Computational Geometry 47 (2014) 75–89

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Automatic deduction in (dynamic) geometry: Loci computation

Francisco Botana [a], Miguel A. Abánades [b,*]

[a] Departamento de Matemática Aplicada I, Universidad de Vigo, Campus A Xunqueira, 36005 Pontevedra, Spain
[b] CES Felipe II, Universidad Complutense de Madrid, 28300 Aranjuez, Spain

ARTICLE INFO

Article history:
Received 10 July 2012
Received in revised form 1 February 2013
Accepted 9 July 2013
Available online 16 July 2013
Communicated by M. Teillaud

Keywords:
Automatic deduction in geometry
Locus
Sage
GeoGebra
Gröbner cover

ABSTRACT

A symbolic tool based on open source software that provides robust algebraic methods to handle automatic deduction tasks for a dynamic geometry construction is presented. The prototype has been developed as two different worksheets for the open source computer algebra system *Sage*, corresponding to two different ways of coding a geometric construction, namely with the open source dynamic geometry system *GeoGebra* or using the common file format for dynamic geometry developed by the *Intergeo* project. Locus computation algorithms based on Automatic Deduction techniques are recalled and presented as basic for an efficient treatment of advanced methods in dynamic geometry. Moreover, an algorithm to eliminate extraneous parts in symbolically computed loci is discussed. The algorithm, based on a recent work on the Gröbner cover of parametric systems, identifies degenerate components and extraneous adherence points in loci, both natural byproducts of general polynomial algebraic methods. Several examples are shown in detail.

1. Introduction

Automated Deduction in Geometry, understood as the study and development of computer programs designed to prove geometry theorems, can be traced back almost five decades to the influential work of Gelernter in the field of Artificial Intelligence [1]. However, the real flourishing of the field came in the early 1980's with the development by Wu of an algebraic method based on Ritt's characteristic set for proving a restricted set of geometry theorems [2]. Impressive results by several authors using Wu's method (e.g. [3–6]) encouraged researchers to consider other algebraic methods, among which those based on Gröbner bases [7] proved to be the most relevant (see [8–10]).

Directly related to automatic proving is the concept of automatic *discovery*, more closely related to the work presented in this article. While automatic proving deals with verifying geometric statements, automatic discovery tries to find complementary hypotheses for statements to become true or, as stated in [11], to "finding the missing hypotheses so that a given conclusion follows from a given incomplete set of hypotheses". A systematic use of algebraic methods based on Gröbner bases to address the question of automatic discovery in geometry was thoroughly developed in [12].

From the appearance of the very first prototypes of automatic provers it was clear the need to develop accompanying graphic interfaces. Almost at the same time appeared the first stable dynamic geometry systems (DGS) with great acceptance, specially in the field of education. Let us recall that a DGS is a computer application that allows the exact on-screen drawing of geometric diagrams and their interactive manipulation by mouse dragging. Also, computer algebra systems (CAS), whose core functionality is the manipulation of mathematical expressions in symbolic form, reached a very high level of

* Corresponding author.
E-mail addresses: fbotana@uvigo.es (F. Botana), abanades@ajz.ucm.es (M.A. Abánades).

0925-7721/$ – see front matter
http://dx.doi.org/10.1016/j.comgeo.2013.07.001

development and started becoming available in most teaching and research institutions. Several authors since then have postulated a deeper cooperation between DGS and CAS in order to enhance the abilities of dynamic geometry programs with algebraic proving and discovering algorithms. In particular, in [12] one can read that their method can be regarded as "the core of a future program [...] that allows, when linked simultaneously with a tool for displaying geometric constructions and a symbolic computation package, the interactive exploration of geometric properties".

Considerable attention and efforts have been dedicated since then to these emerging new tools which can be termed symbolic – dynamic geometry environments, a synthesis of DGS and CAS. In particular, two main approaches have been followed. Some DGS incorporate their own code to perform symbolic computations (e.g. [13]), while others choose to reuse existing Computer Algebra Systems (e.g. [14–17]).

In this article we present an efficient solution in this second direction strictly based on open source software applications. In particular, the developed prototype symbolically computes equations of geometric loci and determines the validity of geometric statements specified in an open source DGS. Moreover, an extension of the prototype has been developed that accepts, as input, the description of a geometric locus specified using a common DGS file format developed in a European project and accepted by most European DGS. Finally, complementing the symbolic computation of geometric loci, a tool to detect degenerate components as well as extraneous adherence points in the returned loci has been developed based on the recent Gröbner Cover algorithm for solving parametric systems in [18]. It is illustrated with several examples of loci and envelopes. To the best of our knowledge, the described approach is the first to implement in a completely automatic way (i.e. with no user interaction at all) the removal of extraneous parts (degenerate components in particular) from objects computationally generated in a DGS. We feel this is an important step in the solution of one of the main current bottlenecks in the development of dynamic geometry environments. We are confident that this method will be integrated in coming versions of these environments.

Following are a few words on the main building blocks of our system, namely the open source applications GeoGebra and Sage and the DGS common file format Intergeo.

1.1. GeoGebra

GeoGebra is an open source DGS with algebraic capabilities, establishing a direct relationship between the objects in the different windows: graphics, algebra, and spreadsheet. It was created in 2001 by Markus Hohenwarter as part of his Master in Mathematics Education at the University of Salzburg, Austria, winning him the 2002 European Academic Software Award (EASA) in the category of Mathematics. Since then, the tool, using the word of mouth and Internet, spread rapidly throughout the world, and has become a collaborative project with impressive figures: users in 190 countries, versions in 62 languages and almost a million visitors to its web site every month.

It is worth mentioning the ongoing creation of a network of International GeoGebra Institutes (IGI) that serves as a platform from which teachers and researchers from around the world work together to promote activities and research related to GeoGebra and its applications. Currently there are 180 official IGIs in 80 countries.

1.2. Sage

Sage (http://sagemath.org) is a free open source CAS designed to be a viable multi-platform free open source alternative to proprietary – and expensive – systems such as Mathematica, Maple or Matlab. Besides being open source, the integration of multiple tools and the possibility of remote access via the Internet make their most notable features.

Built out of nearly 100 open source packages (including Singular, Axiom, Maxima, ...), Sage features a unified interface that takes the form of a notebook in a web browser or the command line. Using the notebook, Sage connects either locally to your own Sage installation or to a Sage server on the network. Inside the Sage notebook you can create embedded graphics, beautifully typeset mathematical expressions, add and delete input and share your work.

Sage was created in 2004 by William Stein (http://wstein.org), professor at the University of Washington, motivated, among other things, by several disagreements over some accessibility issues with the developers of Magma, a highly specialized commercial CAS in whose development he had collaborated.

1.3. Intergeo

The *Intergeo* (i2g) file format is an XML-based specification designed to describe any construction created with a DGS. It is based on OpenMath [19], a standard for representing mathematical objects with their semantics and was developed as part of the Intergeo project (http://i2geo.net), an eContentplus European project in which the authors took part dedicated to the sharing of interactive geometry constructions across systems.

An i2g file takes the form of a compress file package. The file intergeo.xml provides a textual description of the construction in two parts, one with the elements of the construction that describes the (static) initial configuration and one where the geometric constraints among the elements are included. A detailed specification of the file format can be found at http://i2geo.net/xwiki/bin/view/I2GFormat/, where several examples are provided.