International Journal of Hybrid Information Technology
Vol.8, No.1 (2015), pp.353-360
http://dx.doi.org/10.14257/ijhit.2015.8.1.31

2D Geometric Constraint Optimum Solving Based on Problem Decomposition

Xue-Yao Gao[1], Chun-Xiang Zhang[2] and Zhi-Mao Lu[3]

[1]*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China*
[2]*School of Software, Harbin University of Science and Technology, Harbin 150080, China*
[3]*School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China*
z6c6x666@163.com

Abstract

Constraint solving is widely applied to many fields including computer aided design, 2 dimension (2D) model design and computer aided manufacturing. Geometric constraint solution is a difficult problem because there are a large number of entities and related parameters in 2D sketches. In this paper, a new method which decomposes geometric constraint relations based on entity-parameter graphs is proposed for reducing the size of constraint solution. A geometric constraint problem is decomposed into many independent sub-problems. Then, particle swarm optimization algorithm is used to solve constraint equations in each sub-problem. Solutions of all sub-problems are integrated to obtain the original problem's solution. In experiments, the proposed method is applied to HUST-CAID system. Experimental results show that the method can effectively solve 2 dimension geometric constraints.

Keywords: *constraint solving; 2D sketches; geometric constraint relations; particle swarm*

1. Introduction

In CAD system, geometric constraint is often used to describe entities and relationships between two entities in 2 dimension sketches. Geometric constraint solution is that when design requirements are modified by users, graphics which meet new constraints are drawn automatically based on the existing design sketches in CAD system. The quality of constraint solving is very important to the performance of CAD modeling system. Cao presents a broader range handling method to unify special constraints with ordinary constraints and gives a transformation approach to transform under-constrained problems into well-constrained ones. A d-tree decomposition algorithm is described based on node degrees in geometric constraint graphs [1]. Imbach applies geometric knowledge to specialize a so-called coefficient parameter continuation in 3D geometric constraint systems. Even though this method does not ensure to obtain all solutions, it provides several real ones. In new approach, geometric knowledge is used to search new solutions [2]. Albarelli casts the problem into a game-theoretic framework for guiding the inlier selection towards a consistent subset of correspondences. It makes geometric constraints depend on motion parameter knowledge and some semi-local properties estimated from local appearances of image features [3]. Haller presents a kind of body and cad structure which is constrained by pairwise coincidence, angular and distance constraints. At the same time, 21 relevant geometric

ISSN: 1738-9968 IJHIT

constraints are given and their corresponding infinitesimal rigidity theory for these structures is proposed [4]. Zhang gives a reconfiguration theory of geometric puzzles to model the topology change, in which methods of partition and assembly process analysis are given. It aims to extract kinematic chains as links and joints. At the same time, the puzzle unlocking method is described. Configuration constraint rearrangement is defined as reconfiguration links and joints [5]. Yeguas presents a new method to model large and complex three-dimension scenes, which can process any model intuitively provided by users. It exploits adjacent relationships between shapes and objects in model [6]. Ait-Aoudia proposes a decomposition-recombination planning algorithm, in which 2 dimension geometric constraint problems are solved by a graph reduction method. Based on key concept of skeletons, the complete constraint problem is decomposed into sub-constraint problems. Then, solutions of these sub-problems are recombined as the whole problem's solution [7]. Mathis integrates decomposition with reparameterization in order to reparameterize and decompose a geometric constraint system. In process of reparameterization, its purpose is not to minimize the number of added constraints, but to decompose the system in which each one owns a minimal number of such added constraints [8]. Liu defines equivalences of geometric constraint graphs and gives a novel method to transform a closed-loop constraint graph into an equivalent open-loop one. DOF reduction analysis and disturbing method is applied to identify whether these closed-loop constraints can be broken from a certain edge [9]. Yi decomposes constraints into two categories including original constraints and additional constraints for a geometric constraint multi-solution problem. Multiple solutions are found from constraint equations. Then, genetic algorithm and ant algorithm are combined to search for the optimal solution from these solutions [10]. Gao decomposes a 2-dimensional or 3-dimensional constraint problem into C-tree. When the proposed method is applied, a geometric constraint problem is decomposed into basic merge patterns. Its purpose is to obtain the smallest geometric constraint problem when the original problem is solved [11]. Joan-Arinyo uses non-trivial-width interval parameters to solve general geometric constraint problems. These parameters may be irrelevant to the problem's domain. Experiments show that it can solve geometric problems with tolerances, check constraint feasibility and analyze link motion of planar mechanisms [12].

In this paper, entity-parameter graphs are adopted to describe geometric constraint relations in 2 dimension CAD model. When entity-parameter graphs are decomposed, a geometric constraint set related to modification operations is gotten. The purpose is to decrease the size of constraint solving. At the same time, particle swarm optimization algorithm is applied to solve this geometric constraint set.

2. Decompose Geometric Constraints based on Entity-parameter Graphs

In the process of establishing 2D CAD models, new geometric constraints are introduced. When a model is edited, these existing geometric constraints will change. In order to obtain a model satisfied by customers, all geometric constraint equations are synthesized, and simultaneous equations are solved. In a 2D CAD model, there are a large number of entities. A large number of parameters are involved in each entity. In order to describe constraint relationships between two entities, multiple geometric constraint equations are used. In addition, constraint equations often contain multiple parameters. For a simple 2D CAD model, the solution of its constraint equations is complex and difficult.

A 2D CAD model is often composed by many entities. An entity is only associated with several entities, but it is not relevant to the rest ones. Operations for a model are often a gradual process. Every operation is conducted based on the existing model, and an entity is only involved. An entity's change is reflected in changes of its parameters, which causes adjustments of some geometric constraint relationships. But the rest constraint relationships are not affected. These affected constraint equations are synthesized to get the original problem's solution, in which original complex equations need not to be solved. The size and difficulty of constraint solution are reduced. Therefore, complex model operations can be transformed into many simple ones. In this paper, entity-parameter graphs are used to describe 2D CAD models. Geometric constraint equations related to model operations are gotten by decomposing entity-parameter graphs.

Entity-parameter graph is a mixed graph and often used to describe 2D CAD models. An entity-parameter graph is formally defined as EPG=(E, P, C). E is a set which contains all entities in this model. Here, entities are 2D basic geometric graphs including vertexes, lines, curves, circles, arcs and etc. P is a set which contains parameter information of all entities. In order to determine the entity which a parameter belongs to, the form **entity**. **parameter** is used to describe a parameter. C is a set which contains all geometric constraints in model. In a geometric constraint, entities and parameters related to it are defined. In an entity-parameter graph, circles are used to denote entities and boxes are used to denote parameters in entities. At the same time, a directed arc is drawn from a circle representing an entity to a box representing its parameter. Relationships between entities and entities are described by geometric constraints. Triangle is used to denote geometric constraints. Meanwhile, an undirected arc is drawn from a triangle representing geometric constraint to a box representing its parameter. A 2D CAD model is shown in Figure 1, and its entity-parameter graph is shown in Figure 2.

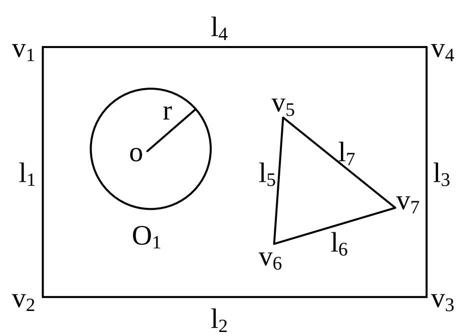

Figure 1. 2D CAD Model

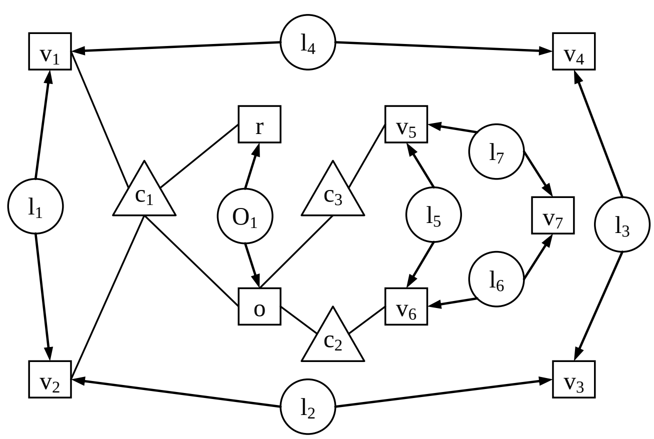

Figure 2. Entity-parameter Graph of 2D CAD Model

Geometric constraint equation CE changes because model operations are implemented. The steps of decomposing geometric constraints in model are shown as follows:

(1)Initialize set A and set S. A=Φ, S=Φ.

(2)Travel entity-parameter graph EPG and triangle node which corresponds to CE is found.

(3)Start from triangle node and box nodes are found by undirected arcs. All parameters denoted by theses boxes are added into set S.

(4)while(S!=Φ){

①An element par is selected from set S.

② Travel entity-parameter graph EPG and all triangle nodes denoting geometric constraints which are associated with it are found by undirected arcs. Its corresponding constraint equation is CF.

③A=A+{CF}.

④Start from triangle node denoting CF and box nodes denoting parameters are found by undirected arcs. All parameters are added into set S.

⑤S=S-{par}.

}

(5)Output all geometric constraint equations in set A.

3. Geometric Constraint Solving

Decompose entity-parameter graphs, relevant geometric constraint equations are obtained including $CE_1(X)=0$, $CE_2(X)=0$, …, $CE_n(X)=0$. Here, parameter X is feature vector, $X=(x_1, x_2, \ldots, x_n)^T$. These n equations are synthesized and solved. The solution of X can be gotten. However, the solving load is very great. In this paper, geometric constraint equations are respectively squared and results are summed. The sum is set to 0, which is shown in equation (1).

$$F(X) = \sum_{i=1}^{n} CE_i^2(X) = 0 \tag{1}$$

The solution of equation (1) is consistent with that of original equations. Here, the solution of algebraic equations is converted into an optimization problem in order to get feature vector X that minimizes the value of function F(X). Particle swarm optimization algorithm is applied to solve the problem.

Suppose that there are m particles in particle swarm. In the (k+1)th iteration, the ith particle updates its velocity and position according to equation (2) and equation (3).

$$v_{id}^{k+1} = v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{id}^{k}) \tag{2}$$

$$x_{id}^{k+1} = x_{id}^{k+1} + v_{id}^{k+1} \tag{3}$$

Here, x_{id}^{k} denotes the ith particle's current position, and its fitness function value is fitness_i. p_{id}^{k} represents the best position which the ith particle has passed through and its fitness function value is pbestvalue_i. p_{gd}^{k} denotes the current optimal particle's position in this swarm and its fitness function value is gbestvalue. Here, c_1 and c_2 are learning factors. r_1 and r_2 are random numbers. w is a weighting coefficient. k is the current iteration number. N is the number of iterations.

Particle swarm optimization algorithm is applied to solve geometric constraint equations. At the beginning, c_1, c_2, r_1, r_2, w and N are respectively initialized. In the (k+1)th iteration, velocity v_{id}^{k+1} and position x_{id}^{k+1} of the ith particle are updated respectively according to equation (2) and equation (3). In the process of optimum solving, each particle's fitness function value is computed according to F(*X*) as shown in equation (1). When current iteration number *k* is larger than *N*, the iteration process is over. Then the optimal solution of feature vector *X* is gotten and it is the optimal solution of geometric constraint equations.

4. Experiment

HUST-CAID system is developed by Institute of Computer Application Technology in Harbin University of Science and Technology. The proposed method is applied to HUST-CAID system. Source model is shown in Figure 3. There are a rectangle, two triangles and three circles. Circle 1 is a big circle. The center of circle 1 is (100, 150) and its radius is 200. Circle 2 and circle 3 are two small circles. The center of circle 2 is (160, 230) and its radius is 100. The center of circle 3 is (40, 70) and its radius is 100. Circle 1 is tangent internally with circle 3. Circle 2 is tangent externally with circle 3.

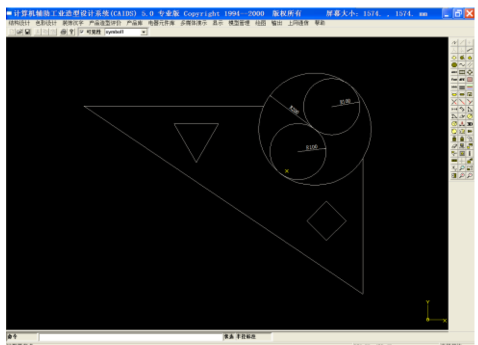

Figure 3. Source Model

Parameters in source model are modified. The center of circle 2 is changed into (172, 246) and its radius is changed into 80. The center of circle 3 is changed into 120. The center (x_1, x_2) of circle 3 need to be solved. Under geometric constraint that circle 2 is externally tangent with circle 3, equation (4) can be determined. It is shown in equation (4).

$$(x_1 - 172)^2 + (x_2 - 246)^2 = 200^2 \tag{4}$$

Under geometric constraint that circle 1 is internally tangent with circle 3, equation (5) can be determined. It is shown in equation (5).

$$(x_1 - 100)^2 + (x_2 - 150)^2 = 80^2 \tag{5}$$

Equation (4) and equation (5) are integrated according to the above method. So, fitness function F(*X*) is gotten and it is shown in equation (6), *X*=(x_1, x_2).

$$F(X) = ((x_1 - 172)^2 + (x_2 - 246)^2 - 200^2)^2 + ((x_1 - 100)^2 + (x_2 - 150)^2 - 80^2)^2 \tag{6}$$

Particle swarm optimization algorithm is applied to solve geometric constraint equations as shown in equation (6). Global optimal solutions in iterations are shown in Table 1.

Table 1. Global Optimal Solution in Each Iteration

Iteration number	Optimal solution of x_1	Optimal solution of x_2
2	-4.613601	-1.7069086
3	45.923782	90.324615
5	60.405518	79.857704
12	61.942802	79.22548
15	59.009422	80.883545
18	60.12772	80.31477
23	57.898373	81.87185
25	50.154457	87.513756
27	52.18459	85.852394
35	53.527077	84.872955
38	53.400173	84.964226
39	51.586777	86.31332
41	51.783245	86.1634
44	52.055553	85.95858
45	52.11426	85.91442

From Table 1, it can be seen that optimal solution of x_1 is 52.11426 and optimal solution of x_2 is 85.91442 at the 45th iteration. Equations (4) and equation (5) are synthesized. Then, simultaneous equations are solved manually. The exact solution of x_1 is 52 and the exact solution of x_2 is 86. It can be found that solutions in particle swarm optimization algorithm are very close to exact ones. Parameters in source model are modified according to the obtained optimal solutions. The target model is shown in Figure 4.

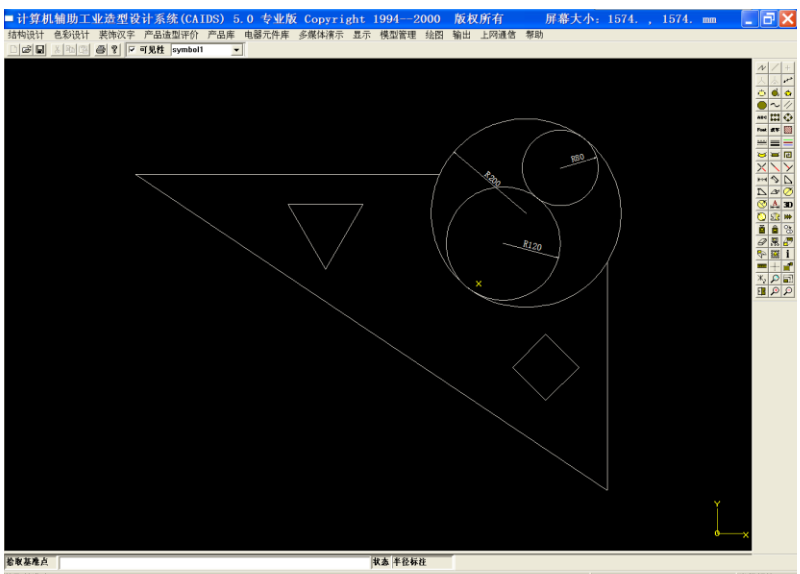

Figure 4. Target Model

5. Conclusions

2D geometric constraint solving is a key problem in CAD modeling. In this paper, entity-parameter graphs are used to decompose 2D geometric constraints. The complex geometric constraint problem is decomposed into several independent sub-problems. At the same time, particle swarm optimization algorithm is applied to solve constraint equations of each sub-problem. The proposed method is integrated into HUST-CAID system, and its performance is improved.

Acknowledgements

This work is supported by Science and Technology Research Funds of Education Department in Heilongjiang Province under Grant Nos. 12541125.

References

[1] C. H. Cao, P. Wang and L. G. Cao, "Well-constrained and under-constrained geometric constraint solving based on D-tree decomposition", Journal of Northeastern University, vol. 35, (**2014**), pp. 626-629.
[2] R. Imbach, P. Schreck and P. Mathis, "Leading a continuation method by geometry for solving geometric constraints", CAD Computer Aided Design, vol. 46, (**2014**), pp. 138-147.
[3] A. Albarelli, E. Rodolà and A. Torsello, "Imposing semi-local geometric constraints for accurate correspondences selection in structure from motion: a game-theoretic perspective", International Journal of Computer Vision, vol. 97, (**2012**), pp. 36-53.
[4] K. Haller, "Body-and-cad geometric constraint systems", Computational Geometry: Theory and Applications, vol. 45, (**2012**), pp. 385-405.
[5] L. P. Zhang and J. S. Dai, "Reconfiguration mechanism with interlocking geometric constraints from puzzles", Proceedings of the ASME Design Engineering Technical Conference, Chicago, (**2012**), pp. 1169-1176.
[6] E. Yeguas, "Example-based procedural modelling by geometric constraint solving", Multimedia Tools and Applications, vol. 60, (**2012**), pp. 1-30.
[7] S. Ait-Aoudia and S. Foufou, "A 2D geometric constraint solver using a graph reduction method", Advances in Engineering Software, vol. 41, (**2010**), pp. 1187-1194.
[8] P. Mathis, P. Schreck and R. Imbach, "Decomposition of geometrical constraint systems with reparameterization", Proceedings of the ACM Symposium on Applied Computing, Trento, (**2012**), pp. 102-108.
[9] Y. Liu, J. H. Yong and B. Wang, "Solving 3D geometric constraints for a class of closed-loop assemblies", Journal of Computer-Aided Design and Computer Graphics, vol. 20, (**2008**), pp. 1171-1175.
[10] R. Q. Yi, W. H. Li, H. Yuan, D. Wang and W. Guo, "Multi-solutions problem in geometric constraint", Journal of Jilin University, vol. 38, (**2008**), pp. 871-875.
[11] X. S. Gao, Q. Lin and G. F. Zhang, "A C-tree decomposition algorithm for 2D and 3D geometric constraint solving", Computer-Aided Design, vol. 38, (**2006**), pp. 1-13.
[12] R. Joan-Arinyo and N. Mata, "A constraint solving-based approach to analyze 2D geometric problems with interval parameters", Proceedings of the Symposium on Solid Modeling and Applications, Ann Arbor, (**2001**), pp. 11-17.

Authors

Xue-Yao Gao, she is Ph.D. She is also an associate professor in Harbin University of Science and Technology. Her research interests are CAD and model retrieval. She has authored and coauthored more than 20 journal and conference papers in these areas.

Chun-Xiang Zhang, Corresponding author. He is Ph.D. He is also a professor in Harbin University of Science and Technology. His research interests are CAD and machine learning. He has authored and coauthored more than 50 journal and conference papers in these areas.

