

Tipo de artículo: Artículo original
Temática: Soluciones Informáticas
Recibido: 8/05/18 | Aceptado: 5/7/18 | Publicado: 27/07/18

Método de Descomposición-Recombinación a emplear en una aplicación CAD

Decomposition-Recombination method to be used in a CAD application

Gustavo García González¹, Augusto César Rodríguez Medina²

¹Universidad de las Ciencias Informáticas, ggarcia@uci.cu

²Universidad de las Ciencias Informáticas, augusto@uci.cu

* Autor para correspondencia: ggarcia@uci.cu

Resumen

En el presente trabajo se enuncian una serie de algoritmos para la descomposición-recombinación de los modelos geométricos basados en restricciones que maneja una aplicación CAD, con el objetivo de identificar el adecuado, teniendo en cuenta la eficiencia, completitud y aplicabilidad en dos y tres dimensiones. El algoritmo identificado se aplicará a una aplicación CAD de desarrollo nacional que posee como premisa brindar una herramienta que asista a los procesos de diseño en la industria cubana, empleando herramientas y tecnologías de código abierto, para garantizar la soberanía tecnológica y eliminar las limitaciones que supone el uso de herramientas propietarias para tales fines.

Palabras clave: completitud, descomposición, eficiencia, modelo, recombinación, restricciones.

Abstract

In this paper, a series of algorithms for the decomposition-recombination of the constraint-based geometric models handled by a CAD application are presented, with the aim of identifying the appropriate algorithm, taking into account the efficiency, completeness and applicability in two and three dimensions. The algorithm identified will be applied to a national development CAD system application that has the premise of providing a tool to assist design processes in the Cuban industry, using open source tools and technologies to guarantee technological sovereignty and eliminate the limitations imposed by the use of proprietary tools for such purposes.

Keywords: completeness, constraints, decomposition, efficiency, model, recombination.

Introducción

El modelado por restricciones permite a los usuarios describir formas especificando relaciones entre los elementos geométricos. Un *solver* de restricciones deriva entonces automáticamente el diseño pretendido explotando estas restricciones.

La evolución de los modelos geométricos basados en restricciones está estrechamente ligada a los sistemas de diseño asistido por ordenador (CAD, por sus siglas en inglés) paramétricos y basados en características. Desde la introducción del diseño paramétrico por parte de Pro / Engineer en la década de 1980, la mayoría de los principales sistemas CAD adoptaron modelos geométricos basados en restricciones como tecnología principal. Los modelos geométricos basados en restricciones permitieron a los sistemas CAD proporcionar un modelo de datos más potente al tiempo que ofrecen una interfaz de usuario intuitiva. Posteriormente, también se encontraron aplicaciones a campos como el diseño de vinculación, modelado químico, visión por computadora y geometría dinámica.[1]

Hoffmann et al., [2], plantea las propiedades deseables de los *solvers* incluyen eficiencia, simplicidad, riqueza de dominio y solidez.

Los sistemas CAD contemporáneos emplean una gran cantidad de elementos y restricciones geométricas, de tal manera que se hace primordial el estudio de estrategias destinadas a aumentar la eficiencia, denominadas Descomposición-Recombinación(DR).

Entre los sistemas CAD contemporáneos se encuentran AutoDesk Incorporated, Dassault Systèmes, el Solid Edge, FreeCAD, LibreCAD, entre otros. La gran mayoría de estos sistemas se encuentran bajo licencias propietarias, por lo que debido al bloqueo impuesto sobre Cuba por parte de los Estados Unidos de América, se hace imposible la compra y uso de estas tecnologías en la industria cubana, trayendo como efecto que no se pueda comercializar planos de proyectos que se hayan realizado con estas herramientas debido a posibles sanciones y litigios.

Teniendo en cuenta la limitante antes mencionada en la Facultad 4 de la Universidad de las Ciencias Informáticas, se encuentra el desarrollo de una herramienta CAD para la industria nacional, la cual cuenta con un *solver* de restricciones en 2D poco eficiente y carece de uno en 3D. De tal manera que se ha identificado la necesidad de una

investigación sobre las estrategias algorítmicas a seguir para obtener un sistema de restricciones geométricas eficiente en 2D y 3D.

Materiales y métodos o Metodología computacional

Con el fin de caracterizar los *solver* existentes en las aplicaciones comerciales actuales, se ha observado y analizado los tiempos de ejecución con un conjunto de objetos y restricciones geométricas sobre ellos. Este análisis permitió determinar la alta capacidad de algunos sistemas CAD de resolver sistemas complejos.

Se ha realizado una investigación de 14 artículos publicados por la comunidad científica referentes al tema en cuestión, que recorren desde los primeros pasos en esta esfera en los 90s hasta la actualidad, los cuales permitieron una caracterización y valoración de los algoritmos existentes.

Teniendo en cuenta las tecnologías empleadas en el sistema CAD en desarrollo, al cual se le incorporará los resultados de la presente investigación, se han identificado como tecnologías a emplear el lenguaje de programación C++11, el framework de desarrollo Qt5.5 y la biblioteca OpenCascade Community Edition 0.17.

Resultados y discusión

Los *solvers* de los sistemas CAD contemporáneos son basados en la construcción de grafos que relacionan las restricciones con los elementos geométricos. Estos *solvers* deben ser capaces de manejar un gran volumen de elementos y restricciones geométricas, por lo que se utiliza la estrategia “divide y vencerás” para su solución. Esta estrategia aplicada a los sistemas de restricciones geométricas se denomina Descomposición-Recombinación, la cual consiste en tres pasos básicos, véase Hoffmann et al., [3], Ait-Aoudia, et al., [4], que lo resumen como:

1. Dividir el problema P en subproblemas P_1, P_2, \dots, P_n
2. Resolver cada subproblema P_i de forma recursiva con algoritmos conocidos
3. Construir una solución de P mediante la fusión de soluciones de los subproblemas P_1, P_2, \dots, P_n

Los *solvers* de restricciones que se ajustan a esta descripción se conocen como solucionadores de descomposición-recombinación (*solvers* de DR).

Hasta ahora, casi todas las técnicas GCS (*Geometric Constraint Solving*, en español Resolución de Restricciones

Geométricas) basadas en grafos se ocupan de las descomposiciones que comienzan con un sistema de restricción global, por ejemplo, en los años noventa, el método *Triangle-decomposition* (TD) de Owen [5], el método de reducción-descomposición de Hoffmann et al. [6], el método de descomposición de S-Tree de Joan-Arinyo et al. [7], así como en los años 2000, Lee et al. plantea un método de clasificación de regla [8]. Todos consideraron los triángulos como patrones básicos para descomponer sistemas de restricción 2D, con alta eficiencia y robustez.[9]

Los métodos anteriores limitan la descomposición del sistema y las intenciones de diseño en gran medida, debido a la diversidad y la complejidad de sus estructuras de diseño. A este respecto, los investigadores han propuesto algunos métodos de descomposición más generales, tales como, en los años noventa, el método de secuencias de construcción general [10] propuesto por Latham y Middleditch (LMA), en la década de 2000, el Algoritmo Frontera Modificado de Hoffmann et al. (MFA) [11, 12], Gao et al., Árbol de conectividad(C-Tree) [13], así como en los 2010s, el *Skeleton Decomposition-recombination* (S-DR) propuesto por Samy et al.[14]

El algoritmo propuesto por Gao et al. [13] posee una complejidad $O(n^2(n+e)e)$, al igual que el MFA de Hoffmann et al. [11, 12], pero el primero es más sencillo de comprender e implementar. El S-DR [13] es un algoritmo solamente aplicable a sistemas en 2D, por lo que sus uso es limitado a la hora de su empleo en 3D.

Sun et al. [9] plantean un nuevo algoritmo de descomposición-recombinación general para dos y tres dimensiones al cual denominaron ILMA, el cual consiste en aplicar el LMA iterativamente. Este algoritmo mejora considerablemente la eficiencia de la descomposición manteniendo la premisa de la robustez del *solver*. El ILMA es completo y conlleva una implementación eficiente, pues posee una complejidad algorítmica de $O(n(3n+e)(e-n))$, mejorando así la de algoritmos actuales como el C-Tree y el MFA.[9]

Teniendo en cuenta la completitud y mejora considerable de la complejidad algorítmica del ILMA con respecto a los otros algoritmos existentes en la literatura y su aplicabilidad en dos y tres dimensiones, se identificó como el algoritmo de descomposición-recombinación más adecuado a aplicar en aplicaciones CAD que trabajen con una gran cantidad de elementos y restricciones geométricas.

Conclusiones

La investigación realizada arrojó que:

- Los algoritmos TD, S-Tree, entre otros, que consideran a los triángulos como patrones básicos, limitan la descomposición del sistema y las intenciones de diseño en gran medida, debido a la diversidad y la complejidad de sus estructuras de diseño.
- Se han propuesto algunos métodos de descomposición-recombinación más generales que los antes mencionados, como, el LMA, C-Tree, MFA, S-DR e ILMA. Entre estos se recomienda en la presente investigación el uso del ILMA en aplicaciones CAD actuales, pues es completo, más eficiente en cuanto a complejidad algorítmica que los otros métodos abordados y es aplicable en dos y tres dimensiones.

Referencias

Tarrés-Puertas, Marta I. Direct Tree Decomposition of Geometric Constraint Graphs, 2014

Hoffman, C. M., Lomonosov, A., and Sitharam, M. Decomposition plans for geometric constraint systems, part i: Performance measures for CAD. *Journal of Symbolic Computation*, 2001

Gao, X.-S., Hoffmann, C. M., and Yang, W.-Q. Solving spatial basic geometric constraint configurations with locus intersection, 2002

Ait-Aoudia, S., and Foufou, S. A 2D geometric constraint solver using a graph reduction method. *Advances in Engineering Software*, 2010

J. C. Owen, Algebraic solution for geometry from dimensional constraints. *ACM Symposium Foundation of Solid Modeling*, 1991.

I. Fudos, C. M. Hoffmann, A graph-constructive approach to solving systems of geometric constraints. *ACM Trans. Graph.*, 1997

R. Joan-Arinyo, A. Soto-Riera, S. Vila-Marta, J. Vilaplana-Pasto, Revisiting decomposition analysis of geometric constraint graphs, *Computer-Aided Design*, 1992

Kyu-Yeul Lee, O-Hwan Kwon, Jae-Yeol Lee, Tae-Wan Kim, A hybrid approach to geometric constraint solving with graph analysis and reduction. *Advances in Engineering Software*, 2003

Wenhui Li, Mingyu Sun, Chun Han, Baoliang Mu. A New Incremental Decomposition Technique for 2D and 3D Geometric Constraint Systems. *Journal of Information & Computational Science*, 2015

R. S. Latham, A. E. Middleditch, Connectivity analysis: A tool for processing geometric constraints, *Computer-Aided Design*, 1996

- C. M. Hoffmann, A. Lomonosov, M. Sitharam, Decomposition plans for geometric constraint systems, Part I: Performance measures for CAD, *J. Symbolic Comput.*, 2001.
- C. M. Hoffmann, A. Lomonosov, M. Sitharam, Decomposition plans for geometric constraint systems, Part II: New algorithms, *J. Symbolic Comput.*, 2001
- X. S. Gao, Q. Lin, G. F. Zhang, A C-tree decomposition algorithm for 2D and 3D geometric constraint solving, *Computer-Aided Design*, 2006
- Samy Ait-Aoudia, Sebti Foufou. A 2D geometric constraint solver using a graph reduction method, 2010