Graphical Models 82 (2015) 99–109

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Geometric operations using sparse interpolation matrices

Ioannis Z. Emiris, Tatjana Kalinka, Christos Konaxis*

Dept Informatics and Telecommunications, University of Athens, Greece

ARTICLE INFO

Article history:
Received 10 November 2014
Revised 16 May 2015
Accepted 10 June 2015
Available online 23 June 2015

Keywords:
Matrix representation
Implicitization
Linear algebra
Sparse polynomial
Membership
Sidedness operation

ABSTRACT

Based on the computation of a superset of the implicit support, implicitization of a parametrically given hypersurface is reduced to computing the nullspace of a numeric matrix. Our approach predicts the Newton polytope of the implicit equation by exploiting the sparseness of the given parametric equations and of the implicit polynomial, without being affected by the presence of any base points. In this work, we study how this interpolation matrix expresses the implicit equation as a matrix determinant, which is useful for certain operations such as ray shooting, and how it can be used to reduce some key geometric predicates on the hypersurface, namely membership and sidedness for given query points, to simple numerical operations on the matrix, without need to develop the implicit equation. We illustrate our results with examples based on our Maple implementation.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

A fundamental question in changing representation of geometric objects is implicitization, namely the process of changing the representation of a geometric object from parametric to implicit. It is a fundamental operation with several applications in computer-aided geometric design (CAGD) and geometric modeling. There have been numerous approaches for implicitization, including resultants, Gröbner bases, moving lines and surfaces, and interpolation techniques.

In this work, we restrict attention to hypersurfaces and exploit a matrix representation of hypersurfaces in order to perform certain critical operations efficiently, without developing the actual implicit equation. Our approach is based on potentially interpolating the unknown coefficients of the implicit polynomial, but our algorithms shall avoid actually computing these coefficients. The basis of this approach is a sparse interpolation matrix, sparse in the sense that it is constructed when one is given a superset of the implicit polynomial's monomials. The latter is computed by means of sparse resultant theory, so as to exploit the input and output sparseness, in other words, the structure of the parametric equations as well as the implicit polynomial.

The notion that formalizes sparseness is the support of a polynomial and its Newton polytope. Consider a polynomial f with real coefficients in n variables $t_1, \dots, t_n$, denoted by

$$f = \sum_a c_a t^a \in \mathbb{R}[t_1, \dots, t_n],\, a \in \mathbb{N}^n,\, c_a \in \mathbb{R},$$
$$\text{where } t^a = t_1^{a_1} \cdots t_n^{a_n}.$$

The *support* of f is the set $\{a \in \mathbb{N}^n : c_a \neq 0\}$; its *Newton polytope* $N(f) \subset \mathbb{R}^n$ is the convex hull of its support. All concepts extend to the case of Laurent polynomials, i.e. with integer exponent vectors $a \in \mathbb{Z}^n$.

The main ingredient of our method is the Newton polytope of the implicit equation, or *implicit polytope* and the set of lattice points it contains, which we call *implicit support*. The vertices of the implicit polytope are called *implicit vertices*. The implicit polytope is computed from the Newton polytope of the sparse (or toric) resultant, or *resultant polytope*, of auxiliary polynomials defined by the parametric equations. Under certain genericity assumptions, the implicit polytope coincides with a projection of the resultant polytope, see Section 3. In general, a translate of

* Corresponding author.
E-mail address: ckonaxis@di.uoa.gr (C. Konaxis).

http://dx.doi.org/10.1016/j.gmod.2015.06.007
1524-0703/© 2015 Elsevier Inc. All rights reserved.

the implicit polytope is contained in the projected resultant polytope, in other words, a superset of the implicit support is given by the lattice points contained in the projected resultant polytope, modulo the translation. A superset of the implicit support can also be obtained by other methods, see Section 2; the rest of our approach does not depend on the method used to compute this support.

The predicted support is used to build a numerical matrix whose kernel is, ideally, 1-dimensional thus yielding (up to a nonzero scalar multiple) the coefficients corresponding to the predicted implicit support. This is a standard case of *sparse interpolation* of the polynomial from its values. When dealing with hypersurfaces of high dimension, or when the support contains a large number of lattice points, then exact solving is expensive. Since the kernel can be computed numerically, our approach also yields an approximate sparse implicitization method.

Our method of sparse implicitization was developed in [12,13], see Section 2. It handles hypersurfaces given parametrically by polynomial, rational, or trigonometric parameterizations and, furthermore, automatically handles the case of base points. The standard version of the method requires to compute the monomial expansion of the implicit equation. However, it would be preferable if various operations and predicates on the hypersurface could be completed by using the matrix without developing the implicit polynomial. This is an area of strong current interest, since expanding, storing and manipulating the implicit equation can be very expensive, whereas the matrix offers compact storage and fast, linear algebra computations. This is precisely the premise of this work.

The main contribution of this work is to show that the matrix representation can be very useful when based on sparse interpolation matrices. First, from the interpolation matrix we construct a matrix which is numeric except for its last row. When this matrix is non-singular, its determinant equals the implicit equation (up to a constant multiple). This allows us to use the (nonzero) sign of the determinant of the numeric matrix obtained by evaluating its symbolic last row, to decide sidedness for query points q that do not lie on the hypersurface $p(x) = 0$. Second, we use the interpolation matrix, independently of its rank, to reduce the membership test $p(q) = 0$, for a query point q and a hypersurface defined implicitly by $p(x) = 0$, to rank tests on numeric matrices.

Moreover, we implement an alternative interpolation matrix using the linear relations between the implicit and parametric expressions of the normal to the curve or surface at any given point, see e.g., [4]. This method is also known as Hermite interpolation. The new matrix is not of smaller size, but the number of sample points is reduced. With curves/surfaces, our method requires about half/one third of the sample points, respectively. The matrices again can be used to numerically evaluate membership and sidedness.

Our methods work either with parameterized objects or with objects given by a point cloud along with normals at the points. When the parametric equations are not known, support prediction is not possible hence we use bounds on the implicit degree or try successively larger simplices. Our algorithms have been implemented in Maple. To emphasize algorithms and practical results we detail code and experiments; for readability we omit proofs of the statements. All omitted proofs can be found in [11].

The paper is organized as follows: Section 2 overviews previous work. Section 3 describes our approach to predicting the implicit support while exploiting sparseness, presents our implicitization algorithm based on computing a matrix kernel and focuses on the case of high dimensional kernels. In Section 4 we formulate membership and sidedness tests as numerical linear algebra operations on the interpolation matrix. Our Maple implementation is described in Section 5 along with some examples. We conclude with further work and open questions.

2. Previous work

This section overviews existing work.

If S is a superset of the implicit support, then the most direct method to reduce implicitization to linear algebra is to construct a $|S| \times |S|$ matrix M, indexed by monomials with exponents in S (columns) and $|S|$ different values (rows) at which all monomials get evaluated. Then the vector of coefficients of the implicit equation is in the kernel of M. This idea was used in [12,19,24]; it is also the starting point of this paper.

An interpolation approach was based on integrating matrix $M = SS^{\top}$, over each parameter $t_1, \ldots, t_n$ [6]. Then the vector of implicit coefficients is in the kernel of M. In fact, the authors propose to consider successively larger supports in order to capture sparseness. This method covers polynomial, rational, and trigonometric parameterizations, but the matrix entries take big values (e.g. up to 10^{28}), so it is difficult to control its numeric corank, i.e. the dimension of its nullspace. Thus, the accuracy of the approximate implicit polynomial is unsatisfactory. When it is computed over floating-point numbers, the implicit polynomial does not necessarily have integer coefficients. They discuss post-processing to yield integer relations among the coefficients, but only in small examples.

Our method of sparse implicitization was introduced in [12], where the overall algorithm was presented together with some results on its preliminary implementation, including the case of approximate sparse implicitization. The emphasis of that work was on sampling and oversampling the parametric object so as to create a numerically stable matrix, and examined evaluating the monomials at random integers, random complex numbers of modulus 1, and complex roots of unity. That paper also proposed ways to obtain a smaller implicit polytope by downscaling the original polytope when the corresponding kernel dimension was higher than one.

One issue was that the kernel of the matrix might be of high dimension, in which case the equation obtained may be a multiple of the implicit equation. In [13] this problem was addressed by describing the predicted polytope and showing that, if the kernel is not 1 dimensional, then the predicted polytope is the Minkowski sum of the implicit polytope and an extraneous polytope. The true implicit polynomial can be obtained by taking the greatest common divisor (GCD) of the polynomials corresponding to at least two and at most all of the kernel vectors, or via multivariate polynomial factoring.

Our implicitization method is based on the computation of the implicit polytope, given the Newton polytopes of the

Download English Version:

https://daneshyari.com/en/article/442375

Download Persian Version:

https://daneshyari.com/article/442375

Daneshyari.com