ELSEVIER

Computer-Aided Design 34 (2002) 529–539

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Exact and efficient computation of moments of free-form surface and trivariate based geometry

Octavian Soldea, Gershon Elber*, Ehud Rivlin

Department of Computer Science, Technion-IIT, Haifa 32000, Israel

Abstract

Two schemes for computing moments of free-form objects are developed and analyzed. In the first scheme, we assume that the boundary of the analyzed object is represented using parametric surfaces. In the second scheme, we represent the boundary of the object as a constant set of a trivariate function. These schemes rely on a pre-computation step which allows fast re-evaluation of the moments when the analyzed object is modified. Both schemes take advantage of a representation that is based on the B-spline blending functions. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Surface and volume analysis; Object recognition; Moments of inertia; Dynamics

1. Introduction

Moments of objects are intrinsic to their shape. Thus, the ability to compute moments is necessary in a vast range of applications. Moments of inertia are used in mechanical design and analysis. For example, in the design of aircraft, ships, and automobiles the computation of moments of inertia (and products of inertia) are employed toward the determination of the dynamics of the vehicle. Moreover, in these application areas, B-spline surfaces are now the standard representation used in design.

Moments of inertia are even used in animation of human characters [7,8]. In Refs. [7,8], control systems for human characters that perform several motion operations such as running, bicycling, or vaulting are considered. The animation processes require several successive re-evaluations of parameters that are described by equations, taking into account the moments of inertia of different body parts [8]. Moreover, in Ref. [7] the authors describe algorithms for automatically adapting existing simulated behaviors to new characters and require moments of inertia of the different body parts toward this end as well. A metamorphosis animation from a man to a woman was performed by creating intermediate models and relying on linear interpolation of the moments of inertia of the simulated body parts. Another useful application of moments can be found, for example, in optical character recognition systems [1]. Being intrinsic, moments of inertia can clearly be used in other object recognition applications; see Ref. [15] for a survey.

The literature on the computation of moments is vast. Work that deals with the problem of fast computation of moments in different models can be found, for example [10,14,18]. In Ref. [14], the authors describe a model in which the volume of solids bounded by subdivision surfaces can be approximated. They sketch an extension to the evaluation of higher-order moments as well. However, in Ref. [14] the stress is on the subdivision process that the authors manage toward the volume estimation technique and its complexity.

In Ref. [10], the authors define the $(p + q + r)$-th order moment as

$$m_{p,q,r} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} x^p y^q z^r f(x, y, x),$$

where $N \times N \times N$ is the size of the 3D volume of voxels, and

$$f(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \text{ is in the object} \\ 0 & \text{otherwise} \end{cases}.$$

Ref. [10] only deals with uniform density objects that are represented in uniform resolutions. A computation model is presented in which moments are evaluated with no need for multiplications, given uniform density objects and uniform resolution. The trivariate expressions of $x^p y^q z^r$, required in the moments definition, are computed in a pre-processing scheme. In order to compute $m_{p,q,r}$, one is required to

* Corresponding author. Tel.: +972-4-829-4338; fax: +972-4-822-1128.
E-mail address: gershon@cs.technion.ac.il (G. Elber).

0010-4485/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(01)00124-5

compute all the moments $m_{p',q',r'}$ where $p' \leq p$, $q' \leq q$, $r' \leq r$. Ref. [10] presents a sequential algorithm, a parallel one, and a VLSI chip design which performs the computation. There is no generalization to the computation of moments of objects in higher dimension, although it is possible.

In Ref. [18], the authors use a variant of the discrete divergence theorem that is featured in Ref. [4]. The computation is based on discretization of spels (space elements) of the domain of an implicit function which describes the boundary of the object. The authors admit that this method is sensitive to noise and discretization errors. The algorithm assumes a pre-computation stage that performs surface tracking. The input of the surface tracking algorithm consists of a point on the surface. Moreover, due to the surface tracking stage, the proposed algorithm assumes that the input objects have no holes. An $O(N^2)$ complexity algorithm is claimed for convex objects. As in Ref. [10], in order to compute $m_{p,q,r}$, one is required to compute all the moments $m_{p',q',r'}$ where $p' \leq p$, $q' \leq q$, $r' \leq r$. This algorithm can be generalized for higher dimension objects.

Other related results are Refs. [5] and [2]. In Ref. [5], the authors assume that the input is a free-form object bounded by parametric surfaces. They employ the Stokes theorem [4] to transform the computation of moments using integrals on volumes to integrals on the surface boundaries, of the input objects. In Ref. [2], the author reformulates the computation of the area/volume of an object enclosed by a parametric B-spline curve/surface as a bi-linear/tri-linear form. This computation is also based on the Stokes theorem.

In this work, we propose two schemes toward the computation of moments of objects represented in (1) free-form polynomial or rational surface geometry or (2) via constant sets of trivariate polynomial or rational volumetric functions. We will refer to the first scheme as the *surface scheme* and to the second as the *volumetric scheme*. In the surface scheme, we assume that the input object is bounded by a set of parametric B-spline surfaces, whereas in the volumetric scheme we assume that the boundary of the object is defined as a constant set of a trivariate B-spline function. For both representations, a major portion of the computation could be performed a priori, allowing efficient re-evaluations of moment during actual use.

B-spline blending functions are widely used in geometric modeling [3]. These polynomial or rational functions offer many attractive properties such as refinement and subdivisions operations, and local support. Given an object that is represented using parametric B-spline surfaces, there exists some work that deals with the computation of volumes [2]. Objects represented using parametric Bernstein–Bézier forms are considered in Ref. [5]. The results in Ref. [5] could be improved and extended with the proposed pre-computation that is suggested in this work, following ideas from Ref. [2].

The presented surface scheme has the advantage that one can employ non-uniform B-spline functions with different orders in any direction and several resolutions, thus, the presented surface scheme is more general than the models known in the literature until now. The presented volumetric scheme is mostly usable in the context of uniform knot sequences, while the ability to employ non-uniform knot sequences is available at extra computational cost. Interestingly enough, the volumetric scheme allows one to compute the moments of objects with varying densities. In the case of uniform B-spline blending functions and coefficients along a uniform three-dimensional grid, the computational demands could be greatly decreased and significant portions of the computation could be made a priori.

This paper is organized as follows. Section 2 reviews the concept of zero moments, defines the notation used for B-spline functions, and contains a description of the approach that we develop in the following sections. Section 3 lays down the algorithmic approach that we take toward the computation of moments for objects bounded by free-form parametric surfaces. The approach toward the computation of moments of objects bounded by constant sets of trivariate functions is described in Section 4. In Section 5, we go into the details of the implementation of the volumetric scheme for uniform B-spline functions. In Section 6, we present some examples and a comparison between the results computed by our algorithms and values that were derived discretely as well as analytically, for different objects. Finally, we conclude in Section 7.

2. Background

Let $B_{i,k,\tau}(t)$ be the i-th B-spline blending function of order k (degree $k-1$) defined over knot sequence τ. Hereafter, we will simply employ $B_i(t)$ or $B_{i,k}(t)$ to denote $B_{i,k,\tau}(t)$. Given surface $S(u,v) = \sum_i \sum_j P_{ij} B_{i,k_u}(u) B_{j,k_v}(v)$, $P_{ij}(x_{ij}, y_{ij}, z_{ij})$, the direction of the normal, $\mathcal{N}$, of $S(u,v) = (x(u,v), y(u,v), z(u,v))$ is:

$$\mathcal{N} = \left(\frac{\partial y}{\partial u}\frac{\partial z}{\partial v} - \frac{\partial y}{\partial v}\frac{dz}{\partial u}, \frac{\partial z}{\partial u}\frac{\partial x}{\partial v} - \frac{\partial z}{\partial v}\frac{\partial x}{\partial u}, \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u} \right), \tag{1}$$

having $\mathcal{N} = (\mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_z)$. Following Ref. [5], the signed zero moment $\mathcal{M}_0$, or the volume V, enclosed by the parametric surface $S(u, v)$ equals,

$$\mathcal{M}_0 = V = \int_U z\mathcal{N}_z(u,v)\mathrm{d}u\mathrm{d}v$$

$$= \int_U z\left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u} \right)\mathrm{d}u\mathrm{d}v \tag{2}$$

where U is the parametric domain of $S(u, v)$. In Ref. [2], the author derives Eq. (2) for B-spline surfaces as follows:

$$V = \int_U z\mathcal{N}_z(u,v)\mathrm{d}u\mathrm{d}v$$

$$= \int_U \sum_{l_u}\sum_{l_v} z_{l_u,l_v} B_{l_u,k_u}(u)B_{l_v,k_v}(v)\left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right)\mathrm{d}u\mathrm{d}v$$

$$= \int_U \sum_{l_u}\sum_{l_v} z_{l_u,l_v} B_{l_u,k_u}(u)B_{l_v,k_v}(v)$$

$$\left(\sum_{i_u}\sum_{i_v} x_{i_u,i_v} B'_{i_u,k_u}(u)B_{i_v k_v}(v)\sum_{j_u}\sum_{j_v} y_{j_u,j_v} B_{j_u,k_u}(u)B'_{j_v,k_v}(v)\right)$$

$$-\sum_{i_u}\sum_{i_v} x_{i_u,i_v} B_{i_u,k_u}(u)B'_{i_v k_v}(v)\sum_{j_u}\sum_{j_v} y_{j_u,j_v} B'_{j_u,k_u}(u)B_{j_v,k_v}(v)\mathrm{d}u\mathrm{d}v$$

$$= \sum_{l_u}\sum_{l_v} z_{l_u,l_v}\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\int_U B_{l_u,k_u}(u)B_{l_v,k_v}(v)$$

$$(B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v k_v}(v)$$

$$- B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v))\mathrm{d}u\mathrm{d}v.$$

This reformulation of the volumes allows one to rewrite Eq. (2) as a tri-linear form in x_{i_u,i_v}, y_{j_u,j_v}, and z_{l_u,l_v}. Let

$$\phi^0_{i_u,i_v j_u,j_v,l_u,l_v}(u,v) = B_{l_u,k_u}(u)B_{l_v,k_v}(v)$$
$$(B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v,k_v}(v)$$
$$-B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v)), \quad (3)$$

and let

$$\Phi^0_{i_u,i_v,j_u,j_v,l_u,l_v} = \int_U \phi^0_{i_u,i_v,j_u,j_v,l_u,l_v}(u,v)\mathrm{d}u\mathrm{d}v.$$

Then, with the assumption of a pre-computation of the values of $\Phi^0_{i_u,i_v,j_u,j_v,l_u,l_v}$ that solely depends on the function space containing $S(u, v)$, one can re-evaluate the volume of surface $S(u, v)$ as a tri-linear product of the coefficients of the surface, P_{ij}:

$$V = \sum_{l_u}\sum_{l_v} z_{l_u,l_v}\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\Phi^0_{i_u,i_v,j_u,j_v,l_u,l_v}. \quad (4)$$

As long as the function space of $S(u, v)$ is fixed, i.e. the same knot sequences and orders, $\Phi^0_{i_u,i_v,j_u,j_v,l_u,l_v}$ remains the same. Moreover, a change in a few control points amounts to the re-evaluation of Eq. (4), for the modified coefficients and terms only.

We will employ a similar scheme to compute moments of higher orders, in the following sections. Expanding $\mathcal{N}_z$ in Eq. (1), we have,

$$\mathcal{N}_z(u,v) = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}$$

$$= \sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}(B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v,k_v}(v)$$

$$-B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v))$$

$$= \sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\psi_{i_u,i_v,j_u,j_v}(u,v), \quad (5)$$

where,

$$\psi_{i_u,i_v,j_u,j_v}(u,v) = B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v,k_v}(v)$$
$$- B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v).$$

Moreover, $\phi^0_{i_u,i_v,j_u,j_v,l_u,l_v}(u,v) = B_{l_u,k_u}(u)B_{l_v,k_v}(v)\psi_{i_u,i_v,j_u,j_v}(u,v)$ (see Eq. (3)).

3. Moments of free-form surface geometry

In Subsection 3.1, first order moments will be considered, whereas in Subsection 3.2 second order moments are discussed. The extension to higher order moments is fairly simple and follows similar guidelines.

3.1. Computation of first order moments

Following Ref. [5], the first order moments of an object enclosed by parametric surface $S(u, v)$ can be expressed as:

$$\begin{cases} \mathcal{M}_1 = \int_U xz\mathcal{N}_z\mathrm{d}u\mathrm{d}v, \\ \mathcal{M}_2 = \int_U yz\mathcal{N}_z\mathrm{d}u\mathrm{d}v, \text{ and} \\ \mathcal{M}_3 = \int_U \frac{z^2}{2}\mathcal{N}_z\mathrm{d}u\mathrm{d}v. \end{cases} \quad (6)$$

Then, by substituting $\mathcal{N}_z(u,v)$ from Eq. (5) into Eq. (6),

$$\mathcal{M}_1 = \int_U \sum_{m_u}\sum_{m_v} x_{m_u,m_v} B_{m_u,k_u}(u)B_{m_v,k_v}(v)$$

$$\sum_{l_u}\sum_{l_v} z_{l_u,l_v} B_{l_u,k_u}(u)B_{l_v,k_v}(v)$$

$$\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\psi_{i_u,i_v,j_u,j_v}(u,v)\mathrm{d}u\mathrm{d}v$$

$$= \sum_{m_u}\sum_{m_v} x_{m_u,m_v}\sum_{l_u}\sum_{l_v} z_{l_u,l_v}\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}$$

$$\int_U B_{m_u,k_u}(u)B_{m_v,k_v}(v)B_{l_u,k_u}(u)B_{l_v,k_v}(v)\psi_{i_u,i_v,j_u,j_v}(u,v)\mathrm{d}u\mathrm{d}v$$

$$= \sum_{m_u}\sum_{m_v} x_{m_u,m_v}\sum_{l_u}\sum_{l_v} z_{l_u,l_v}\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\Phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v},$$

where

$$\Phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v} = \int_U \phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}(u,v)dudv,$$

and

$$\phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}(u,v) = B_{m_u,k_u}(u)B_{m_v,k_v}(v)B_{l_u,k_u}(u)B_{l_v,k_v}(v)\psi_{i_u,i_v,j_u,j_v}(u,v).$$

Following the computation of $\mathscr{M}_1$, the values $\mathscr{M}_2$ and $\mathscr{M}_3$ can be similarly derived as,

$$\mathscr{M}_2 = \sum_{m_u}\sum_{m_v} y_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \Phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v},$$

and

$$\mathscr{M}_3 = \frac{1}{2}\sum_{m_u}\sum_{m_v} z_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \Phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}.$$

$\Phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}$ can be pre-computed, once the function space of the surface $S(u, v)$ is determined. That is, the knot sequences and the orders of the surface are set.

3.2. Computation of second order moments

Following Ref. [5], the second moments of an object enclosed by parametric surface $S(u, v)$ can be expressed as:

$$\mathscr{M}_{11} = \int_U x^2 z \mathscr{N}_z \mathrm{d}u\mathrm{d}v,$$

$$\mathscr{M}_{12} = \int_U xyz \mathscr{N}_z \mathrm{d}u\mathrm{d}v,$$

$$\mathscr{M}_{13} = \int_U x\frac{x^2}{2} \mathscr{N}_z \mathrm{d}u\mathrm{d}v,$$

$$\mathscr{M}_{22} = \int_U y^2 z \mathscr{N}_z \mathrm{d}u\mathrm{d}v,$$

$$\mathscr{M}_{23} = \int_U y\frac{z^2}{2} \mathscr{N}_z \mathrm{d}u\mathrm{d}v,$$

and

$$\mathscr{M}_{33} = \int_U y\frac{z^3}{3} \mathscr{N}_z \mathrm{d}u\mathrm{d}v.$$

Then,

$$\begin{aligned}
\mathscr{M}_{11} = & \int_U \sum_{n_u}\sum_{n_v} x_{n_u,n_v} B_{n_u,k_u}(u)B_{n_v,k_v}(v) \\
& \sum_{m_u}\sum_{m_v} x_{m_u,m_v} B_{m_u,k_u}(u)B_{m_v,k_v}(v) \sum_{l_u}\sum_{l_v} z_{l_u,l_v} B_{l_u,k_u}(u)B_{l_v,k_v}(v) \\
& \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v}\, \psi_{i_u,i_v,j_u,j_v,}(u,v)\mathrm{d}u\mathrm{d}v \\
= & \sum_{n_u}\sum_{n_v} x_{n_u,n_v} \sum_{m_u}\sum_{m_v} x_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \int_U B_{n_u,k_u}(u)B_{n_v,k_v}(v)B_{m_u,k_u}(u)B_{m_v,k_v}(v)B_{l_u,k_u}(u)B_{l_v,k_v}(v) \\
& \psi_{i_u,i_v,j_u,j_v}(u,v)\mathrm{d}u\mathrm{d}v \\
= & \sum_{n_u}\sum_{n_v} x_{n_u,n_v} \sum_{m_u}\sum_{m_v} x_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v},
\end{aligned}$$

where

$$\Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v} = \int_U \phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v}(u,v)dudv$$

and

$$\begin{aligned}
& \phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v}(u,v) \\
& = B_{n_u,k_u}(u)B_{n_v,k_v}(v)\phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}(u,v) \\
& = B_{n_u,k_u}(u)B_{n_v,k_v}(v)B_{m_u,k_u}(u)B_{m_v,k_v}(v)B_{l_u,k_u}(u)B_{l_v,k_v}(v)\psi_{i_u,i_v,j_u,j_v}(u,v).
\end{aligned}$$

Similarly, the values of $\mathscr{M}_{12}$, $\mathscr{M}_{13}$, $\mathscr{M}_{22}$, $\mathscr{M}_{23}$, and $\mathscr{M}_{33}$ can be evaluated into:

$$\begin{aligned}
\mathscr{M}_{12} = & \sum_{n_u}\sum_{n_v} x_{n_u,n_v} \sum_{m_u}\sum_{m_v} y_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v},
\end{aligned}$$

$$\begin{aligned}
\mathscr{M}_{13} = & \frac{1}{2}\sum_{n_u}\sum_{n_v} x_{n_u,n_v} \sum_{m_u}\sum_{m_v} z_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v},
\end{aligned}$$

$$\begin{aligned}
\mathscr{M}_{22} = & \sum_{n_u}\sum_{n_v} y_{n_u,n_v} \sum_{m_u}\sum_{m_v} y_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v},
\end{aligned}$$

$$\begin{aligned}
\mathscr{M}_{23} = & \frac{1}{2}\sum_{n_u}\sum_{n_v} y_{n_u,n_v} \sum_{m_u}\sum_{m_v} z_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \\
& \Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v},
\end{aligned}$$

and

$$\mathscr{M}_{33} = \frac{1}{3}\sum_{n_u}\sum_{n_v} z_{n_u,n_v}\sum_{m_u}\sum_{m_v} z_{m_u,m_v}\sum_{l_u}\sum_{l_v} z_{l_u,l_v}\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}$$
$$\Phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v}.$$

The computation of the Φ^i functions demands the ability to compute products and integrals of B-spline basis functions. See Refs. [9,13] for several ways to evaluate these operators in the B-spline domain.

4. Moments of trivariate based geometry

In Subsection 4.1, we will introduce several key results that will be employed in Subsection 4.2 toward the computation of the moments. That is, they are integrals on products of B-spline blending functions with monomials.

4.1. Integration of products of B-spline blending functions with monomials

Assume $[t_m, t_{m+1})$ is the last nontrivial interval of some B-spline function, and denote by x^i the i-th power of x while $x^{(i)}$ denotes the i-th derivative of x.

Proposition 4.1. The following identities hold:

$$\int B_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\sum_{j=i}^{m} B_{j,k+1}(t), \tag{7}$$

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\left(t\sum_{j=i}^{m} B_{j,k+1}(t) - \sum_{j=i}^{m} B_{j,k+2}(t)\sum_{s=i}^{j}\frac{(t_{s+k+1} - t_s)}{k+1}\right) \tag{8}$$

.

Proof. Eq. (7) is proved in Ref. [9]. We will prove the identity (8) here.

The derivative of a B-spline basis function equals [3]:

$$B'_{i,k}(t) = (k-1)\left[\frac{B_{i,k-1}(t)}{t_{i+k-1} - t_i} - \frac{B_{i+1,k-1}(t)}{t_{i+k} - t_{i+1}}\right].$$

It follows that

$$B_{i,k-1}(t) = \frac{t_{i+k-1} - t_i}{k-1}B'_{i,k}(t) + \frac{t_{i+k-1} - t_i}{t_{i+k} - t_{i+1}}B_{i+1,k-1}(t),$$

or

$$B_{i,k}(t) = \frac{t_{i+k} - t_i}{k}B'_{i,k+1}(t) + \frac{t_{i+k} - t_i}{t_{i+k+1} - t_{i+1}}B_{i+1,k}(t). \tag{9}$$

Multiplying both sides of Eq. (9) by t and integrating, we have,

$$\int tB_{i,k}(t)\mathrm{d}t = \int t\frac{t_{i+k} - t_i}{k}B'_{i,k+1}(t)\mathrm{d}t$$
$$+ \int t\frac{t_{i+k} - t_i}{t_{i+k+1} - t_{i+1}}B_{i+1,k}(t)\mathrm{d}t$$
$$= \frac{t_{i+k} - t_i}{k}\int tB'_{i,k+1}(t)\mathrm{d}t$$
$$+ \frac{t_{i+k} - t_i}{t_{i+k+1} - t_{i+1}}\int tB_{i+1,k}(t)\mathrm{d}t.$$

Applying Eq. (9) successively while $i < m$, one obtains,

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\sum_{j=i}^{m-1}\int tB'_{j,k+1}(t)\mathrm{d}t$$
$$+ \frac{t_{i+k} - t_i}{t_{m+k} - t_m}\int tB_{m,k}(t)\mathrm{d}t.$$

Multiplying Eq. (9) by t and integrating for the last nontrivial domain $[t_m, t_{m+1})$ we have

$$\int tB_{m,k}(t)\mathrm{d}t = \int t\frac{t_{m+k} - t_m}{k}B'_{m,k+1}(t)\mathrm{d}t$$

and hence,

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\sum_{j=i}^{m}\int tB'_{j,k+1}(t)\mathrm{d}t. \tag{10}$$

Integrating Eq. (10) by parts,

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\left(\sum_{j=i}^{m} tB_{j,k+1}(t) - \sum_{j=i}^{m}\int B_{j,k+1}(t)\right).$$

Using Eq. (7) one obtains:

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\left(\sum_{j=i}^{m} tB_{j,k+1}(t) - \sum_{j=i}^{m}\frac{t_{j+k+1} - t_j}{k+1}\sum_{s=j}^{m} B_{s,k+2}(t)\right).$$

Therefore,

$$\int tB_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\left(t\sum_{j=i}^{m} B_{j,k+1}(t) - \sum_{j=i}^{m} B_{j,k+2}(t)\sum_{s=i}^{j}\frac{t_{s+k+1} - t_s}{k+1}\right),$$

proving Eq. (8).□

Recall that $x^{(i)}$ denotes the i-th derivative of x. Let $\delta_{x,y} = (t_{x+y} - t_x)/y$. Then,

Theorem 4.2.

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\sum_{j=0}^{l}(-1)^j(t^l)^{(j)}\sum_{s_0=i}^{m}B_{s_0,k+j+1}(t)$$
$$\sum_{s_1=i}^{s_)}\delta_{s_1,k+j}\cdots\sum_{s_j=i}^{s_{j-1}}\delta_{s_j,k+1}, \tag{11}$$

for any i, k, $l \geq 0$.

Proof. We prove that Eq. (11) holds for any i, $k \geq 0$ by induction on $l \geq 0$. In all stages of the proof we take into account that i, $k \geq 0$ but do not write it explicitly.

From Proposition 4.1 Eqs. (7) and (8) hold. These equalities are special cases of Eq. (11) where $l = 0$ and $l = 1$, respectively.

Let $l > 0$. Assume Eq. (11) holds for $l-1$. We will prove that Eq. (11) holds for l.

Multiply both sides of Eq. (9) by t^l and integrate. We have:

$$\int t^l B_{i,k}(t)\mathrm{d}t = \int t^l \delta_{i,k}B'_{i,k+1}(t)\mathrm{d}t$$
$$+ \int t^l \frac{t_{i+k} - t_i}{t_{i+k+1} - t_{i+1}} B_{i+1,k}(t)\mathrm{d}t$$
$$= \delta_{i,k}\int t^l B'_{i,k+1}(t)\mathrm{d}t$$
$$+ \frac{t_{i+k} - t_i}{t_{i+k+1} - t_{i+1}}\int t^l B_{i+1,k}(t)\mathrm{d}t.$$

As in the proof of Ref. [8], one can successively apply Eq. (9) while $i < m$ and obtain

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\sum_{j=i}^{m-1}\int t^l B'_{i,k+1}(t)\mathrm{d}t$$
$$+ \frac{t_{i+k} - t_i}{t_{m+k} - t_m}\int t^l B_{m,k}(t)\mathrm{d}t.$$

For the last nontrivial domain we have,

$$\int t^l B_{m,k}(t)\mathrm{d}t = \int t^l \delta_{m,k}B'_{m,k+1}(t)\mathrm{d}t,$$

or

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\sum_{j=i}^{m}\int t^l B'_{j,k+1}(t)\mathrm{d}t. \tag{12}$$

Integrating Eq. (12) by parts,

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\left(\sum_{j=i}^{m} t^l B_{j,k+1}(t) - \sum_{j=i}^{m}\int l t^{l-1}B_{j,k+1}(t)\mathrm{d}t\right).$$

From the inductive hypothesis, Eq. (11) holds for $l-1$. Thus,

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\Bigg(\sum_{j=i}^{m} t^l B_{j,k+1}(t) - l\sum_{j=i}^{m}\Bigg(\sum_{p=0}^{l-1}(-1)^p(t^{l-1})^{(p)}\delta_{j,k+1}$$
$$\sum_{s_0=j}^{m}B_{s_0,k+p+2}(t)\sum_{s_1=j}^{s_0}\delta_{s_1,k+p+1}\cdots\sum_{s_p=j}^{s_{p-1}}\delta_{s_p,k+2}\Bigg)\Bigg)$$
$$= \delta_{i,k}\Bigg(\sum_{j=i}^{m} t^l B_{j,k+1}(t) + \Bigg(\sum_{p=1}^{l}(-1)^p(t^l)^{(p)}\sum_{j=i}^{m}\delta_{j,k+1}\sum_{s_0=j}^{m}B_{s_0,k+p+1}(t)$$
$$\sum_{s_1=j}^{s_0}\delta_{s_1,k+p}\cdots\sum_{s_{p-1}=j}^{s_{p-2}}\delta_{s_{p-1},k+2}\Bigg)\Bigg)$$
$$= \delta_{i,k}\Bigg(\sum_{j=i}^{m} t^l B_{j,k+1}(t) + \Bigg(\sum_{p=1}^{l}(-1)^p(t^l)^{(p)}\sum_{s_0=i}^{m}B_{s_0,k+p+1}(t)\sum_{j=i}^{s_0}\delta_{j,k+1}$$
$$\sum_{s_1=j}^{s_0}\delta_{s_1,k+p}\cdots\sum_{s_{p-1}=j}^{s_{p-2}}\delta_{s_{p-1},k+2}\Bigg)\Bigg)$$
$$= \delta_{i,k}\Bigg(\sum_{j=i}^{m} t^l B_{j,k+1}(t) + \Bigg(\sum_{p=1}^{l}(-1)^p(t^l)^{(p)}\sum_{s_0=i}^{m}B_{s_0,k+p+1}(t)\sum_{s_1=i}^{s_0}\delta_{s_1,k+p}$$
$$\sum_{j=i}^{s_1}\delta_{j,k+1}\cdots\sum_{s_2=j}^{s_1}\delta_{s_2,k+p}\cdots\sum_{s_{p-1}=j}^{s_{p-2}}\delta_{s_{p-1},k+2}\Bigg)\Bigg)$$
$$= \delta_{i,k}\Bigg(\sum_{j=i}^{m} t^l B_{j,k+1}(t) + \Bigg(\sum_{p=1}^{l}(-1)^p(t^l)^{(p)}\sum_{s_0=i}^{m}B_{s_0,k+p+1}(t)$$
$$\sum_{s_1=i}^{s_0}\delta_{s_1,k+p}\cdots\sum_{s_{p-1}=i}^{s_{p-2}}\delta_{s_{p-1},k+2}\sum_{j=i}^{s_{p-1}}\delta_{j,k+1}\Bigg)\Bigg).$$

Thus,

$$\int t^l B_{i,k}(t)\mathrm{d}t = \delta_{i,k}\Bigg(\sum_{p=0}^{l}(-1)^p(t^l)^{(p)}\sum_{s_0=i}^{m}B_{s_0,k+p+1}(t)$$
$$\sum_{s_1=i}^{s_0}\delta_{s_1,k+p}\cdots\sum_{s_{p-1}=i}^{s_{p-2}}\delta_{s_{p-1},k+2}\sum_{j=i}^{s_{p-1}}\delta_{j,k+1}\Bigg)$$

or Eq. (11) holds for l.□

Corollary 4.3. The following equality holds as a particular case of Theorem 4.2 for $l = 2$:

$$\int t^2 B_{i,k}(t)\mathrm{d}t = \frac{t_{i+k} - t_i}{k}\left(t^2 \sum_{j=i}^{m} B_{j,k+1}(t) - 2t \sum_{j=i}^{m} B_{j,k+2}(t) \sum_{s=i}^{j} \frac{t_{s+k+1} - t_s}{k+1} + 2\sum_{j=i}^{m} B_{j,k+3}(t) \sum_{s=i}^{j} \frac{t_{s+k+2} - t_s}{k+2} \sum_{p=i}^{s} \frac{t_{p+k+1} - t_p}{k+1}\right). \tag{13}$$

We will employ the results of Eqs. (7), (8), and (13) in the coming section.

4.2. Computation of moments of trivariate based geometry

Let dm be a differential element of mass. In our system, the mass distribution is defined as:

$$f(u, v, w) = \sum_{i,j,l} p_{i,j,l} B_i(u) B_j(v) B_l(w),$$

where $p_{i,j,l}$ is the (i, j, l)-th scalar coefficient of the trivariate function f.

The computation of the zero moment assumes the evaluation of the integral

$$I = \int_x \int_y \int_z \mathrm{d}m,$$

whereas the first order moments can be obtained from the values

$$I_x = \int_x \int_y \int_z x\mathrm{d}m, I_y = \int_x \int_y \int_z y\mathrm{d}m, \text{and } I_z = \int_x \int_y \int_z z\mathrm{d}m.$$

The computation of the principal axes of inertia assumes the evaluation of the following integrals:

$$I_{xx} = \int_x \int_y \int_z (y^2 + z^2)\mathrm{d}m, I_{yy} = \int_x \int_y \int_z (x^2 + z^2)\mathrm{d}m,$$

$$I_{zz} = \int_x \int_y \int_z (x^2 + y^2)\mathrm{d}m, I_{xy} = \int_x \int_y \int_z xy\mathrm{d}m, \tag{14}$$

$$I_{xz} = \int_x \int_y \int_z xz\mathrm{d}m, \text{and } I_{yz} = \int_x \int_y \int_z yz\mathrm{d}m.$$

Thus, any of the values of I, I_x, I_y, I_z, I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, and I_{yz} could be computed with the aid of Eqs. (7), (8), and (13). For example,

$$I_{xx} = \int_u \int_v \int_w \sum_{i,j,l} p_{i,j,l}(v^2 + w^2) B_i(u) B_j(v) B_l(w)\mathrm{d}u\mathrm{d}v\mathrm{d}w$$

$$= \sum_{i,j,l} p_{i,j,l}\bigg(\int_u B_i(u)\mathrm{d}u \int_v v^2 B_j(v)\mathrm{d}v \int_w B_l(w)\mathrm{d}w + \int_u B_i(u)\mathrm{d}u \int_v B_j(v)\mathrm{d}v \int_w w^2 B_l(w)\mathrm{d}w\bigg).$$

Any one of the integrals of Eq. (14) can be similarly decomposed, computing all the moments of inertia. In order to evaluate these volume integrals, one can use the identities of Proposition 4.1.

5. Implementation details

We have implemented the computation of moments for trivariate uniform B-spline blending functions with floating end conditions. The assumption of the uniformity of the knots allows us significant simplifications in the computations. Assume $t_{i+1} = t_i + \gamma$ for any $i \in N$.

Proposition 5.1. Given a uniform knot sequence, the following equalities hold:
(i)

$$\int_{t_{i+l}}^{t_{i+l+k}} B_{i+l,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} B_{i,k}(t)\mathrm{d}t,$$

(ii)

$$\int_{t_{i+l}}^{t_{i+l+k}} tB_{i+l,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} tB_{i,k}(t)\mathrm{d}t + \gamma l \int_{t_i}^{t_{i+k}} B_{i,k}(t)\mathrm{d}t,$$

and (iii)

$$\int_{t_{i+l}}^{t_{i+l+k}} t^2 B_{i+l,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} t^2 B_{i,k}(t)\mathrm{d}t + 2\gamma l \int_{t_i}^{t_{i+k}} tB_{i,k}(t)\mathrm{d}t + \gamma^2 l^2 \int_{t_i}^{t_{i+k}} B_{i,k}(t)\mathrm{d}t.$$

Proof. Since the blending functions are uniform, $B_{i,k}(t) = B_{i+l,k}(t + \gamma l)$, and (i) holds. Moreover,

$$\int_{t_{i+l}}^{t_{i+l+k}} tB_{i+l,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} (t + \gamma l) B_{i+l,k}(t + \gamma l)\mathrm{d}t = \int_{t_i}^{t_{i+k}} (t + \gamma l) B_{i,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} tB_{i,k}(t)\mathrm{d}t + \gamma l \int_{t_i}^{t_{i+k}} B_{i,k}(t)\mathrm{d}t$$

and thus (ii) holds. Similarly,

$$\int_{t_{i+l}}^{t_{i+l+k}} t^2 B_{i+l,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} (t + \gamma l)^2 B_{i+l,k}(t + \gamma l)\mathrm{d}t = \int_{t_i}^{t_{i+k}} (t + \gamma l)^2 B_{i,k}(t)\mathrm{d}t = \int_{t_i}^{t_{i+k}} t^2 B_{i,k}(t)\mathrm{d}t + 2\gamma l \int_{t_i}^{t_{i+k}} tB_{i,k}(t)\mathrm{d}t + \gamma^2 l^2 \int_{t_i}^{t_{i+k}} B_{i,k}(t)\mathrm{d}t$$

and thus (iii) holds.□

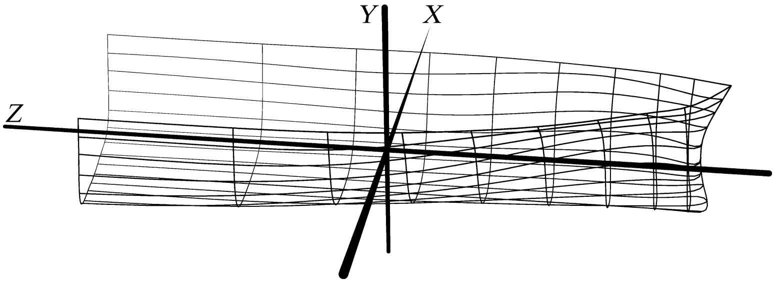

Fig. 1. The moments of the hull of the presented ship are computed in Table 1. This part is centered around its center of mass, computed using the presented algorithm.

We have symbolically computed the values of

$$\int_{t_0}^{t_k} B_{0,k}(t)\mathrm{d}t, \int_{t_0}^{t_k} tB_{0,k}(t)\mathrm{d}t, \text{and} \int_{t_0}^{t_k} t^2B_{0,k}(t)\mathrm{d}t, \tag{15}$$

for specific orders in the Maple symbolic manipulation environment [11]. Now, with the aid of the formulas of Proposition 5.1, one can efficiently recompute the integrals (15) for all discrete domains t_i to t_{i+k}, recursively.

Let $\mathscr{M}$ be the mass of the object and let $\bar{I}_{xx}$ be the moment of inertia along the axis Ox when the center mass, (x_c, y_c, z_c), of the object is in general position. Further, let I_{xx} be the same moment of inertia when the center mass is at the origin. I_{xx} is known as the *canonical moment*. We will relate to $\bar{I}_{yy}$, $\bar{I}_{zz}, \bar{I}_{xy}, \bar{I}_{xz}, \bar{I}_{yz}$, I_{yy}, I_{zz}, I_{xy}, I_{xz}, and I_{yz} in a similar fashion. The parallel axis theorems, (see Ref. [12]), provide the relations between the values of $\bar{I}_{xx}$, $\bar{I}_{yy}$, $\bar{I}_{zz}$, $\bar{I}_{xy}$, $\bar{I}_{xz}$, $\bar{I}_{yz}$, and I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, and I_{yz}. From the parallel axis theorems, $\bar{I}_{xx} = I_{xx} + \mathscr{M}(y_\mathrm{c}^2 + z_\mathrm{c}^2)$ and similar equations hold for all the other values. We only consider canonical moments in the examples and tables of Section 6.

In order to avoid numerical computation errors due to the fact that $\bar{I}_{xx}$ could be much larger than I_{xx} we proceed as follows. After the pre-computation of Eq. (15), we shift the parametric domain of the object so that its center mass is as close as possible to the origin. Denote by $\{t_i^x\}$, $\{t_j^y\}$, and $\{t_l^x\}$ the knot sequences corresponding to the directions of x, y, and z, respectively. Let $(t_{i_\mathrm{c}}^x, t_{j_\mathrm{c}}^y, t_{l_\mathrm{c}}^z) = (\lfloor x_\mathrm{c} \rfloor, \lfloor y_\mathrm{c} \rfloor, \lfloor z_\mathrm{c} \rfloor)$. Finally, compute the moments of the objects around $(x_\mathrm{c} - t_{i_\mathrm{c}}^x, y_\mathrm{c} - t_{j_\mathrm{c}}^y, z_\mathrm{c} - t_{l_\mathrm{c}}^z)$. In Section 6, we refer to the computation of the integrals in Eq. (15) as pre-computation or table construction as it could be conducted off-line and to all other operations as the computation of the moments, that are conducted in real time.

6. Experimental results

We have computed the moments of several free-form surfaces as well as trivariate based geometry. We have employed the Irit [6] modeling environment in our implementation and experiments that are presented in this section. We have analyzed the behavior of our algorithms on trivariate objects such as a human head, and primitive trivariate shapes such as spheres, as well as bivariate geometry such as a ship hull.

We have computed the first and second order moments of inertia of a ship hull, see Refs. [16,17] for the data set. This

Table 1
Moments of a ship hull shown in Fig. 1

Mass	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}
6325.3	667.49	297.11	6286.8	-3.3276×10^{-10}	5.2063×10^{-10}	17,442

Table 2
Trivariate methods based, discrete, and analytic computations of moments of a sphere

r	Patch	Trivariate			Discrete			Analytic	
		Mass	$I_{xx} = I_{yy} = I_{zz}$	Time (s)	Mass	$I_{xx} = I_{yy} = I_{zz}$	Time (s)	Mass	$I_{xx} = I_{yy} = I_{zz}$
1/2	24^3	111.30	400.9	0.066	111.30	396.3	0.01	113.10	407.2
1/2	48^3	112.79	406.5	0.518	112.79	405.3	0.064	113.10	407.2
1/2	96^3	112.98	406.7	4.082	112.98	406.4	0.545	113.10	407.2
1/8	128^3	266.73	1712	10.44	266.73	1701	1.338	268.08	1716
1/4	128^3	2141.6	54,860	10.54	2141.6	54,771	1.327	2144.7	54,903
1/2	128^3	17,155	1757×10^3	10.41	17,155	1756×10^3	1.35	17,157	1757×10^3

surface was modeled as a bicubic B-spline surface of size 8 × 8 mesh. The surface has a fixed thickness set to it (2 inches thick for a total length of shown part of 384 feet).

The computation consisted of two steps. First, we computed the volume and the first order moments, and translated the center of mass of the ship hull into the origin. In the second step we computed the moments of second order.

Fig. 1 shows the surface of a ship hull centered at the origin. Table 1 presents the mass and the second order computed moments. I_{xy} and I_{xz} are virtually zero due to the symmetry of the shape.

We have also computed the moments of inertia with the presented trivariate based method, of several primitives such as spheres and cylinders, as well as complex volumetric data such as a medical scan of a human head and a human knee. The results were compared to the straightforward discrete computation approach, in which we computed the moments of the objects by summing up over all the voxels and taking into account the distances of the voxels to the origin. Moreover, our results were compared to the analytic solution, whenever available. This comparison allows us to analyze the accuracy of our model. Moreover, as will be shortly shown, not only have we been able to evaluate the moments directly from the trivariate, but the trivariate based methods provide a better approximation, as compared to the discrete approach.

We have executed our program on a SUN Ultra-Enterprise E4500. The pre-processing stage time is bounded by 0.19 ms for the most complex patch, that consists of 128 × 128 × 256 scalar values.

In the following tables, the spheres and the cylinders are characterized by radius r and height h. All the trivariate patches of the spheres and cylinders are cubes with a unit size.

In Table 2, the moments of inertia of six different spheres are compared. Here, we have that $I_{xy} = I_{xz} = I_{yz} = 0$. The computation times of the algorithm based on trivariate methods and the times required for the discrete method are also provided in seconds.

In Table 3, the moments of inertia of six cylinders are compared. Again, here we have that $I_{xy} = I_{xz} = I_{yz} = 0$. The computation times of the algorithm based on the introduced trivariate methods and the times required for the discrete method are also provided in seconds.

We also employ two real life examples—a human head and a human knee. The scalar values that were provided in these data sets were assumed to reflect the mass. While not necessarily the case, one can always map these scalar values that hint on the tissue type into the actual mass.

In Table 4, we employ the algorithm of the computations of moments on a real scanned image of a human head (see Fig. 2). While our algorithm needs 4.368 s to compute the moments, the discrete algorithm needs 0.503 s for the same task. The time required to modify one scalar coefficient and re-evaluate

Table 3
Trivariate methods based, discrete, and analytic computations of moments of a cylinder

r	h	Patch	Trivariate				Discrete				Analytic		
			Mass	$I_{xx} = I_{yy}$	I_{zz}	Time (s)	Mass	$I_{xx} = I_{yy}$	I_{zz}	Time (s)	Mass	$I_{xx} = I_{yy}$	I_{zz}
1/2	1	$16^2 \times 64$	189.98	4117	372.6	0.074	189.98	4109	364.7	0.012	201.06	4490	402.1
1/2	1	$32^2 \times 128$	196.70	4327	389.9	0.614	196.70	4325	387.9	0.079	201.06	4490	402.1
1/2	1	$64^2 \times 256$	199.53	4423	398.1	4.901	199.5	4422	397.6	0.655	201.06	4490	402.1
1/8	1/4	$128^2 \times 256$	780.61	19,244	6189	20.85	780.61	19,211	6157	2.641	804.25	20,374	6434
1/4	1/2	$128^2 \times 256$	6359.9	6359×10^2	2030×10^2	20.98	6359.9	6356×10^2	2028×10^2	2.612	6434	6520×10^2	2059×10^2
1/2	1	$128^2 \times 256$	51,195	2061×10^4	6546×10^3	21.10	51,195	2061×10^4	6543×10^3	2.604	51,472	2086×10^4	6588×10^3

Table 4
Trivariate methods based computations of moments of a human head

Patch	Mass	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}
$128^2 \times 54$	2.436×10^8	3.920×10^{11}	2.779×10^{11}	5.514×10^{11}	-1.837×10^9	7.137×10^7	-1.334×10^9

Table 5
Trivariate methods based computations of moments of a human knee

Patch	Mass	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}
$128^2 \times 63$	3.853×10^8	6.607×10^{11}	5.290×10^{11}	9.351×10^{11}	-4.873×10^9	-2.341×10^7	-4.296×10^9

the new moments of the object as a result of this single coefficient change is 8.37 ms. In Table 5, we present the results of computations on a human knee (see Fig. 3). While the trivariate based method needs 4.94 s to compute the moments, the discrete method requires 0.58 s for the same task.

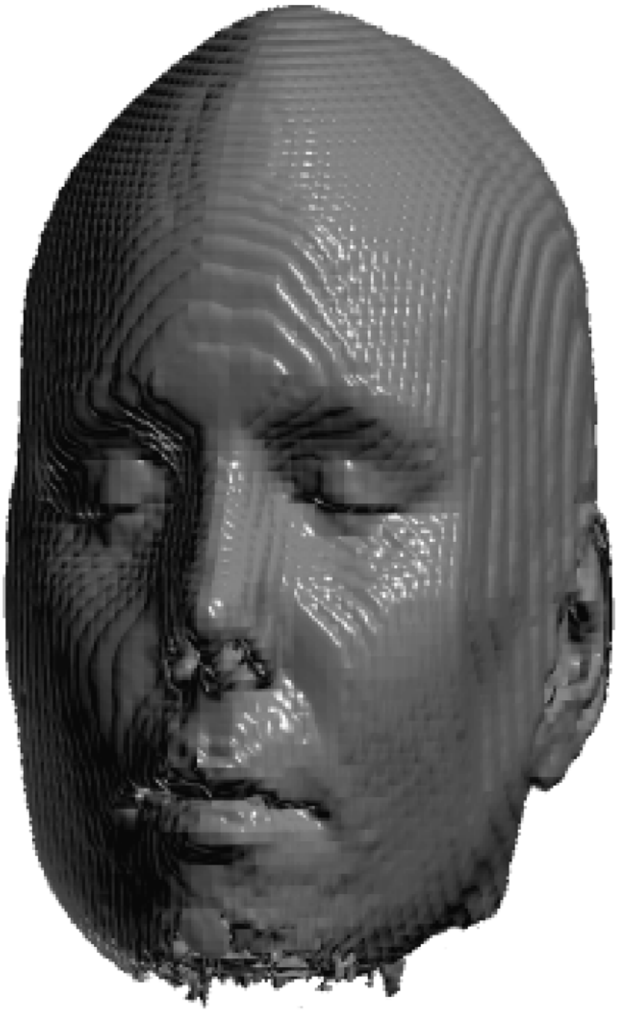

Fig. 2. A human head of which moments are computed and shown in Table 4.

7. Conclusions and future work

In the volumetric scheme, in order to compute a certain moment we do not need to compute all the lower order moments. Our experimental results in Section 6 show that the proposed algorithms result in a direct evaluation scheme of moments that is also more robust and accurate, where the errors of computation for several objects are presented in the tables of Section 6. Moreover, the volumetric scheme is equally valid, regardless of the existence of holes in the object or even discontinuities or disjoint parts.

Finally, while in both schemes we have considered mostly the computation of moments of second order, the generalization for moments of arbitrary orders is technical. Future work will also deal with the possibility of the generalization to objects in higher dimensions.

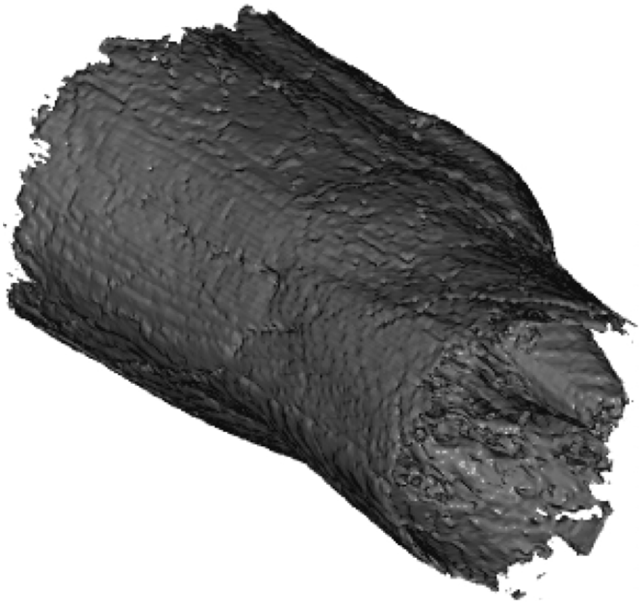

Fig. 3. A human knee of which moments are computed and shown in Table 5.

Acknowledgements

Figs. 2 and 3 are part of the UNC public domain volumetric data set that was produced using Siemens Medical Systems Inc.

References

[1] Cash GL, Hatamian M. Optical character recognition by the method of moments. Computer Vision, Graphics and Image Processing 1987;39:291–310.

[2] Elber G. Linearizing the area and volume constraints. The Center for Intelligent Systems Tech. Report, CIS 2000-04, Computer Science Department, Technion.

[3] Farin G. Curves and surfaces for CAGD—a practical guide. 4th ed. New York: Academic Press, 1996.

[4] Fikhtengol'ts GM. The fundamentals of mathematical analysis. New York: Pergamon Press, 1965.

[5] Gonzales-Ochoa C, McCamnon S, Peters J. Computing moments of objects enclosed by piecewise polynomial surfaces. ACM Transactions on Graphics 1998;17(3):143–57.

[6] Irit Version 8.0. http://www.cs.technion.ac.il/~irit

[7] Hodgins JK, Pollard NS. Adapting simulated behaviours for new characters. SIGGRAPH 97 Proceedings, Annual Conference Series, Los Angeles, CA. 1997. p. 153–62.

[8] Hodgins JK, Wooten WL, Brogan DC, O'Brien JF. Animating human athletics. SIGGRAPH 95 Proceedings, Annual Conference Series, Los Angeles, CA. 1995. p. 71–8.

[9] Kazinnik R, Elber G. Orthogonal decomposition of non-uniform B-spline spaces using wavelets. Eurographics 1997;16(3):C27–38.

[10] Li B-C, Shen J. Pascal triangle transform approach to the calculation of 3D moments. CVGIP: Graphical Models and Image Processing 1992;54(4):301–7.

[11] Maple. www.maplesoft.com, Monagan M, Geddes K, Heal K, Labahn G, Vorkoetter S. Maple V programming guide for Release 5. Springer-Verlag, 1997.

[12] Meriam JL. Dynamics. New York: Wiley, 1966.

[13] Morken K. Some identities for products and degree raising of splines. Constructive Approximation 1991;7:195–208.

[14] Peters J, Nasri A. Computing volumes of solids enclosed by recursive subdivision surfaces. Eurographics 1997;16(3):C89–94.

[15] Prokop RJ, Reeves AP. A survey of moment-based techniques for unoccluded object representation and recognition. CVGIP: Graphical Models and Image Processing 1992;54(5):438–60.

[16] Rogers DF. B-spline surfaces for ship hull design. SIGGRAPH 80 Proceedings, Annual Conference Series, Seattle, WA. 1980. p. 14–8.

[17] Rogers DF, Adams JA. Mathematical elements for computer graphics. 2nd ed. New York: McGraw-Hill, 1990.

[18] Yang L, Albregtsen F, Taxt T. Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalization to higher dimensions. Graphical Models and Image Processing 1997;59(2):97–108.

Octavian Soldea received his BSc (1993) from the Alexandru Ioan Cuza University of Iasi, Romania, and his MSc (1999) from the Technion, Israel Institute of Technology, both in Computer Science. He is a PhD Student in the Computer Science Faculty of the Technion, Israel Institute of Technology. His research interests include computer aided geometric design, computer vision, and continued fractions.

Gershon Elber is an Associate Professor in the Computer Science Department, Technion, Israel. His research interests span computer aided geometric designs and computer graphics. Professor Elber received a BS in computer engineering and an MS in computer science from the Technion, Israel, in 1986 and 1987, respectively, and a PhD in computer science from the University of Utah, USA, in 1992. He is a member of the ACM and IEEE. Professor Elber can be reached at the Technion, Israel Institute of Technology, Department of Computer Science, Haifa 32000, Israel. E-mail: gershon@cs.technion.ac.il

Ehud Rivlin received the BSc and MSc degrees in computer science and the MBA degree from the Hebrew University in Jerusalem, and the PhD from the University of Maryland. Currently he is an Associate Professor in the Computer Science Department at the Technion, Israel Institute of Technology. His current research interests are in machine vision and robot navigation.