Approximate Implicitization using Chebyshev Polynomials

Oliver Barrowclough\textsuperscript{1}

SINTEF ICT
Department of Applied Mathematics

October 8, 2011

\textsuperscript{1}joint work with Tor Dokken (SINTEF)
## Contents

- Introduction and Motivation
  - Curve and surface representation
  - High degree implicit surfaces
- Approximate Implicitization
  - Least squares approach
  - Using orthogonal bases for approximation
- Examples
  - Comparing numerical stability of the approaches
  - Approximate implicitization of Newells’ teapot patches
Introduction

Representation of Curves and Surfaces

- Parametric representation: Rational surface given by
  \[ p(s, t) = (p_1(s, t), p_2(s, t), p_3(s, t), h(s, t)) \quad \text{for} \quad (s, t) \in \Omega \]
  and bivariate polynomials \( p_1, p_2, p_3, h \) (homogeneous form).

- Implicit (algebraic) representation: Surface given by
  \[ \{(x, y, z, w) : q(x, y, z, w) = 0\} \]
  where \( q \) is a polynomial in homogeneous form.

- For intersection algorithms it is useful to have both representations available...
Introduction

Motivation - Intersection Algorithms

(a) Surface-surface intersection  
(b) Surface self-intersection  
(c) Surface raytracing
Implicitization

**Exact methods**

- Traditional methods give exact results:
  - Gröbner bases,
  - Resultants and moving curves/syzygies [Sederberg, 1995],
  - *Linear algebra*.
- Often performed using symbolic computation.
- Surface implicitization can result in very high degrees.
- Algorithms are often slow (especially Gröbner bases).
Implicitization

Implicit degree of parametric surfaces

- Tensor-product bicubic patch
- 16 control points
- Total implicit degree 18
- Defined implicitly by 1330 coefficients!
- Approximation is desirable
Approximate methods

- *Approximate* methods where the degree $m$ can be chosen are desirable:
  - keep the degree low,
  - better stability for floating pt. implementation,
  - faster algorithms.
- Approximation should be good within a region of the parametric curve/surface.
- Algorithms give exact results if the degree is high enough.
Preliminaries

- First, describe implicit polynomial \( q \) in a basis \( (q_k)_{k=1}^{M} \), of degree \( m \):

\[
q(x) = \sum_{k=1}^{M} b_k q_k(x)
\]

with unknown coefficients \( b \).

- A good error measure is given by algebraic distance \( q(p(s)) \).
Approximate Implicitization

Original method (singular value decomposition)

- Original method [Dokken, 1997], gives general framework:
- Form matrix \( D = (d_{jk})_{j,k=1}^{L,M} \) such that
  \[
  q(p(s)) = \sum_{k=1}^{M} b_k q_k(p(s)) = \sum_{k=1}^{M} b_k \sum_{j=1}^{L} \alpha_j(s) d_{jk}.
  \]
  where \((\alpha_j)_{j=1}^{L}\) is a polynomial basis in \(s\).
- An approximation is given by right singular vector \(v_{\text{min}}\)
corresponding to smallest singular value of \(D\).
Approximate Implicitization

**Original method**

- Choosing different polynomial bases solves different approximation problems:
- Orthogonal bases solve continuous least squares problems

\[ \min_{\|b\|_2=1} \int_{\Omega} q(p(s))^2 w(s) \, ds. \]

- Bernstein/Lagrange bases solve problems which approximate the least squares problem.
Approximate Implicitization

### Least squares / weak approximation

- Introduced in [Dokken, 2001], [Corless et al., 2001]:

\[
\min_{\|b\|_2=1} \int_{\Omega} q(p(s))^2 w(s) \, ds.
\]

- Method: Form matrix \( \mathbf{M} = (m_{kl})_{k,l=1}^{M} \),

\[
m_{kl} = \int_{\Omega} q_k(p(s)) q_l(p(s)) w(s) \, ds
\]

- The eigenvector corresponding to the smallest eigenvalue as the solution.
Approximate Implicitization

Orthogonal basis method

The original method using orthogonal polynomials can be used instead:

- Choose a basis \( (T_j)_{j=1}^{L} \) that is orthonormal w.r.t. \( w \):

\[
(M)_{kl} = \int_{\Omega} q_k(p(s))q_l(p(s))w(s) \, ds \\
= \int_{\Omega} \left( \sum_{j=1}^{L} T_j(s)d_{jk} \right) \left( \sum_{i=1}^{L} T_i(s)d_{ik} \right) w(s) \, ds \\
= \sum_{i=1}^{L} \sum_{j=1}^{L} d_{jk}d_{ik} \int_{\Omega} T_j(s)T_i(s)w(s) \, ds \\
= \sum_{j=1}^{L} d_{jk}d_{jl} \\
= (D^T D)_{kl}
\]
Approximate Implicitization

Comparison of methods

- The two methods are mathematically equivalent.
- Singular values of $D$ are square roots of eigenvalues of $D^T D = M$, thus smaller condition numbers for $D$.
- Original method is more numerically stable.
- Original method avoids costly integration of high degree polynomials.
### Why Chebyshev polynomials?

- Near equioscillating behaviour in algebraic error function.
- Number of roots appears to correspond to convergence rates.
- Fast algorithm - based on point sampling, fast Fourier transform (FFT).
- Solves a least squares problem.
- Directly generalizable to tensor-product surfaces.
Approximate Implicitization

Convergence rates of approximate implicitization

<table>
<thead>
<tr>
<th>Implicit degree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence rate</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>20</td>
<td>27</td>
</tr>
</tbody>
</table>

Curves in $\mathbb{R}^2$

<table>
<thead>
<tr>
<th>Implicit degree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence rate</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

Surfaces in $\mathbb{R}^3$

Convergence as we approximate smaller regions of the curve or surface.
Approximate Implicitization

Algorithm - Chebyshev method

- Generate parametric samples $p_j = p(t_j)$ at Chebyshev nodes $t_j = (\cos((j - 1)\pi/(L - 1)) + 1)/2$, for $j = 1, \ldots, L$.
- Compute a matrix $D_0 = (q_k(p_j))_{j=1,k=1}^{L,M}$.
- Compute $D$ by applying Discrete Cosine Transform to columns of $D_0$ (using fast Fourier transform methods).
- Perform SVD of $D$ ($= UV^T$).
Examples

Numerical stability of weak method

- Exact implicitization of degree 5 curve using double precision:

\[
sing(D) = \begin{pmatrix}
\vdots \\
2.45 \times 10^{-6} \\
6.05 \times 10^{-7} \\
3.59 \times 10^{-7} \\
4.58 \times 10^{-8} \\
1.24 \times 10^{-8} \\
6.15 \times 10^{-18}
\end{pmatrix}, \quad
eig(M) = \begin{pmatrix}
\vdots \\
6.02 \times 10^{-12} \\
3.65 \times 10^{-13} \\
1.29 \times 10^{-13} \\
2.09 \times 10^{-15} \\
1.50 \times 10^{-16} \\
6.84 \times 10^{-19}
\end{pmatrix}
\]
Examples

Newell’s 32 teapot patches:

- 32 parametric patches.
- All patches are bicubic.
Examples

Implicitization of teapot spout patches:

- Exact implicit degree 18.
- Approximated by degree 6 surfaces.
- Extra branches present.
- Can combine with other approximations to remove branches.
Examples

Implicitization degrees of Newells’ teapot

<table>
<thead>
<tr>
<th></th>
<th>Exact $m$</th>
<th>Approximate $m$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>32 patches</td>
</tr>
<tr>
<td>rim</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>upper body</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>lower body</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>upper handle</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>lower handle</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>upper spout</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>lower spout</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>upper lid</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>lower lid</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>bottom</td>
<td>15</td>
<td>3</td>
</tr>
</tbody>
</table>
Examples

Implicitization of 32 teapot patches:

- 32 approximately implicitized bicubic patches.
- All patches of degree \( \leq 6 \).
- Extra branches present.
- No continuity conditions used.
Examples

Implicit teapot with fewer patches:

- 26 parametric patches.
- 5 approximately implicitized patches.
- All patches of degree $\leq 6$. 
Approximate Implicitization using Linear Algebra

Thank you!

References:
