Attributed Relational Graph 를 이용한 영상 패턴의 인식에 관한 연구

이 광 기, 전 중 남, 이 창 한, 이 한 욱, 박 규 태

연세대학교 전자공학과

A Study on Image Pattern Recognition using Attributed Relational Graph

Kwang Kee Lee, Joong Nam Jeon, Chang Han Lee,
Han Wook Lie and Kyu Tae Park

Yonsei University

ABSTRACT

Algorithms that represent given pattern in the form of an ARG (Attributed relational graph) using not only structural relations but also symbolic or numerical attributes, and then recognize that pattern by graph matching process are presented in this paper. Based on definitions of pattern deformational models, algorithms that can find GPECI(Graph preserved error correcting isomorphism), SGECI(subgraph ECI) and DSECI(Double sub graph ECI) are proposed and comparisons among these algorithms are described.

To be useful in performig practical tasks, efficient schemes for extraction of ARG representation from raw image are needed. In this study, given patterns are restricted within objects having distinct skeleton, and then the information which is necessary for recognition and analysis is successfully extracted.

1. 서 론

패턴의 구조적 관계를 표현할 수 있는 일반적인 수학적 모델에 대한 연구의 일환으로, 노드와 브랜취에 심볼(symbol) 및 속성값(attributed value)을 부여할 수 있는 ARG (attributed relational graph) 에 대한 다양한 연구가 진행되고 있다. [1,2,3] ARG는 노드와, 노드들 간의 관계를 표현하는 브랜취들로 이루어진 관계적 구조이며 각각의 노드와 브랜취는 그에 할당된 속성을 가질 수 있다. 주어진 패턴을 인식하는 방법은 입력패턴의 구조 및 속성을 ARG로 묘사하고 이를 원형패턴에 대한 ARG와 매칭하는 것이다.

본 연구에서는 Tsai와 Fu의 ARG와 ARG에 대한 GPECI(graph-preserved error-correcting isomorphism)[1] 알고리듬을 고찰하고 이를 SGECI (subgraph ECI), DSECI (double subgraph ECI) 으로 확장하여 그 응용 가능성을 넓히고자 한다. 각 알고리듬은, 입력패턴을 원형패턴의 훼손된 모델로 정의한 후 각각의 훼손에 대한 정의를 기초로 최소의 distance 를 갖는 매핑 함수를 찾는 것을 목적으로 한다. 이때 distance 는 입력패턴에 대한 ARG 로 부터 원형패턴에 대한 ARG 로 변형 (transformation) 시키는데 필요한 비용으로 정의한다. [2]

또한 2차원적 구조를 갖는 물체로 부터 패턴에 대한 골격(skeleton) 을 얻은 후 이 골격을 기초로 ARG 로 묘사를 추출한다. 같은 방법으로 몇개의 물체들로 이루어진 전체 영상의 해석에 필요한 정보들을 추출한다.

본 연구에서는 그래프 변환기와 매칭부를 Prolog로 구현하였는데 이는 ARG의 묘사와 매칭 조건을 표현하는데 효율적일 뿐 만 아니라 패턴 및 영상 정보의 관리에 있어서도 많은 장점이 있음을 확인할 수 있었다. 이에따라, ARG와 그 매칭 조건은 각각 prolog에서의 사실(fact)과 룰(rule)로 표현된다.

2. ARG (Attributed relational graph) 의 정의

2.1 ARG의 정의

원시형태와 관계로부터 다음과 같이 ARG에 대한 정의를 내린다. 각 노드들은 원시형태 혹은 부분패턴 (subpattern) 에 해당하며 두 노드 사이의 브랜취는 이들 원시형태나 부분패턴 사이의 관계에 해당한다.

[정의 2.1] 노드들의 유한집합 N={i|i=1,2,...,n} 과 브랜취들의 유한집합 B={j|j=1,2,...,m} 로 이루어진 ARG를 가정하자. 이때 노드 i와 브랜취 j는 아래와 같은 predicate들로 표현된다.

1) nod(i) : 노드 i가 존재한다.
2) bran(j,i1,i2) : 서로 다른 노드 i1에서 i2로 향하는 브랜취 j가 존재한다.
3) nod_inter(i,s) : 노드 i에 할당된 구문론적 심볼은 s이다.
4) bran_inter(j,u) : 브랜취 j에 할당된 구문론적 심볼은 u이다.
5) nod_attr(i,x) : 노드 i의 의미벡터는 x 이다.
6) bran_attr(j,y) : 브랜취 j의 의미벡터는 y 이다.

2.2 패턴 훼손 모델(pattern deformational model)

원형패턴에 대한 입력패턴의 훼손의 성격과 그 허용범위의 제한에 따라 다음과 같은 패턴훼손모델을 정의할 수 있으며 ARG 상의 노드 및 브랜취에 대한 삭제(deletion), 첨가(insertion), 대체 (substitution) 에 대한 해석에 따라 각각의 훼손 모델에 대처할 수 있는 매칭 알고리듬을 가정할 수 있다. (아래의 정의에서 Gw와 Gw'는 w 및 w'에서 속성을 제거한 그래프 (unlabeled-graph)를 의미한다.)

1) Graph-preserved deformation
훼손이 Gw의 구조(structure)에는 영향을 주지 않고 단지 원시형태와 관계에 대한 정보만을 훼손시킨 형태. Gw~Gw'.

2) Partial graph-preserved deformation
입력패턴이 그의 원형패턴에 대한 원시형태와 관계의 일부를 손실한 경우. 즉 Gw'가 Gw의 subgraph인 경우.

3) structural deformation
위의 1),2)의 범위를 넘는 훼손. 즉 Gw의 subgraph와 Gw'의 subgraph가 동일한 경우.

2.3 ECI(error-correcting isomorphism) 의 정의

지금까지의 정의를 기초로 다음과 같은 조건을 만족하는 함수 h : w' -> w 를 w'로 부터 w로의 ECI (error-correcting isomorphism) 이라 한다.

1) N'의 모든 원소 OBS에 대해 다음과 같은 rule을 만족하는 N의 원소 PURE가 존재한다.

```
map_nod(OBS,PURE) :- nod_inter_obs(OBS,LAB_OBS),
                     nod_inter_pure(PURE,LAB_PURE),
                     deformed_nod(LAB_PURE,DEF_LIST),
                     member(LAB_OBS,DEF_LIST).
```

2) B'의 모든 원소 BR_OBS에 대하여 다음과 같은 rule을 만족하는 B의 원소 BR_PURE가 존재한다.

```
exist_bran(OBS1,PURE1,OBS2,PURE2) :-
```

```
bran_obs(BR_OBS,OBS1,OBS2),!,
bran_pure(BR_PURE,PURE1,PURE2),
nod_inter_obs(OBS1,LAB_OBS1),
nod_inter_obs(OBS2,LAB_OBS2),
bran_inter_obs(BR_OBS,LAB_BR_OBS),
bran_inter_pure(BR_PURE,LAB_BR_PURE),
deformed_bran(LAB_OBS1,LAB_OBS2,LAB_BR_PURE,
              DEF_BRAN),
member(LAB_BR_OBS,DEF_BRAN).
```

3) 함수 h는 일대일 (one-to-one) 이다.

위의 ECI의 조건은 2.2의 패턴웨손모델에 적용되어 각각 GPECI, SGECI, DSECI 알고리듬을 구성한다.

2.4 ARG 상에서의 Distance Measure

실제의 응용에 있어서 중요한 문제는 원형패턴과 입력패턴이 전적으로 동일한가의 문제가 아니라 두 패턴이 얼마나 유사한가 혹은 두 패턴 사이에 얼마나 차이가 있는가의 문제이므로 이를 위해 두패턴의 유사도 (similarity)가 정의되어야 한다.

본 연구에서는 두 ARG 사이의 distance 를 w' 로 부터 w 로 변형 (transformation) 시키는데 드는 비용으로 정의한다. 이때 변형은 노드와 브랜취에 대한 삭제 (deletion), 첨가 (insertion) 그리고 대체(substitution) 를 의미하며 구문론적 웨손과 의미론적 웨손을 모두 고려하여 distance 를 계산하였다.

3. ARG 상의 매칭 알고리듬

이 장에서는 그래프 매칭 과정을 상태 공간 문제(state space problem)로 모사하고, 이를 기초로 2.3에서 상정된 GPECI, SGECI, DSECI 을 찾는 알고리듬에 대해 설명한다.

3.1 상태 공간 모사 (state space description)

1) 상태 모사 (state description)

state는 다음과 같은 형태로 표현된다.

```
state( DISTANCE , MAPLIST )
```

MAPLIST : 현재까지 매칭짝들의 리스트

DISTANCE : distance 값

초기상태는 state(0,[map(0,0)]) 이며 distance 가 0 이며 매칭된 노드가 없는 상태를 나타낸다.

2) 오퍼레이터(operator)

다음과 같은 조건을 만족하는 N'의 원소 OBS와 N의 원소 PURE를 모두 찾는다.

```
find(OBS,PURE,MAPLIST) :- map_nod(OBS,PURE),
                          not(mem_obs(OBS,MAPLIST)),
                          not(mem_pure(PURE,MAPLIST)),
                          map_bran(OBS,PURE,MAPLIST).
```

MAPLIST 의 첫번째 원소를 map(OBS1,PURE1) 이라 할 때, OBS 를 OBS - OBS1 + 1 로 제한하면 중복되는 노드를 갖지 않는 상태공간을 구성할 수 있다. map_bran (OBS,PURE, MAPLIST)은 새로운 매칭짝인 map(OBS1,PURE1) 이 현재까지 확장된 매칭짝들과의 관계에 있어서 일관성이 유지되었는지를 검사 (consistency check) 한다. 이때 2.3의 exist_bran에 역방향에 대한 rule을 첨가한다.

3) 목표 상태 (goal state)

목표상태는 다음과 같이 표현할 수 있다.

```
state( TOT_DIST , [map(LAST_OBS,PURE)|REST] )
```

LAST_OBS는 N'의 마지막 매칭 대상 노드이다. 즉 목표상태는 N'의 모든 노드가 매칭된 상태를 의미하며 두 ARG 사이의 distance 는 TOT_DIST 이다.

위의 상태모사를 SGECI이나 DGECI에 적용시키기 위해서 오퍼레이터를 다음과 같이 변형시킨다. SGECI 에서는 N'∪(0) 와 N 에서, DGECI 에서는 N' ∪ (0) 와 N ∪ (0) 에서 다음에 확장시킬 매칭 대상 노드를 선택한다. 즉 SGECI 에서는 map(0,PURE), DGECI 에서는 map(0,PURE)나 map(OBS,0)와 같은 유사 매칭짝 (pseudo matching pair)을 가질 수 있다.

3.2 순서 탐색 알고리듬 (odered search algorithm)

3.2.1 GPECI 에 대한 순서 탐색 알고리듬

2.3의 ECI 조건과 3.1의 상태공간 모사를 기초로 GPECI에 대한 순서탐색 알고리듬을 구성할 수 있다. 이때 GPECI에 대한 휴리스틱 룰(heuristic rule)[1]을 만족해야만, 계속 확장시킬 노드들의 리스트인 OPEN에 포함시킨다.

3.2.2 SGECI 에 대한 순서 탐색 알고리듬

SGECI은 GPECI과 기본적으로 동일한 알고리듬을 적용시킬 수 있으며 GPECI를 특별한 경우로 포함한다. 목표상태에 포함되지 못한 N의 노드들은 그에 대응하는 N'의 노드가 존재하지 않는 것이므로 이에 대하여 유사 매칭짝 map(0, PURE)를 갖는 것으로 간주할 수 있다. 만약 이들 유사 매칭짝에 대한 distance를 고려하는 것이 의미가 있을 경우에는 유사 매칭짝의 각 원소 map(0,PURE)에 대한 distance 가 경로 비용에 포함되어 있어야 한다. 그러므로 3.1 의 목표상태를 N 의 모든 노드가 매칭된 상태의 모사로 바꾸어 준다. SGECI에 대한 순서탐색 알고리듬을 프로우 차트로 나타내면 [그림 3.1] 과 같다.

3.3. DSECI 에 대한 매칭 알고리듬

3.3.1 일 방향 매칭 (one way matching) 알고리듬

DSECI 에서는 구조가 다른 두 ARG 사이의 매칭을 수행할 수 있어야 하므로 3.2에서와 같은 휴리스틱을 사용할 수 없다. 그러므로 현재 까지의 매칭짝들에 대한 distance의 증가 즉 현 상태까지의 경로의 최소비용을 평가함수 (evaluation function)로 이용하여 상태공간 탐색을 수행한다. 이는 균일 비용 탐색과 기본적으로 동일하다.

DGECI에 대한 일 방향 매칭에서는 현 노드가 확장될 때마다 map(OBS,0)을 유효한 매칭짝으로 간주하여 MAPLIST에 덧붙여 계속 확장시킨다. 또한 현재의 확장 대상 노드에서 N' 의 모든 노드가 매칭되었으면서 N 의 노드 중 아직 매칭되지 않은 노드가 있을 경우에는 3.2.2 에서와 같이 map(0,PURE) 를 덧붙여 주는 오퍼레이터를 동작시킨다. DSECI 에서의 목표상태는 N' 와 N 의 모든 노드가 매칭된 상태이다.

DSECI 에서는 두 그래프의 구조가 전혀 다르더라도 최소의 distance 를 갖는 매칭 함수를 찾는 것을 목적으로 하기 때문에 OPEN 리스트가 비어있을 (empty) 수 없다.

[그림 3.3]은 [그림 3.2] 와 같은 두 ARG w' 및 w에 DSECI 에 대한 일방향 매칭 알고리듬을 적용시킨 결과이다.

3.3.2 양 방향 매칭 (two way matching) 알고리듬

3.3.1 에서는 현재의 노드를 확장할 때마다 map(OBS,0) 를 확장 가능한 매칭짝으로 간주해 주어야 하므로 N' 의 모든 노드가 매칭될 때까지, 확장을 할 때마다 유사 매칭짝을 포함하는 또하나의 노드를 항상 OPEN 에 첨가시켜 주어야 한다는 부담이 있다. 이를 피하기 위해서는 maximal clique 을 찾을 때까지 유사 매칭짝을 고려하지 않는 양방향 매칭을 수행하여야 한다. 일방향 매칭에서는 다음에 확장될 N'의 노드를, 오퍼레이터를 작동시키기 전에 미리 정하지만 양방향 매칭에서는 이런 제한을 두지 않는다. [그림 3.4]는 DSECI에 대한 양방향 매칭 알고리듬의 프로우 차트이며 오퍼레이터에 대한 레이블인 OPERATOR 에 따라 동작시킬 수 있는 오퍼레이터가 정해지게 된다.

일방향 알고리듬에서는 중복되는 노드가 발생하지 않으므로 CLOSED 리스트를 관리할 필요가 없지만, 양방향 매칭에서는 중복을 피하려면 확장 대상 노드가 이미 확장된 노드인지를 CLOSED와 비교해 주어야 한다. 또한 CLOSED 리스트와의 비교를 위해, 확장된 노드들을 OPEN 리스트에 첨가 시키기 전에 그 매칭짝들을 차례로 배열 (sorting) 해 줄 필요가 있다.

3.3.3 백 트랙킹 (back tracking) 알고리듬

지금까지의 각 알고리듬에서는 현 상태에서의 distance 를 평가함수로 사용하여 확장시킬 노드에 대한 순서를 재조정함으로써, 맹목적 탐색 (blind search) 을 피하면서도 최소의 distance 를 갖는 매칭 함수를 찾을 수 있었다. 그러나 3.3.1 과 3.3.2 에서는 N' 와 N 의 갯수가 증가함에 따라 관리해야 할 메모리가 과도하게 늘어날 수 있는 단점이 있다. 그러나 백트랙킹 알고리듬은 맹목적 탐색을 하지만

메모리의 제한을 받지 않을 수 있다. 이는 기본적으로 가장 최근에 생성된 노드를 가장 먼저 확장시키는 깊이 우선 탐색(depth first search) 방법이며 그 특성상 양방향 매칭을 이용한다. 다음은 그 개요이다.

1) 3.3.2 의 i)과 비슷한 오퍼레이터를 사용하여 가장 최근에 생성된 노드를 확장시킨다.

2) 1)의 오퍼레이터를 사용하여 더이상 확장시킬 수 없을 때, 즉 maximal clique 을 발견하면 그 이전 노드로 백 트랙킹하여 다시 확장을 계속한다.

3) 백 트랙킹이 일어날 때마다 유사 매칭확율 포함한 distance를 계산하여 현재까지 일어진 최적의 결과와 비교한다.

4. 그래프 변환 및 시스템 구성

4.1 그래프 변환

본 연구에서는 인식 대상 물체를 2차원적 구조를 갖는 thin object로 제한하여 물체의 골격에 해당하는 RVS(radial-valued-skeleton) 을 SPTA(safe point thinning algorithm)[4]을 사용하여 추출하였으며, 두께 (thickness)에 대한 특징값 (attribute value)으로 사용되는 radial-value는 SPTA의 적용시 대상 화소가 골격의 구성요소로 결정될 때까지의 반복 수행 횟수로 부터 얻을 수 있다. RVS는 generalized-cone 개념을 2-D에 적용시킨 것으로 thin object에 대한 적절한 묘사를 할 수 있다. [5]

다음은 RVS로 부터 skeletal-arm과 그들간의 연결관계에 대한 정보를 얻기 위해 [그림 4.1]과 같이 특징점(feature point)을 추출한다. 물체에 대한 RVS와 특징점을 추출해 낸 후에는 이를 skeleta-arm들로 분할할 수 있으며 특징점으로 부터 또 다른 특징점에 도달할 때 까지 연결점을 따라 스캐닝 하면서 얻을 수 있다.

이제 위의 과정을 거친 skeletal-arm 들을 물체에 대한 부분패턴으로 간주하여, 입력영상을 구성하는 각 패턴에 대한 ARG를 묘사한다. 입력패턴에 대한 ARG 묘사를 위해 [5]에서와 유사한 attribute 를 사용한다.

4.2 시스템 구성

그래프 변환기로 부터 얻어진 물체에 대한 묘사는 인식에 필요한 정보를 제공해 줄 뿐만 아니라 입력화상 및 패턴의 해석에 필요한 정보를 갖는 Short term knowledge 를 구성한다. 또한 Short term knowledge 에는 인식과정에서의 중간 결과가 덧붙어진다.

한편 물체에 대한 인식을 수행하기 위해서는 인식 대상 물체, 즉 원형패턴에 대한 정보를 미리 가지고 있어야 하는데 이는 영상의 해석에 필요한 룰(rule) 들과 함께 Long term knowledge 에 해당한다. Long term knowledge 는 전체 시스템의 상태에 영향을 받지 않는 정적 (static) 정보로서, 각 원형패턴에 대한 묘사와 영상의 해석 혹은 추론 (inference) 에 필요한 룰들로 구성된다. 공간관계 (spatial relation) 에 대한 룰은 다음과 같은 형태를 갖는다.

```
spatial_relation(CID1,CID2,above) :-
        cluster(CID1),cluster(CID2),
        y_max(CID1,Y1_MAX),y_min(CID2,Y2_MIN),
        Y1_MAX<Y2_MIN,
        above_cond(CID1,CID2).
```

매칭 결과는 다음과 같은 형태로 표현되며 Short term knowledge 에 등록된다.

```
candidate(CID,OBJECT,STATE)
  OBJECT : 물체의 이름
  STATE  : 3장에서의 목표상태와 동일.
```

다음은 동적 정보의 관리와 최적의 매칭 결과를 찾는 과정이다.

```
identifier(CID,OBJECT,STATE) :-
        best_match(CID,OBJECT,STATE),!.
identifier(CID,OBJECT,_) :-
        identify(CID,OBJECT),fail.
identifier(CID,OBJECT,state(MIN,LMAP)) :-
        findall(DIS,candidate(_,_,state(DIS,_)),L),
        minim_dis(L,MIN),
        candidate(CID,OBJECT,state(MIN,LMAP)),
        assertz(best_match(CID,OBJECT,state(MIN,LMAP)))
        del_all_candidate,!.
```

이미 물체 CID 에 대한 인식이 수행된 후라면 Short term knowledge 에 best_match(CID,OBJECT,STATE) 가 존재할 것이므로 다시 매칭을 수행할 필요가 없으며, 그렇지 않다면 아래와 같은 과정을 수행한다.

```
identify(CID,object1) :-
        cluster(CID),
        get_pure_data("object1.pure"),
        graph_matcher(CID,[state(0,[map(0,0)])],OBS,STATE),
        assertz(candidate(CID,object1,STATE)).
identify(CID,object2) :- ....
                  :
identify(CID,objectN) :- ....
```

매칭을 시도해야 할 원형패턴의 수 N이 늘어난다면 위와 같이 N번의 매칭이 수행된 이후라야 정확한 결과를 얻을 수 있다. 그러므로 효율적인 인식 시스템을 구성하기 위해서는 매칭 과정 이전에 매칭을 시도해야 할 원형의 수를 줄여줄 수 있는 경험적 지식(heuristics)이 필요하다.

5. 실험 및 결과 고찰

5.1 매칭 알고리듬의 비교

3장에서 제안된 각 알고리듬은 각기 상이한 패턴 훼손 모델을 기초로 하여 제안된 것이므로 매칭 속도, 메모리 소비 그리고 매칭 능력 면에서 상이한 결과를 나타낸다.

[표 5.1] 은 GPECI가 존재하는 [1]에서의 두ARG 에 대한 매칭을 각 조건에 따라 깊이 우선 (depth first), 균일 비용 (uniform cost), 3.2의 순서 탐색 알고리듬에 적용시킨 결과이다. SGECI는 GPECI를 특수한 경우로 포함하므로 매칭 결과는 동일하다.

위의 결과에서 알 수 있듯이 두 ARG 사이에 GPECI 혹은 SGECI가 존재할 때는 순서 탐색을 할 수 있으므로 매칭 속도 및 메모리 소비의 측면에서 가장 이상적인 결과를 얻을 수 있다. 같은 그래프에 대하여 3.3의 DSECI에 대한 각 알고리듬을 사용하여 매칭을 수행할 경우 동일한 매칭 결과를 얻을 수 있으나 순서 탐색 알고리듬에 비하여 5-10 배 이상의 시간과 메모리를 소비하였다. 그러나 GPECI 혹은 SGECI가 존재하지 않는 경우에는 3.3절 과 같은 DSECI 알고리듬을 통해서만 두 그래프 사이의 distance 를 계산할 수 있다.

Eshera와 Fu는 [2]에서, 두 그래프의 inexact matching을 상태공간 (state space)을 구성하는 부분과 최적 경로 (shortest path)를 찾는 부분으로 나누어 수행하였다. 이방법은 3.3.2 만큼의 메모리를 소비하면서 3.3.3 보다 더 많은 수행시간을 소비하게 된다.

5.2 영상 인식

3장의 그래프 매칭 알고리듬을 사용하여 thin object 에 대한 인식을 수행한다. 4장에서 제안한 그래프 변환기로 부터 입력 패턴에 대한 묘사를 한 후 이를 원형 패턴에 대한 ARG와 매칭한다. 이를 위한 전체 시스템 구성도는 [그림 5.1]과 같다.

다음은 입력영상([그림 4.1])에 대한 ARG 묘사를 그래프 변환기로 부터 얻은 후 물체의 인식 및 spatial relation 을 얻은 결과이다.

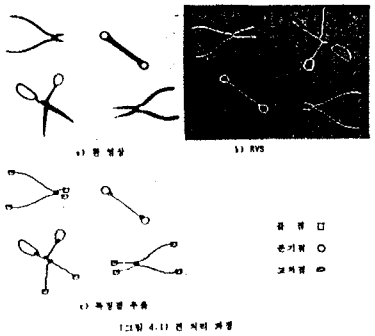
```
Goal : best_match(CID,OBJECT,_)
CID-1, OBJECT-plier
CID-2, OBJECT-wrench
CID-3, OBJECT-scis
CID-4, OBJECT-long_nose
4 Solutions
Goal : spatial_relation(CID1,CID2,RELATION)
CID1-1, CID-3, RELATION-above
CID1-2, CID-4, RELATION-above
CID1-3, CID-1, RELATION-below
CID1-4, CID-2, RELATION-below
CID1-1, CID-2, RELATION-left
CID1-3, CID-4, RELATION-left
CID1-2, CID-1, RELATION-right
CID1-4, CID-3, RELATION-right
8 Solutions
```

6. 결 론

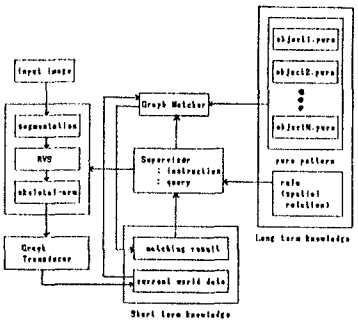
본 연구에서는 노드와 브랜치에 심볼(symbol) 및 속성값

(numerical attribute)을 부여하여 패턴에 대한 더욱 정확한 모사를 기할 수 있는 ARG(Attributed relational graph)와, ARG 상의 매칭 알고리듬을 상태공간 상의 탐색(state space search)을 기초로 제안하였다. 두 ARG 사이의 distance를 사용하여 입력패턴과 원형패턴 사이의 유사도 (similarity)를 측정할 수 있었으며, 그 결과 ARG 의 효율적인 표현 기능 뿐만 아니라 매칭 시간의 단축 및 매칭 결과에 대한 신뢰도를 향상시킬 수 있었다.

각 부분패턴과 그들간의 관계에 해당하는 노드와 브랜취를 얼마나 정확하고 적절하게 추출할 수 있는가의 문제는 매칭 과정에서의 효율성과 직결되는 문제이므로 앞으로의 연구 방향은 더욱 효율적인 매칭 알고리듬에 대한 연구뿐만 아니라, 원 영상으로 부터 효과적으로 ARG 모사를 추출할 수 있기 위한 연구가 병행되어야 할 것이다.



[그림 4.1] 전 처리 과정



[그림 5.1] 전체 시스템 구성도

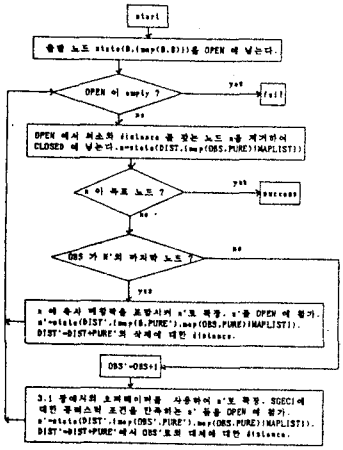
[1] W. H. Tsai and K. S. Fu, "Error-correcting isomorphism of attributed relational graphs for pattern analysys," IEEE Trans. Syst., Man, Cybern., vol.SMC-9,pp.757-768,1979.

[2] M. A. Eshera and K. S. Fu, "A graph distance measure for image analysis," IEEE Trans. Syst., Man, Cybern., vol.SMC-14,pp.398-408,1984.

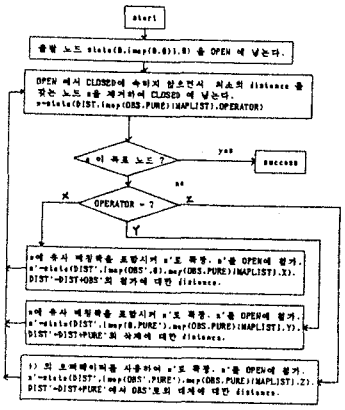
[3] M. A. Eshera and K. S. Fu, "An image understading system using attributed symbolic representation and inexact matching," IEEE Trans. Pattern Anal. and Machine Intell., vol.PAMI-8,pp.604-618,1986.

[4] N. J. Naccache and R. Shingal, " SPTA : A proposed algorithm for thinning binary patterns," IEEE Trans. Syst., Man, Cybern., vol.SMC-14,pp.409-418,1984.

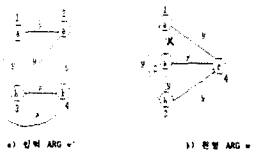
[5] A. C. Kak, K. L. Boyer, C. H. Chen, R. J. Safranek, and H. S. Yang, "A knowledge-based robotic assembly cell," IEEE Expert,vol.1,no.1,pp.63-83,1986.



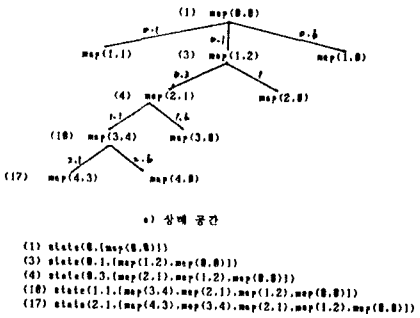
[그림 3.1] SGECI 알고리듬



[그림 3.4] DSECI two-way matching



[그림 3.2] 두 ARG w' 및 w



[그림 3.3] DSECI에 대한 일방향 매칭

(소수점 이하 1/100 초)

조 건	알고리듬	1)	2)	3)
GPECI	depth first	0.44	0.16	32
GPECI	uniform-cost	0.66	0.17	28
GPECI	ordered search	0.27	0.22	8
SGECI	ordered search	0.33	0.22	8

1) error-correcting 기능을 첨가 시켰을 때의 매칭 시간
2) error-correcting 기능이 없을 때의 매칭 시간
3) 1)의 경우에 발생된 노드의 갯수

[표 5.1] 각 알고리듬의 비교