The Correctness Proof of a Decomposing Approach*

YUAN Bo, LI Yan-tao, SUN Jia-guang

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

E-mail: yuanbo@public2.east.cn.net; lyt@ncc.cs.tsinghua.edu.cn

http://ncc.cs.tsinghua.edu.cn

Received June 4, 1999; accepted December 3, 1999

Abstract: Mathematically a 2D constrained design system can be modeled by m independent nonlinear equations with n design variables and the design process can be viewed as a process of solving a geometric constraint system. Design decomposition is a highly effective way to improve a geometric constraint solver to make it efficient and robust. This paper reports a graph based decomposing approach and gives the correctness proof of the approach: (1) this approach can deal with the decomposition of structurally under-constrained systems, (2) this approach can detect structurally over-constrained systems, (3) the approach can terminate within finite number of steps, and (4) the solving steps obtained through the decomposing approach are structurally consistent.

Key words: design decomposition; geometric constraint solving; graph clipping; graph reducing

A 2D constrained design system can be treated as a set of nonlinear equations. The constraints correspond to nonlinear equations and the independent parameters of the 2D geometric elements correspond to design variables. To solve a 2D constrained design system efficiently and robustly, many research efforts have been made. One of the main efforts is to decompose a complex design system into a series of simple subsystems, which is called design decomposition[1]. The constraint solving process with design decomposition consists of two phases: (1) decomposing phase and (2) execution phase[2]. A design system can be represented by a constraint graph whose vertices correspond to geometric elements and whose edges correspond to constraints. This paper describes a graph based decomposing approach extended from graph constructive approaches[3~6], and presents the correctness proof of the decomposing approach on the basis of some restrictions and definitions.

1 C-R Approach

In the following sections, the geometric elements in a 2D constrained design system include points, lines, circles, line segments and circular arcs. A geometric element has its own degrees of freedom (DOFs), which allow it to vary in shape, position, size, and orientation. For example, the DOFs of a point, a line and a circle are 2, 2 and 3 respectively. According to the involved scope, the geometric constraints in a design system can be classified into two classes: global ones and local ones. The global constraints include horizontal (line), vertical (line) and fixed

---

* This project is supported by the National Natural Science Foundation of China under Grant No. 69772019 (国家自然科学基金). **YUAN Bo** was born in 1971. He received his Ph.D. degree in the Department of Computer Science and Technology of Tsinghua University. His current research areas include parametric design, solid modeling, and assembly design. **LI Yan-tao** was born in 1975. He is a Ph.D. student at the Department of Computer Science and Technology of Tsinghua University. His current research areas include geometric constraint solving and feature-based design. **SUN Jia-guang** was born in 1946. He is a professor and doctoral supervisor of the Department of Computer Science and Technology of Tsinghua University. His current research areas include computer-aided design, computer graphics, and software engineering.

(point, line or circle), while the local constraints include incidence, parallelism, perpendicularity, tangency, distance, angle and so on.

A geometric constraint reduces the DOF of a design system by a certain number, called the valency of the constraint[6]. The valencies of most geometric constraints are 1, but some constraints like the distance of two lines have a valency more than 1.

A geometric constraint system can be represented precisely by an undirected graph, and whose vertices correspond to geometric elements and whose edges correspond to geometric constraints respectively.

Let $G=\{V,E\}$ be a constraint graph, and some notations are defined for further discussion:

Definition 1. The DOF of a vertex $v\in V$ is equal to the DOF of the corresponding geometric element, noted as $DOF(v)$.

Definition 2. The Degree of constraint (DOC) of a vertex v is the sum of the valencies of the edges incident to v, noted as $DOC(v)$.

Definition 3. The DOFs of a constraint graph $G=\{V,E\}$ is the sum of DOFs of all vertices in V, noted as $DOF(G)$.

Definition 4. The DOC of a constraint graph $G=\{V,E\}$ is the sum of valencies of all local edges in E, noted as $DOC(G)$.

Definition 5. A graph G is structurally over-constrained if and only if there exists a sub graph $G_s\subseteq G$ such that $DOC(G_s)>DOF(G_s)-3$.

Definition 6. A graph G is structurally under-constrained if it is not structurally over-constrained and $DOC(G)<DOF(G)-3$.

Definition 7. A graph G is structurally well-constrained if it is not structurally over-constrained and $DOC(G)=DOF(G)-3$.

1.1 Clipping and reducing

If there is a vertex $v\in V$ in $G=\{V,E\}$ such that $DOC(v)\leqslant DOF(v)$, then G can be solved by two steps: (1) solve the vertices except v; (2) solve v. Such a vertex v is called a removable vertex, and the related operation to remove v and its incident edges from G is called clipping, denoted by $G\to^{C(v)}G(v)$, where $G(v)=G-\{v\}-R(v)$, $R(v)$ is the edge set incident to v.

After a clipping operation we have

$$DOF(G(v))=DOF(G)-DOF(v)$$
$$DOC(G(v))=DOC(G)-DOC(v)$$

If there is a sub-graph $G_r\subseteq G$ such that G_r is structurally well-constrained then G_r can be reduced to a single vertex r called cluster. The vertices in G_r correspond to a set of geometric elements that can form a rigid body with three DOFs (two translational and one rotational). The combination operation is called reducing.

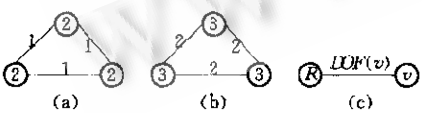

Fig. 1 Three graph patterns for reducing

There are too many graph patterns that correspond to rigid bodies and it is impossible to check them all. Figure 1 shows three graph patterns for reducing, which are similar to the work of Lee and Kim[6].

In pattern (a) the vertices correspond to geometric elements with 2 DOFs. In pattern (b) the vertices are reduced clusters or circles. In pattern (c) the vertex R must be a cluster, and the vertex v can be a geometric element or a cluster, and the sum of valencies of edges between them must be $DOF(v)$. Note that the constraints (edges) in Fig. 1 are all local. The graph reducing operations based on patterns (a), (b) and (c) are denoted by $\to^{R(a)}$, $\to^{R(b)}$ and $\to^{R(c)}$ respectively. Generally a graph reducing operation is noted as $\to^{R}$. Let G_r be a reducing pattern in G and $G \rightarrow^R G'$, and we have

$$DOF(G') = DOF(G) - DOF(G_r) + DOF(r)$$

$$DOC(G') = DOC(G) - DOC(G_r)$$

By integrating the two operations: clipping and reducing, we obtain a graph based decomposing algorithm: C-R algorithm.

1.2 Clipping-reducing algorithm (C-R algorithm)

Let $G=\{V,E\}$ be a constraint graph. The basic idea of clipping-reducing algorithm is as follows:

1. Identify removable vertices in G and remove them one by one until there is no removable vertex in G.

2. Let G' be the constraint graph after step 2. If $G'=\varnothing$ then end. Otherwise, search the reducing pattern (a) or pattern (b) using depth first search algorithm in G'[5]. If there is no reducing pattern (a) or (b) then end, else reduce the matched pattern into a cluster r.

3. Take the cluster vertex r as a seed and search the reducing pattern (c) in G'. If such a reducing pattern (c) is found, then reduce it in a new cluster r'. Let $r \leftarrow r'$ and extend the cluster r continuously until there is no such a reducing pattern (c).

4. Suppose G'' is the constraint graph after step 4. Let $G \leftarrow G''$ and go to step 1.

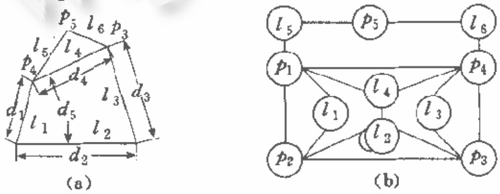

Fig. 2 A design system and its corresponding constraint graph

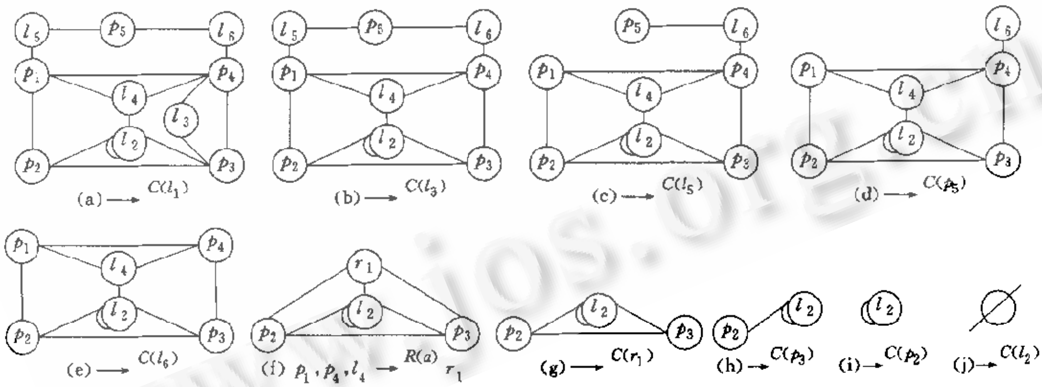

Fig. 3 The decomposing process through C-R approach

By employing the C-R algorithm in the decomposition of geometric constraint systems, we present a new graph based decomposing approach: C-R approach. Figure 2 gives a design system and its corresponding constraint graph. In Figure 2 there are five points and six lines with a set of constraints: $dist\_pp(p_1,p_2,d_1)$, $dist\_pp(p_2,p_3,d_2)$, $dist\_pp(p_3,p_4,d_3)$, $dist\_pp(p_4,p_1,d_4)$, $angle\_ll(l_2,l_4,d_5)$, $hor\_l(l_2)$ and ten incidence relations. The constraint $hor\_l(l_2)$ is a global constraint, while the other constraints are local ones. Figure 3 is a decomposing process of the design system through C-R approach.

After a series of decomposing operations, the original constraint graph becomes $\varnothing$. This illustrates that the constraint graph is decomposed completely by the C-R approach.

2 Correctness of C-R Approach

This section shows the correctness of C-R approach: (1) this approach can deal with the decomposition of structurally under-constrained systems, (2) this approach can detect structurally over-constrained systems, (3) the approach can terminate within finite number of steps, and (4) the solving steps obtained by using this approach are structurally consistent.

Definition 8. Given m independent equations with n variables, if $m=n$ then we say structurally the n variables can be solved by the m equations, or the solving process is structurally consistent.

Definition 9. A clipping operation or a reducing operation is generally named as a decomposing operation noted as $\rightarrow$. Every decomposing operation corresponds to a solving step in the solving sequence.

Definition 10. The sequences of clipping operations, reducing operations and decomposing operations are denoted by $\rightarrow^{C*}$, $\rightarrow^{R*}$ and $\rightarrow^{*}$ respectively.

Theorem 1. If a constraint graph G is not structurally over-constrained and there is a vertex $v \in V$ such that $DOC(v) < DOF(v)$, then G is structurally under-constrained.

Proof. Let $G(v) = G - \{v\} - R(v)$. If $G(v)$ is structurally under-constrained then G is also structurally under-constrained. Otherwise, suppose $G(v)$ is structurally well-constrained. From $G = G(v) + \{v\} + R(v)$ we have

$$DOC(G) = DOC(G(v)) + DOC(v)$$

$$DOF(G) = DOF(G(v)) + DOF(v)$$

Since $DOC(G(v)) = DOF(G(v)) - 3$ and $DOC(v) < DOF(v)$, then

$$DOC(G) = DOF(G(v)) - 3 + DOC(v) < DOF(G(v)) + DOF(v) - 3 = DOF(G)3$$

Therefore, G is structurally under-constrained. □

From Theorem 1 and the definition of clipping operation we immediately get the following corollary:

Corollary 1. An under-constrained graph can be decomposed by using clipping operations, in case that there exist removable vertices in the constraint graph.

Theorem 2. If there exist two vertices $v_1, v_2 \in G = \{V, E\}$ such that the sum of valencies of the edges between the two vertices is larger than $DOF(v_1)$ or $DOF(v_2)$, then the design system is structurally over-constrained.

Proof. The sum of valencies of the edges between v_1 and v_2 is denoted by $DOC(v_1, v_2)$, then we have $DOC(v_1, v_2) > DOF(v_1)$ or $DOC(v_1, v_2) > DOF(v_2)$. Consider the sub-graph $G_s = \{V_s, E_s\} \subseteq G$, where $V_s = \{v_1, v_2\}$ and E_s is the edges between v_1 and v_2. Then we have

$$DOF(G_s) = DOF(v_1) + DOF(v_2)$$

$$DOC(G_s) = DOC(v_1, v_2)$$

Since vertices v_1 and v_2 correspond to geometric elements or rigid bodies, we know that $DOF(v_1) \leqslant 3$ and $DOF(v_2) \leqslant 3$. Suppose $DOC(v_1, v_2) > DOF(v_1)$ without loss of generality, then

$$DOF(G_s) - DOC(G_s) = DOF(v_1) + DOF(v_2) - DOC(v_1, v_2) < DOF(v_2) \leqslant 3$$

Therefore, $DOC(G_s) > DOF(G_s) - 3$, and thus the constraint graph G_s is structurally over-constrained. Since G_s is a sub-graph of G, G is structurally over-constrained. □

When searching for removable vertices or pattern (c) sub-graphs in a constraint graph, if we find a vertex v such that $DOC(v) > DOF(v)$ then the constraint graph is structurally over-constrained according to Theorem 2.

Theorem 3. The decomposing process of the C-R approach terminates in finite steps.

Proof. After each clipping operation the number of vertices of constraint graph decreases by 1, while after each reducing operation the number of vertices of constraint graph decreases by no less than one. Hence, the decomposing process must terminate either when the constraint graph can not be decomposed any more or the constraint graph becomes $\varnothing$. □

Lemma 1. In a solving step, if the number of design variables n is no more than the sum of valencies of constraints m, then the solving step is structurally consistent.

Proof. Mathematically a solving step can be transformed into a set of m independent equations with n variables. If $m=n$ then the solving process is structurally consistent, otherwise if $m<n$ we can assign $n-m$ variables randomly. The remainder m variables can be solved by the m equations. Thus if $m\leqslant n$ then structurally n variables can be solved by the m nonlinear equations, or the solving step is structurally consistent. □

Lemma 2. The solving step corresponding to a clipping operation is structurally consistent.

Proof. In the solving step corresponding to a clipping operation, the number of variables is the DOF of the removed vertex v and the number of nonlinear equations is the DOC of v. Since $DOC(v)\leqslant DOF(v)$, then the solving step is structurally consistent. □

Lemma 3. The solving step corresponding to a reducing operation is structurally consistent.

Proof. Suppose G_r is the sub-graph matching one of three patterns: pattern (a), patterns (b) and pattern (c). For pattern (a) and (b) the number of variables is $DOF(G_r)$ and the number of nonlinear equations is $DOC(G_r)$. Since G_r is a structurally well-constrained graph, $DOC(G_r)=DOF(G_r)-3$. Thus the solving step corresponding to $\rightarrow^{R(a)}$ or $\rightarrow^{R(b)}$ is structurally consistent. For pattern (c) the number of variables is the DOF of the extended vertex v and the number of nonlinear equations is the sum of valencies of the edges incident to v in G_r. Since these two numbers are equal, the solving step corresponding to $\rightarrow^{R(c)}$ is also structurally consistent. □

Lemma 4. If G is a structurally not over-constrained and $G\rightarrow^{*} G_f$, then G_f is structurally not over-constrained.

Proof. The proof is done by induction on the length of the decomposing sequence that derives G_f from G. The induction basis is $G_f=G$ and is trivial. For the induction step, consider the last decomposing step, supposing the current constraint graph G' is structurally not over-constrained. There are two types of decomposing operation:

- The last decomposing step is clipping operation. Now G_f is a sub-graph of G'. Since G' is not structurally over-constrained, from the definition of structurally over-constrained graph we know that G_f is also structurally not over-constrained.

- The last decomposing step is reducing operation. Suppose G_r is the sub-graph which matches a reducing pattern and r is the cluster reduced by the last reducing operation, then

$$DOF(G_f)=DOF(G')-DOF(G_r)+DOF(r)$$
$$DOC(G_f)=DOC(G')-DOC(G_r)$$

Since $DOC(G')\leqslant DOC(G')-3$, $DOC(G_r)=DOF(G_r)-3$ and $DOF(r)=3$, we have

$$\begin{aligned}DOC(G_f)&\leqslant DOF(G')-3-DOC(G_r)=DOF(G_f)+DOF(G_r)-DOF(r)-DOC(G_r)-3\\&=DOF(G_f)-DOF(r)=DOF(G_f)-3\end{aligned}$$

Therefore, if $G\rightarrow^{*} G_f$ then G_f is structurally not over-constrained. □

From Lemma 1, Lemma 2 and Lemma 3 and Lemma 4 we immediate obtain:

Corollary 2. The solving sequence obtained by using C-R approach is structurally consistent.

3 Conclusion

Previous graph constructive approaches can not deal with the decomposition of under-constrained design systems effectively. By applying the two operations: clipping and reducing alternately, the C-R decomposing approach can deal with the decomposition of under-, well- and over-constrained design systems. This paper proves the correctness of the C-R decomposing approach and shows that the C-R decomposing approach can generate a structurally consistent solving sequence or detect the structurally over-constrained problems. However, none of the decomposing approaches can guarantee the numerical consistency of the solving sequence or detect the numerical inconsistency in design systems. In the second phase of constraint solving process: execution phase, the position and shape of geometric elements are calculated sequentially and the numerical inconsistency is also detected along with the solving sequence.

References:

[1] Sridhar, N., Agrawal, R., Kinzel, G.L. Algorithms for the structural diagnosis and decomposition of sparse, underconstrained design systems. Computer-Aided Design, 1996,28(4):237~249.

[2] Bouma, W., Fudos, I., Hoffmann, C.M., *et al*. Geometric constraint solver. Computer-Aided Design, 1995,27(6):487~501.

[3] Owen, J.C. Algebraic solution for geometry from dimensional constraints. In: Proceedings of the 1st Symposium on Solid Modeling Foundations and CAD/CAM Applications. New York: ACM Press, 1991. 379~407.

[4] Kramer, G.A. Solving Geometric Constraint Systems: A Case Study in Linematics. Cambridge, MA: MIT Press, 1992.

[5] Fudos, I., Hoffmann, C.M. A graph-constructive approach to solving systems of geometric constraints. ACM Transactions on Graphics, 1997,16(2):179~216.

[6] Owen, J.C. Algebraic solution for geometry from dimensional constraints. In: Jaroslaw R. ed. Proceedings of the 1st Symposium on Solid Modeling Foundations and CAD/CAM Applications. New York: ACM Press, 1991. 379~407.

一种设计分解的正确性证明

袁 波, 李彦涛, 孙家广

(清华大学 计算机科学与技术系,北京 100084)

摘要: 二维变量化设计系统可以用含有 n 个未知数、m 个方程的非线性方程组表示.通过设计分解可以提高几何约束求解的效率和数值稳定性.给出了一种基于图论的设计分解方法及其正确性证明.该方法可以(1)处理结构欠约束系统的分解;(2)检测出冗余约束.分解算法在有限步内终止,其结果是结构相容的.

关键词: 设计分解;几何约束求解;剪枝;规约

中图法分类号: TP391 **文献标识码**: A