Computer Aided Geometric Design 60 (2018) 18–27

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Hermite mean value interpolation on polygons ☆

Rick Beatson [a,*], Michael S. Floater [b], Carl Emil Kåshagen [b]

[a] Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand
[b] Department of Mathematics, University of Oslo, Moltke Moes vei 35, 0851 Oslo, Norway

A R T I C L E I N F O

Article history:
Received 23 June 2017
Received in revised form 15 January 2018
Accepted 18 January 2018
Available online xxxx

Keywords:
Transfinite interpolation
Hermite interpolation
Mean value coordinates

A B S T R A C T

Hermite mean value interpolation is a method for interpolating function values and derivatives on the boundary of a domain, using boundary integrals. In this paper we specialize the interpolation to polygonal domains and show that if the boundary data is piecewise quadratic, the integrals can be found explicitly and evaluated easily.

1. Introduction

In Floater and Schulz (2008), a method was derived for constructing, in a domain $\Omega \subset \mathbb{R}^2$, a function $g : \Omega \to \mathbb{R}$ that matches the values and normal derivatives of some function f on the boundary $\partial\Omega$. The method comes from minimizing a kind of energy functional, similar to thin-plate energy, but which allows an explicit solution in terms of boundary integrals. Since the method is a natural extension of mean value interpolation to function values, we might call it Hermite mean value interpolation.

In analogy to the fact that mean value interpolation has linear precision, Hermite mean value interpolation has cubic precision, due to the energy functional being minimized. For this reason, it is not surprising that the method appears to perform very well in practice, generating visually smooth and well-behaved interpolants, even when the domain is irregularly shaped. In the current paper we specialize the method to polygonal domains and show that if the boundary data is piecewise quadratic, the integrals can be found explicitly and evaluated easily. Cubic precision for the continuous version of the method was proved in Floater and Schulz (2008). This implies quadratic precision for the discrete version with piecewise quadratic values, and piecewise linear first derivatives on the boundary.

In Li et al. (2013), the same method was derived, independently, based on a mean value property of biharmonic functions. The authors also specialized the method to polygons. They argued that the boundary integrals have a closed form if one makes g cubic along each edge, and the normal derivative of g linear. Due to the choice of normal derivative, the resulting method loses cubic precision but retains quadratic precision. They used the method for image deformation by applying it to vector-valued data. The result is a generalization of mean value coordinates, with more deformation parameters to tweak. Some nice examples of this were shown in Li et al. (2013).

On the other hand, the 'closed formulas' for the integrals in Li et al. (2013) are by no means simple to derive, nor to implement, and involve inverse trigonometric, log, and complex-valued functions. The goal of this paper is to propose a

☆ This paper has been recommended for acceptance by Kai Hormann.

* Corresponding author.
E-mail addresses: rick.beatson@canterbury.ac.nz (R. Beatson), michaelf@math.uio.no (M.S. Floater), carlek@student.matnat.uio.no (C.E. Kåshagen).

https://doi.org/10.1016/j.cagd.2018.01.002
0167-8396/

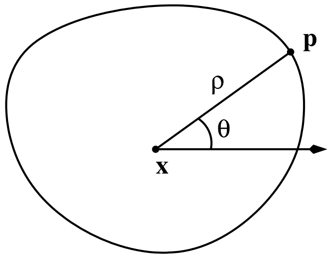

Fig. 1. Definition of $\mathbf{p}$ and ρ.

considerable simplification of the integrals by taking g to be piecewise quadratic along each edge, and the normal derivative of g piecewise linear. The resulting method still has quadratic precision, similar to Li et al. (2013), but the integrals can be computed using only square roots in addition to the usual arithmetic operations. We also focus on the idea of using the interpolation to generate n-sided patches for computer aided design, and compare it with the 'ribbon'-interpolants developed by Várady et al. (2011).

2. Hermite interpolation on convex domains

We start by recalling the construction of Floater and Schulz (2008), specialized to planar domains, and for value and first derivative data only. We will consider both convex and non-convex domains, but it is simpler to explain the mathematical definitions for convex domains. The extension to non-convex ones is very simple, and parallels the generalization of mean value interpolation from convex domains to non-convex domains as in Hormann and Floater (2006) and Dyken and Floater (2009). So, let $\Omega \subset \mathbb{R}^2$ be a bounded, open, convex domain. Given the values and first order partial derivatives of a function f on the boundary $\partial\Omega$, the approach to constructing an interpolant $g : \Omega \to \mathbb{R}$ in Floater and Schulz (2008) was as follows. First, for all boundary points $\mathbf{p} \in \partial\Omega$, we set $g(\mathbf{p}) = f(\mathbf{p})$ and $D_{\mathbf{n}} g(\mathbf{p}) = D_{\mathbf{n}} f(\mathbf{p})$, where $D_{\mathbf{n}}$ denotes the normal derivative in the outward unit normal direction $\mathbf{n}$. Then, we determine $g(\mathbf{x})$ for each point $\mathbf{x} = (x_1, x_2)$ in Ω. With $\mathbf{x} \in \Omega$ fixed, let τ be a linear polynomial in $\mathbb{R}^2$, which we can represent as

$$\tau(\mathbf{y}) = a + \mathbf{b} \cdot (\mathbf{y} - \mathbf{x}), \qquad \mathbf{y} \in \mathbb{R}^2,$$

for some scalar $a \in \mathbb{R}$ and vector $\mathbf{b} \in \mathbb{R}^2$. Then

$$a = \tau(\mathbf{x}) \qquad \text{and} \qquad \mathbf{b} = \nabla\tau(\mathbf{x}). \tag{1}$$

For each unit vector $\mathbf{v} = (\cos\theta, \sin\theta)$ in $\mathbb{R}^2$, $0 \le \theta < 2\pi$, the ray $\{\mathbf{x} + r\mathbf{v} : r \ge 0\}$ intersects the boundary $\partial\Omega$ at the unique point (because Ω is convex)

$$\mathbf{p} = \mathbf{x} + \rho\mathbf{v}, \tag{2}$$

where $\rho = |\mathbf{p} - \mathbf{x}|$, the Euclidean distance between $\mathbf{p}$ and $\mathbf{x}$, see Fig. 1. Let $q_{\mathbf{v}}$ be the Hermite cubic polynomial such that

$$q_{\mathbf{v}}^{(i)}(0) = D_{\mathbf{v}}^i \tau(\mathbf{x}), \qquad q_{\mathbf{v}}^{(i)}(\rho) = D_{\mathbf{v}}^i g(\mathbf{p}), \qquad i = 0, 1, \tag{3}$$

where $D_{\mathbf{v}} g$ denotes the directional derivative of g in the direction $\mathbf{v}$. Then we find τ_* that minimizes

$$E(\tau) := \int_0^{2\pi} \int_0^{\rho} \left(q_{\mathbf{v}}''(r)\right)^2 dr\, d\theta,$$

and set

$$g(\mathbf{x}) = \tau_*(\mathbf{x}) = a_*. \tag{4}$$

It was shown in Floater and Schulz (2008) that τ_* is unique and numerical examples in Floater and Schulz (2008) showed convincingly that g is C^1, although a proof was not found. However, it was proven that if f is a cubic polynomial then $g = f$, i.e., that the method reproduces polynomials of degree ≤ 3, and in this case g is clearly C^1.

How do we find τ_* and subsequently a_*? One approach is as follows. One can first show (Floater, 2015) that $E(\tau)$ is minimized if and only if

$$\int_0^{2\pi} q_{\mathbf{v}}'''(0)\, d\theta = 0 \quad \text{and} \quad \int_0^{2\pi} q_{\mathbf{v}}''(0)\mathbf{v}\, d\theta = \mathbf{0}. \tag{5}$$