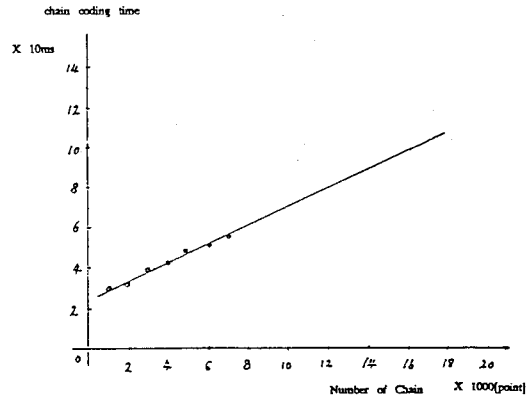


그림 4.1 chain code number및 chain coding time

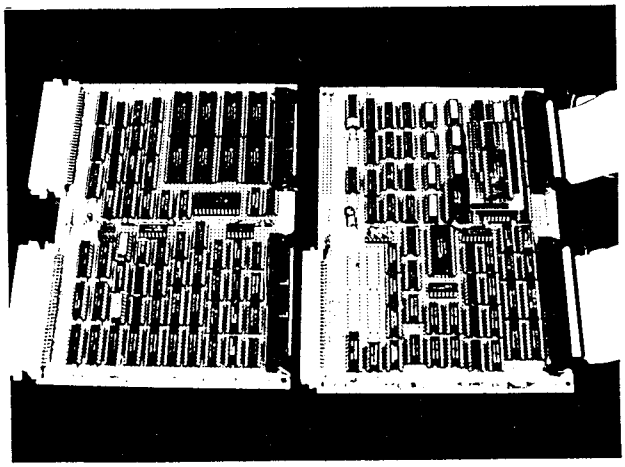


사진 1. Chain Coder의 외관

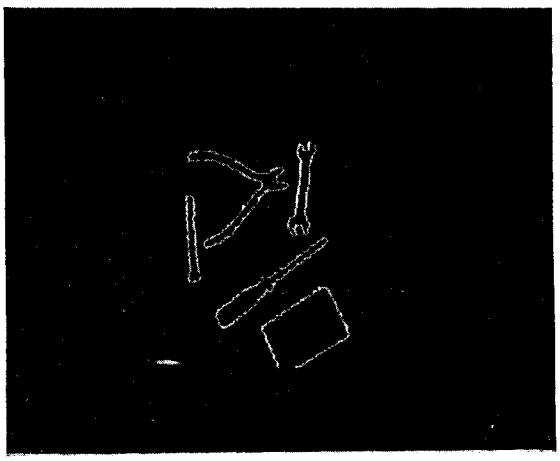


사진 2. Chain coding결과의 Display