Relational Structure Description and Matching Algorithm for 3D Objects

Deren Li
School of Remote Sensing and Informatics

Xinhua Wang
School of Electronic Engineering and Precision Mechanism
Wuhan Technical University of Surveying and Mapping
39, LuoYu Road , Wuhan 430070, P.R.China.

KEY WORDS: Robotics, Vision, Matching, Recognition, Artifical Intelligence.

Abstract:

When relational data structure is used to describe 3D objects, matching problem results in the problem of relational isomorphism or relational homomorphism. This paper presents the relational data structure describing 3D objects, and the trimming algorithm. It uses the multi-constraints and one to one correspondence constraint of relational data structure as the knowledge to trim the searching tree, that simplifies the problem of relational isomorphism in matching. Then the relational matching is determinated, combined with the unit distance function. Finally, a example is presented.

1. Introduction

Photogrammetry, combining with Pattern Recognition, Computer Vision (CV), has been become an important research area. The key of it is the describing method of knowledge. How to describe object determents not only the task and the goal of low--level approach, but also the high--level processing algorithm. Relational data structure is one of important description method. In relational structure description, 3D objects are composed of labeling basic units(for example, edges, surfaces, etc.), and the characters and their relation of basic units give out the description of structure feature of objects. A simplest relation (two-unit relation) is a graph. If the basic units are expressed by the node of graph, the edges of graph are used to express the relation between units. Then, the object matching problem become the sub-graph matching problem. Indeed, the computing perplex of sub-graph match is by index law (exponential function). On the other hand, the simplifies description can not satisfy the practical needs. Tsai and Fu[1] presented the conception of attributed relational graph which give some attributions for node and edge of graph. The description enhances the describing ability of graph, but it brings some problems for low-level processing. Shapiro[2] presents a method of relational description and proposes a relational data structure, which include different relations and multi--unit relations. In the description, the problem of object matching results in the problem of relational isomorphism or homomorphism. As known, isomorphism is the well known constraint satisfactory problem. Indeed, it is the consistent labeling problem. Up to now, computing spend is still the main problem of matching. Haralick's method [3] is efficient, but mathematically has to be processed. In fact, we can use the structure constraints of object to simplify the labeling problem.

In this paper, the authors present a relational data structure which uses the structure constraint to describe objects, and then present the trimming algorithms, which use those constraints to simplify labeling problem. Finally, a distance function of unit feature is evaluated to realize the precision matching. In section 5, a example is presented.

2. Relational data structure describing 3D object

A relational data structure is a two-unit graph D=(U,T), where: U is the basic unit set (each unit can has its attribution), shaped as a=(p,x) (p is the label of node, and $x=(x_1,x_2....x_n)$ is the feature vector). $T=(T_1,T_2....T_K)$ is relational set. Each relation $T_i(i=1......k)$ is a subset, and $T_i=(t_{i1},t_{i2}.....t_{im})$, where: $t_{ij}\subset U$, and $1<|t_{ij}|<|U|$, $|U|$ is the cardinal number of set U. The subset of Ti may be sequential or unsequential, and can be involved some feature vectors.

As to a simplifier object, its edges or surfaces can be considered as units of graph, and the geometric or structure constraints between units can be considered as the relations. Fig 1. is an overview of observed object and its model. Let us make a discussion about its relational data structure. Here, the edges are throughout as basic units, and each edges has its own attributions (category, parameters etc.). According to the geometric or structure constraint, the relational between units can be defined as: connecting relation T1, parallel relation T2, coplanar relation T3.

$T_1=\{(u_1,u_2....u_n)\,|\,u_i\in U$, connecting relation, $n\geq 2\}$

$T_2=\{(u_1,u_2....u_n)\,|\,u_i\in U$, parallel relation, $n\geq 2\}$

$T_3=\{(u_1,u_2....u_n)\,|\,u_i\in U$, coplanar relation, $n\geq 2\}$

Notice that: T_1,T_2,T_3 are unsequential set. On the other hand, each relations can has its attributions. For instance, Angle, connecting method etc. All of those relation description are not unique. Indeed, too much relation description will benefit consistent labeling, but will bring many problems for low-level processing. Based on above relation description, we can conclude the relation description as Tab 1~6 for the object as Fig 1. Where: U is the unit set of object. T is its relation set. L is the unit set of model, and S is the relation set of model. Thus, relational data structure can be defined as $D_a=(U,T)$ and $D_b=(L,S)$ for observed object and model respectively.

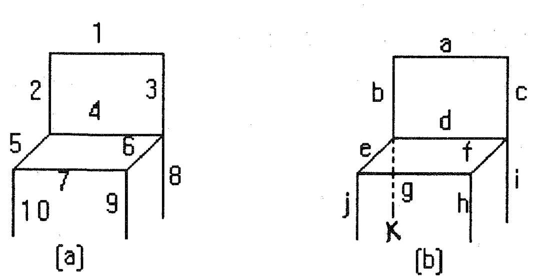

Fig.1 An overview of object and model using edges as units of graph

Notice that: because of the occluding, the edge k in object is lost.

3. Trimming algorithm

Consistent labeling problem can be expressed as a four--unit group (U,L,T,R), where: U is the unit set of observed object, $U=(u_1,u_2....u_n)$; L is the unit label set of model, $L=(l_1,l_2....l_n)$; T is the unit constraint set, shaped as $U=(u_1,u_2....u_n)$, that is a set of N--unit group, $u_\subset U$. Usually, not all N constraint can be labeled for $(u_1,u_2....u_n)$. So, we involved unit--label constraint relation R which tells us which l_i is reasonable labeling for group $(u_1,u_2....u_n)$. R is composed of the structure constraints of object and model.

Thus, we can use all methods of searching tree to consist labeling. There has been many algorithms[2] [4], but they almost put all constraints together to search. As the result, the computing spent is unbearable. So we developed a trimming algorithm as following.

Precision match demands that all matched units have the same structure. So each structure constraints can be considered as knowledge to trim searching tree. The simplifier and efficient algorithm is using unit attributions to trim searching tree. So we can define a unit--label table as:

$$H=\{H(u_1),\ H(u_2),.....\ H(u_m)\} \qquad (1)$$

Where: $H(u_i)$ is the set that can give labeling for unit u_i . Then, we can take further step to trim searching tree based on all constraint relations.

3.1 Trimming algorithm using relational subset

Assume that, we have know the relation data structure of a object expressed as $D_0=\{U,T\}$, and the relational data structure of model expressed as $D_0=\{L,S\}$, where: U, L are unit set and label set, respectively. $T=(T_1,T_2....T_K)$, $S=(S_1,S_2....S_K)$, T_j and S_j expresses a kind of relation. Thus we can define a relation subset table F as:

$$F(t)=\{\ s\in S_j\,|\ |\,s\,|\geq|\,t\,|\} \qquad (2)$$

Where: t has the same or similar relational attribution as s. Indeed, F(t) is the set which can label t labeling relation subset. The practical meaning of expression $|s| \geq |t|$ is that some units of object may be lost, because of occluding. So the number of label relational subset which is labeled for units can be even more.

Define $\omega(u)$ to express the set of unit relational subset that include unit u, that is :

$$\omega(u)=\{ t\in T_j \mid t\in U\} \tag{3}$$

Then:

$$H_j(u)= \bigcap_{t\in\omega(u)} \left(\bigcup_{s\in F(t)} S \right) \tag{4}$$

Where: $H_j(u)$ is the set of that can give u label according to constraint j. Obviously, we hope that the relational sub-isomorphism searched can meet all K relations. So the unit-label table H finally should be:

$$H(u)= \bigcap_{j=1}^{k} H_j(u) \qquad u\in U \quad j=1....k \tag{5}$$

3.2 The trimming algorithm using one to one correspondence

Assume that U,L are unit set and label set, respectively, and we have j units $(u_1,u_2....u_j) \subset U$, and that $H(u_j) \subseteq L' \subset L$, and $|L'| = J$. It means that those j labels can be labeled for the J units, and other units can not be labeled by those labels.

As to the subset (U_n, L_m), Where: $U_n \subset U$, $|U|=n$, $L_m \subset L$, $|L_m|=m$. for every $u_n \subset U$, and $H(u) \subset L_m$, we can get the conclusion as:

1. if m<n, then the relational sub--isomorphism is not exist. It means that a label can not label for more than two units, and the one to one correspondence is not exist.
2. if m>n, do not trimming.
3. if m=n, the trimming algorithm can be defined as:

$$\eta_1\ H(u_i)=\{ l\in H(u_i) \mid l\notin L_n,\ \text{if}\ u_i\notin U_n\} \tag{6}$$

and

$$\eta_1\ H=\{ \eta_1\ H(u_i) \} \qquad i=1,2,M \tag{7}$$

Where: $u_i \subset U$.

The trimming algorithm is iterative to execute for H, until the following equation is valid.

$$\eta_1{}^{k+1}H= \eta_1{}^{k}H \tag{8}$$

Usually, we begin the trimming processing from the subset which has the least units.

For the same principle, as to the relational sub-isomorphism, the correspondence of relational subset is also the one to one correspondence. So the η_1 trimming algorithm can also be applied for the trimming of relation subset tables.

3.3 Trimming algorithm using the correspondence between relational subset and unit-label table

In the procedure of making unit-label table, relational subset and unit-label table are acting each other. The correspondence between the relational subset and the unit-label table can help us to trim unit-label table.

Assume that: We have got a unit-label table $H_j(u)$ based on the constraint relation T_j, and subset (U_n,L_n) is existing, where: $U_n \subset U, Ln \subset L$. It means that the n labels should be labeled to the n units, and other units can not be labeled to those labels. The correspondence also exists between other unit-label table $H_i(u)$and relational subset T_i.

If unit-relation subset t_{ij} and label--relation subset s_{ij} are exited to the constraint relation T_i, define $\Omega(u)$ expressed unit-relation subset including unit u_n, $u \in U_n$ and define f(l) expressed label relational subset including l_n , $l \in L_n$, Then define trimming algorithm η_2 as:

$$\eta_2\ H_i(u)=\{\ l\in f(l)\ |\ l \notin f(l),\ \text{if } u \notin \Omega(u)\} \qquad (9)$$

In practice, we always begin to execute η_2 algorithm for constituting and trimming unite--label table $H_i(u)$ and relational subset T_i from the subset which has the least cardinal number in the unit--label table H_j, and iterate it until the following equation is valid.

$$\eta_2\ {}^{k+1}H = \eta_2\ {}^{k}H \qquad (10)$$

It should be point out that the correspondence relation also exits the correspondence between relational attribution and unit-label table.

4. Relational match

In practice, the existence of relation isomorphism usually do not means precision match. So we involved relation distance function to evaluate the precision match.

After above trimming approach, the deformation between two graphs (for object and model) mainly expresses as the deformation of units displacement. If we have two unit a=(s,x) and b=(t,y), where : s,t are unit labels, and x,y are the unit feature vectors shaped as $(x_1,x_2....x_m)$, $(y_1,y_2....y_m)$ respectively. Thus relational distance function can be defined as:

$$Ds = \sum_{i=1}^{m} g_i (x_i - y_i)^2 \qquad (11)$$

Where: g_i is weight.

Especially, if s=t, then Ds=(s⇒s)=0.

Thus, when Ds⇒min, the precision matching is determined .

5. Example

As to the object and model as Fig.1, the Table 1~6 are the relational data structure. based on formula (2), the relational subset tables F_1, F_2, F_3 can be expressed as table 7~9 by using the constraints (T_1,T_2,T_3). Moreover, table 10~12 are the F_i (i=1,2,3), after executing the trimming approach η_1 . Then according to the formula (3)(4), we can get every unit-label tables (H_1,H_2,H_3) as table 13~15, based on relational constraints. Following formula (5), intersecting result of unit relational tables is the unit-label table expressed as Table 16. Continually executing the trimming algorithm of η_2, we can get a new result as Table 17. Finally, we obtained the unit-label table expressed as Tab 18. after executing distance function (8).

Above results express that, after executing η trimming, Table 17 has almost been a fully trimmed searching tree. precision match by means of distance function is easy. At the same time, the computing spent will obviously reduce. Fig. 2 is another example, and Fig. 2 (a) and (b) are the object and model, respectively. Here, we use surfaces as units, the relations between units can be defined as: adjacency and insect-surface relation T_1, parallel relation T_2, adjacency and insect--point relation T_3.

$T_1=\{(u_1,u_2....u_n)\ |\ u_i\in U$, adjacency relation with common surface, $n\geq 3\}$

$T_2=\{(u_1,u_2....u_n)\ |\ u_i\in U$, parallel relation, $n\geq 2\}$

$T_3=\{(u_1,u_2....u_n)\ |\ u_i\in U$, adjacency relation with common point, $n\geq 3\}$

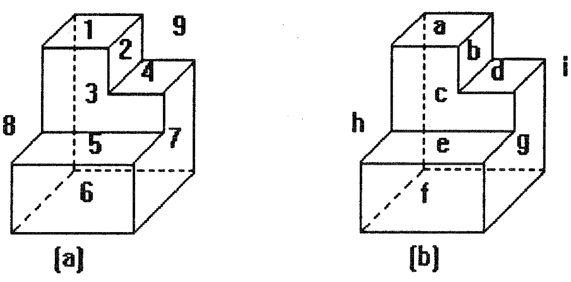

Fig. 2 An overview of object and model using surfaces as units

For the similar reason, every units has its own attributions. Base on above relation description, we can get relational description as Tab.(19)~(22) (For the reason of simplification, we only make a discussion of relation matching using the relation description T_1 and T_2). The unit-label table H_1 is expressed as Tab.(23), following formula (3) and (4). At the same time, H_1 is involved to trim the unit-label table H_2 . Tab.(24) is the result of H_2 which is trimmed by η_2 algorithm. Continually executing the η_2 algorithm, finally we can get the trimmed tree as Tab.(25). Tab.(26) is the result of matching, after executing distance function.

6. Conclusion

Form above theory analysis and examples, we can see that the description based on relational data structure makes the relational description very feasibility and precision. The trimming approach that the paper presented by using relational subset, one to one correspondence and one to one correspondence between relational subsets and unit-label table can efficiently simplify the procedure of consistency labeling. Finally the distance function make it possible to realize precision match.

It should be noticed that, as to more complex recognition for more complex object, if selecting even more unit attributions and relational constraints, we also can get well simplified results. On the other hand, the idea of the paper is popular, and it also can be used for the recognition task using curved surfaces as units, if the relational data structure is structed by means of the proper structure constraints. The trimming algorithm and distance function in the paper are also available.

Reference

1. Tsai and Fu. "Error-correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis", IEEE. Trans. on SMC. Vol. SMC-9. No. 12. p757-768. 1979.

2. Shapiro. L. G., Moriorty. R.M. et. "Matching Three-Dimension Objects Using a Relational Paradigm". Pattern Recognition . Vol.17, No.4.,p385-405, 1984.

3. Haralick. R.M.,Shapiro.L.G. "The Consistent Labeling Problem". IEEE. Trans., PAMI., Vol. PAMI-2, No.3,p04- 519, 1980.

4. P.J.Flynn and A.K.Jain. "CAD-based Computer Vision: from model to relation graphs". IEEE Trans.PAMI, Vol.13, No.2,PP.114-132,1991.

Tab.1 Connecting relaton T_1

t_{1i}	unit
t_{11}	3,4,6,8
t_{12}	2,4,5
t_{13}	5,7,10
t_{14}	6,7,9
t_{15}	1,2
t_{16}	1,3

Tab.2 Connecting relaton S_1

s_{1i}	Label
s_{11}	c,d,f,i
s_{12}	b,d,e,k
s_{13}	e,g,i
s_{14}	f,g,h
s_{15}	a,b,
s_{16}	a,c

Tab.3 Parallel relation T_2

t_{2i}	Label
t_{21}	2,3,8,9,10
t_{22}	1,4,7
t_{23}	5,6

Tab.4 parallel relation S_2

s_{2i}	Unit
s_{21}	b,c,i,j,h,k
s_{22}	a,d,g
s_{23}	e,f

Tab.5 Coplanar relation T_3

t_{3i}	Unit
t_{31}	1,2,3,4,8
t_{32}	4,5,6,7
t_{33}	3,6,8,9
t_{34}	2,5,10
t_{35}	7,9,10

Tab.6 Coplanar relation S_3

s_{3i}	Unit
s_{31}	a,b,c,d,i,k
s_{32}	d,e,f,g
s_{33}	c,f,h,i
s_{34}	b,e,j,k
s_{35}	g,h,j

Tab.7 F_1

t_{1i}	$F_1(t_{1i})$
t_{11}	s_{11},s_{12}
t_{12}	$s_{11},s_{12},s_{13},s_{14}$
t_{13}	$s_{11},s_{12},s_{13},s_{14}$
t_{14}	$s_{11},s_{12},s_{13},s_{14}$
t_{15}	$s_{11},s_{12},s_{13},s_{14},s_{15},s_{16}$
t_{16}	$s_{11},s_{12},s_{13},s_{14},s_{15},s_{16}$

Tab.8 F_2

t_{2i}	$F_2(t_{2i})$
t_{21}	s_{21}
t_{22}	s_{21},s_{22}
t_{23}	s_{21}, s_{22}, s_{23}

Tab.9 F_3

t_{31}	s_{31}
t_{32}	s_{32},s_{33},s_{34}
t_{33}	s_{32},s_{33},s_{34}
t_{34}	$s_{32},s_{33},s_{34}, s_{35}$
t_{35}	$s_{32},s_{33},s_{34}, s_{35}$

Tab.10 $\eta_1 F_1$

t_{1i}	$F_1(t_{1i})$
t_{11}	s_{11},s_{12}
t_{12}	$s_{11},s_{12},s_{13},s_{14}$
t_{13}	$s_{11},s_{12},s_{13},s_{14}$
t_{14}	$s_{11},s_{12},s_{13},s_{14}$
t_{15}	s_{15},s_{16}
t_{16}	s_{15},s_{16}

Tab.11 $\eta_1 F_2$

t_{2i}	$F_2(t_{2i})$
t_{21}	s_{21}
t_{22}	s_{22}
t_{23}	s_{23}

Tab. 12 $\eta 1$ F_3

t_{3i}	$F_3(t_{3i})$
t_{31}	s_{31}
t_{32}	s_{32},s_{33},s_{34}
t_{33}	s_{32},s_{33},s_{34}
t_{34}	$s_{32},s_{33},s_{34}, s_{35}$
t_{35}	$s_{32},s_{33},s_{34}, s_{35}$

Tab.13 H1

Unit	Label
1	a,b,c
2	b,c
3	b,c
4	b,c,d,e,f,k,i
5	b,c,d,e,f,g,h,i,j,k
6	b,c,d,e,f,k,i
7	b,c,d,e,f,g,h,i,j,k
8	b,c,d,e,f,k,i
9	b,c,d,e,f,g,h,i,j,k
10	b,c,d,e,f,g,h,i,j,k

Tab.14 H2

Unit	Label
1	a,d,g
2	b,c,i,j,h,k
3	b,c,i,j,h,k
4	a,d,g
5	e,f
6	e,f
7	a,d,g
8	b,c,i,j,h,k
9	b,c,i,j,h,k
10	b,c,i,j,h,k

Tab.15 H3

Unit	Label
1	a,b,c,d,i,k
2	a,b,c,d,i,k
3	a,b,c,d,i,k
4	a,b,c,d,i,k
5	b,c,d,e,f,g,h,i,j,k
6	b,c,d,e,f,g,h,i,j,k
7	b,c,d,e,f,g,h,i,j,k
8	a,b,c,d,i,k
9	b,c,d,e,f,g,h,i,j,k
10	b,c,d,e,f,g,h,i,j,k

Tab.16 H

Unit	Label
1	a
2	b,c
3	b,c
4	d
5	e,f
6	e,f
7	d,g
8	b,c,i,k
9	b,c,i,j,h,k
10	b,c,i,j,h,k

Tab.17 η_2H

Unit	Label
1	a
2	b,c
3	b,c
4	d
5	e,f
6	e,f
7	g
8	i,k
9	j,h
10	j,h

Tab.18 DS

Unit	Label
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	i
9	h
10	j

Tab.19 Adjacency with common surface T1

t_{1i}	Unit	co-surface
t_{11}	1,2,3,8,9	1
t_{12}	1,2,3,4,9	2
t_{13}	1,3,4,5,7,8	3
t_{14}	2,3,4,7,9	4
t_{15}	3,5,6,7,8	5
t_{16}	5,6,7,8	6
t_{17}	3,4,5,6,7,9	7
t_{18}	1,3,5,6,8,9	8
t_{19}	1,2,4,7,8,9	9

Tab.20 Adjacency with common surface S1

s_{1i}	Unit	co-surface
s_{11}	a,b,c,h,i	a
s_{12}	a,b,c,d,i	b
s_{13}	a,b,c,d,e,g,h	c
s_{14}	b,c,d,g,i	d
s_{15}	c,e,f,g,h	e
s_{16}	e,f,g,h	f
s_{17}	c,d,e,f,g,i	g
s_{18}	a,c,e,f,h,i	h
s_{19}	a,b,d,g,h,i	i

Tab. 21 parallel relation T_2

t_2i	Unit
t_{21}	1,4,5
t_{22}	3,6,9
t_{23}	2,7,8

Tab.22 parallel relation S_2

s_{2i}	Unit
s_{21}	a,d,e
s_{22}	c,f,i
s_{23}	b,g,h

Tab. 23 H_1

Unit	Label
1	a,b,d,e
2	a,b,d,e
3	c
4	a,b,d,e
5	a,b,d,e
6	f
7	g,h,i
8	g,h,i
9	g,h,i

Tab.24 η_2H

Unit	Label
1	a,b,d,e
2	a,b,d,e
3	c
4	a,b,d,e
5	a,b,d,e
6	f
7	g,h
8	g,h
9	i

Tab.25 η_2H

Unit	Label
1	a,d
2	b
3	c
4	a,d
5	e
6	f
7	g,h
8	g,h
9	i

Tab.26 DS

Unit	Label
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i