On the CAD-compatible conversion of S-patches

Péter Salvi

Budapest University of Technology and Economics

Abstract

S-patches have many nice mathematical properties. It is known since their first appearance, that any regular S-patch can be exactly converted into a trimmed rational Bézier surface. This is a big advantage compared to other multi-sided surface representations that have to be approximated to be exported into CAD/CAM systems. The conversion process, however, remained on the theoretical level, with bits and pieces scattered in multiple publications. In this paper we review the entirety of the algorithm, and investigate it from a practical aspect.

1. Introduction

S-patches offer a mathematically perfect generalization of Bézier surfaces to any number of sides. The three-sided S-patch – the Bézier triangle – is widely adopted, but there is yet no standard representation for surfaces with more than four sides.

The lack of success for S-patches may be due to their complex control net structure, and the large number of control points. These, as it turns out, can be circumvented by the automatic generation of the control network using only boundary constraints.[7]

Another common issue with n-sided surface representations is the conversion to tensor product patches, which is crucial for the processing of the model in CAD/CAM systems. Usually some fitting method is used, which either creates n quadrilaterals sharing a central vertex,[5] or a larger four-sided region with approximated trimming curves. The former loses the C^∞ continuity inside the patch, while the latter has an inherent asymmetry, as well as parameterization issues.[9]

Here S-patches seem to have an advantage: it is shown in the original paper[4] that any regular S-patch can be exactly converted into a trimmed rational Bézier surface. The algorithm for this was not described in detail, however, so in this article we will go through all the required steps.

In Section 2 we will briefly review the related publications. In Section 3 we will define the S-patch both in itself, and as the composition of Bézier simplexes. Then the conversion process is described in Section 4. An example and the discussion of some practical issues conclude the paper.

2. Previous Work

S-patches were first published by Loop and DeRose[4] in 1989, but the techniques involved in the conversion process were developed earlier: see Ramshaw's wonderful book[6] on blossoming, and DeRose's paper[1] on the composition of Bézier simplexes. A more efficient variation[2] of the latter was developed a few years later.

This work offers little novelty. In the following sections, we will try to connect the dots, and insert a few missing pieces (notably the polarization of Wachspress coordinates in Section 4.1.1), hopefully making tensor product conversion easy to understand and implement.

3. S-patches

An n-sided S-patch† is defined over a regular n-gon, parameterized by Wachspress barycentric coordinates $\lambda = (\lambda_1, \ldots, \lambda_n)$.[3] Its control points $\{P_\mathbf{s}\}$ are labeled by n nonnegative integers whose sum is the depth of the surface (d). We will use the notation $L_{n,d}$ for the set of all such labels, and $(s_1, \ldots, s_n) = \mathbf{s} \in L_{n,d}$ for one particular label. The norm of a label is its sum, i.e., $|\mathbf{s}| = d$.

---

† This paper deals only with *regular* S-patches, as only these have the required properties for conversion.

The surface point corresponding to a domain point with barycentric coordinates λ is defined as

$$S(\lambda) = \sum_{\mathbf{s} \in L_{n,d}} P_{\mathbf{s}} \cdot B_{\mathbf{s}}^d(\lambda) = \sum_{\mathbf{s} \in L_{n,d}} P_{\mathbf{s}} \cdot \binom{d}{\mathbf{s}} \cdot \prod_{i=1}^n \lambda_i^{s_i}, \quad (1)$$

where $B_{\mathbf{s}}^d(\lambda)$ are Bernstein polynomials, and

$$\binom{d}{\mathbf{s}} = \frac{d!}{\prod_{i=1}^n s_i!} \quad (2)$$

are their multinomial coefficients.

3.1. S-patches as Bézier simplexes

An $(n-1)$-dimensional simplex has n vertices V_i (e.g. in 2D a triangle, in 3D a tetrahedron). Any point can be uniquely expressed by the affine combination of these vertices:

$$p = \sum_{i=1}^n \lambda_i V_i, \quad \sum_{i=1}^n \lambda_i = 1. \quad (3)$$

The coefficients λ_i are called the barycentric coordinates of p relative to the simplex.

A Bézier simplex of dimension $n-1$ and degree d is a polynomial mapping from the barycentric coordinates relative to an $(n-1)$-dimensional simplex, given in Bernstein form:

$$\sum_{\mathbf{s} \in L_{n,d}} P_{\mathbf{s}} \cdot B_{\mathbf{s}}^d(\lambda), \quad (4)$$

where $P_{\mathbf{s}}$ are called the control points of the Bézier simplex. We can see that this is the same as Eq. (1), i.e., the S-patch is a Bézier simplex that maps from an $(n-1)$-dimensional simplex (n barycentric coordinates) to 3D.

Let us introduce the following notations for a Bézier simplex S:

- n_S : # of arguments (i.e., domain simplex dimension + 1)
- d_S : degree
- δ_S : dimension of the control points

These form the characteristic triple $\phi(S) = (n_S, d_S, \delta_S)$. For example, for an n-sided S-patch of depth d, $\phi(S) = (n, d, 3)$.

4. Conversion to trimmed Bézier patches

According to the original paper,[4] any n-sided S-patch of depth d can be converted into an m-sided rational S-patch of depth $(n-2)d$. Also, a four-sided S-patch of depth d can be converted into a tensor product Bézier patch of degree d.

Consequently, the conversion is done in two stages. First the n-sided S-patch is replaced with a four-sided one (Section 4.1), then the tensor product form is computed (Section 4.2), resulting in a surface of degree $(n-2)d$.

4.1. Conversion to quadrilateral S-patches

The Wachspress coordinates of an n-sided regular polygon have the form

$$\lambda_i(p) = \frac{\prod_{j \neq i-1, i} D_j(p)}{\sum_{k=1}^n \prod_{j \neq k-1, k} D_j(p)}, \quad (5)$$

where $D_j(p)$ is the signed distance of p from the j-th side.

When p is given by barycentric coordinates, this can also be expressed as a rational Bézier simplex W_n, that maps from a 2D simplex to a simplex of dimension $n-1$. This is not an obvious result, since Eq. (5) involves Euclidean distances. For the exact construction, and the handling of rational simplexes, see Section 4.1.1.

It is also shown that W_n is *pseudoaffine*, i.e., it has an affine left inverse W_n^{-1}. With this, we can write

$$S \circ W_n = S \circ W_n \circ (W_4^{-1} \circ W_4) = (S \circ W_n \circ W_4^{-1}) \circ W_4. \quad (6)$$

In other words, the four-sided version of the n-sided depth d S-patch S is the composition of three Bézier simplexes, with the following characteristic triples: $\phi(W_4^{-1}) = (4, 1, 3)$, $\phi(W_n) = (3, n-2, n)$, and $\phi(S) = (n, d, 4)$.

Let us see how to generate control points for each of these Bézier simplexes. Since we will have to work with barycentric coordinates relative to some simplex, define the *canonical simplex* as the canonical basis and the origin. In 2D, this will be the triangle $(1,0)$, $(0,1)$, $(0,0)$, so the barycentric coordinates for an (u,v) point will be $(u, v, 1-u-v)$.

The Bézier simplex W_4^{-1} defines the vertices of the four-sided domain in the plane. Since the tensor product patch will be defined on $[0,1] \times [0,1]$, we will use the same square as the image of the 3D simplex. Thus the control points of W_4^{-1} are:

$$\begin{array}{ll}
P_{1000} = (0,1,0), & P_{0100} = (1,1,-1), \\
P_{0010} = (1,0,0), & P_{0001} = (0,0,1). \quad (7)
\end{array}$$

Getting the control points of W_n needs a bit more work, see the next section.

The control points of S are just the control points of the original n-sided S-patch. The astute reader must have noticed that $\delta_S = 4$. This is because W_n is rational, and the composition algorithm needs S to be also in homogenized barycentric form (see below).

4.1.1. Polarization of the Wachspress coordinates

Homogeneous coordinates are normally represented by adding an extra "weight" coordinate, so $(x,y,z) \equiv (wx, wy, wz, w)$, and projection works by dividing with the weight. With Bézier simplexes, we use another type of homogenization, that allows us to use the simplex composition algorithm as it is, even for the rational case: we take the barycentric coordinates (relative to some simplex) and

use them as "normal" coordinates. If we use the canonical simplex, an (x,y,z) point has the barycentric coordinates $(x,y,z,1-x-y-z)$, so in general the homogenized form will be $(x,y,z) \equiv (wx,wy,wz,w(1-x-y-z))$. Projection is done by dividing with the sum of all coordinates.

The homogenized form of the Wachspress coordinates is thus $\left\{ \prod_{j \neq i-1,i} D_j(p) \right\}$. We will work out the Bézier simplex coordinates for this by using its *blossom* or *polarization*. For any homogeneous polynomial $Q(u)$ of degree d, there is a symmetric multilinear function q of d arguments that agrees with it on its diagonal:

$$q(u_1,\ldots,u_d) = q(u_{\pi_1},\ldots,u_{\pi_d}), \qquad (8)$$

$$q(u_1,\ldots,\alpha u_{k_1} + \beta u_{k_2},\ldots,u_d) = \alpha q(u_1,\ldots,u_{k_1},\ldots,u_d) + \beta q(u_1,\ldots,u_{k_2},\ldots,u_d), \qquad (9)$$

$$q(u,\ldots,u) = Q(u). \qquad (10)$$

Eq. (8) shows symmetry, Eq. (9) multilinearity, and Eq. (10) the diagonal agreement.

The rational Bézier simplex control points for Q are easily computed from its polarization:

$$P_{\mathbf{s}}^Q = q(\underbrace{V_1,\ldots,V_1}_{s_1},\underbrace{V_2,\ldots,V_2}_{s_2},\ldots,\underbrace{V_n,\ldots,V_n}_{s_n}), \qquad (11)$$

where V_i are the vertices of the domain simplex.

The blossom of Wachspress coordinates for a regular n-gon is given as

$$q(p_1,\ldots,p_{n-2})_i = \frac{1}{(n-2)!} \cdot \sum_{\pi \in \Pi(n-2)} \prod_{\substack{k=1 \\ j \neq i-1,i}}^{n-2} D_j(p_{\pi_k}), \qquad (12)$$

where $\Pi(n-2)$ is the set of permutations of $\{1,\ldots,n-2\}$, and in the product k goes from 1 to $n-2$ while j goes from 1 to n skipping $i-1$ and i. Now Eq. (11) can be used to compute the coordinates of W_n.

The exact position and rotation of the n-sided polygon affect the quality of the generated control net. A good choice is to use cyclic polygons on the circle that has its origin at $(0.5,0.5)$, and a radius of 0.5.

4.1.2. Simplex composition

Now that all three of the Bézier simplexes are well-defined, we can turn our attention to their composition. Let F and G denote Bézier simplexes, and $H = F \circ G$. Then

$$\phi(H) = (n_G, d_F \cdot d_G, \delta_F), \qquad (13)$$

and the control points are given by

$$P_{\mathbf{s}}^H = \sum_{\substack{|\mathbf{s}_1|+|\mathbf{s}_2|+\cdots+|\mathbf{s}_{d_F}|=d_G \\ \mathbf{s}_1+\mathbf{s}_2+\cdots+\mathbf{s}_{d_F}=\mathbf{s}}} \frac{\binom{d_G}{\mathbf{s}_1}\cdots\binom{d_G}{\mathbf{s}_{d_F}}}{\binom{d_F \cdot d_G}{\mathbf{s}}} \Delta_{\mathbf{0}_{n_F}}(P_{\mathbf{s}_1}^G,\ldots,P_{\mathbf{s}_{d_F}}^G), \qquad (14)$$

Algorithm 1 Efficient composition of two Bézier simplexes.

COMPOSE(F, G):
$\quad P_{\mathbf{s}}^H \leftarrow \mathbf{0}_{\delta_H}$ for all $\mathbf{s} \in L_{n_H,d_H}$
$\quad P_{\mathbf{s}}^{\hat{F},1} \leftarrow P_{\mathbf{s}}^F$ for all $\mathbf{s} \in L_{n_F,d_F}$
$\quad$ **fn** BLOSSOM(k, p):
$\quad\quad$ **for** $\mathbf{s}$ in L_{n_F,d_F-k}:
$\quad\quad\quad P_{\mathbf{s}}^{\hat{F},k+1} \leftarrow \sum_{i=1}^{n_F} p^i P_{\mathbf{s}+\mathbf{e}_i}^{\hat{F},k}$
$\quad$ **fn** REC(k, $\mathbf{s}_{\min}$, $\mathbf{s}_\Sigma$, c, μ):
$\quad\quad$ **if** $k = d_F$:
$\quad\quad\quad P_{\mathbf{s}_\Sigma}^H \leftarrow P_{\mathbf{s}_\Sigma}^H + c \cdot P_{\mathbf{0}_{n_F}}^{\hat{F},d_F+1}$
$\quad\quad$ **else**:
$\quad\quad\quad \mathbf{s} \leftarrow \mathbf{s}_{\min}$
$\quad\quad\quad$ **while** $\mathbf{s} \neq \emptyset$:
$\quad\quad\quad\quad$ BLOSSOM($k+1$, $P_{\mathbf{s}}^G$)
$\quad\quad\quad\quad$ REC($k+1$, $\mathbf{s}$, $\mathbf{s}_\Sigma + \mathbf{s}$, $c \cdot \binom{d_G}{\mathbf{s}}/\mu$, $\mu+1$)
$\quad\quad\quad\quad \mu \leftarrow 1$
$\quad\quad\quad\quad \mathbf{s} \leftarrow$ NEXT($\mathbf{s}$)
$\quad$ REC(0, $\mathbf{0}_{n_G} + \mathbf{e}_{n_G} \cdot d_G$, $\mathbf{0}_{n_G}$, $d_F!$, 1)
$\quad P_{\mathbf{s}}^H \leftarrow P_{\mathbf{s}}^H / \binom{d_H}{\mathbf{s}}$ for all $s \in L_{n_H,d_H}$
$\quad$ **return** P^H

---

where $\mathbf{0}_{n_F}$ is a multi-index or vector of length n_F consisting entirely of zeros, and

$$\Delta_{\mathbf{s}}(p_1,\ldots,p_k) = \begin{cases} \sum_{i=1}^{n_F} p_1^i P_{\mathbf{s}+\mathbf{e}_i}^F & \text{if } k=1, \\ \sum_{i=1}^{n_F} p_k^i \Delta_{\mathbf{s}+\mathbf{e}_i}(p_1,\ldots,p_{k-1}), & \text{otherwise.} \end{cases} \qquad (15)$$

Here p_k^i is the i-th coordinate of p_k, and $\mathbf{s} + \mathbf{e}_i$ is the multi-index $\mathbf{s}$ with the i-th position increased by one.

4.1.3. Efficient composition algorithm

The equations shown in the previous section, when implemented naïvely, are highly inefficient. DeRose et al.[2] show an algorithm for computing it in a less computationally intensive manner. It is based on caching the results of the recursive calls to Δ, and using the fact that Δ is symmetric in its arguments.

Algorithm 1 shows the pseudocode. Here NEXT($\mathbf{s}$) is the next multi-index in a lexicographical ordering, e.g. $0021 \rightarrow 0030 \rightarrow 0102 \rightarrow 0111 \rightarrow \ldots$, returning $\emptyset$ after the last element (3000 in the example). Since Δ is now computed only for this lexicographical permutation of its arguments, its result is multiplied by $d_F!$ divided by the factorials of the argument multiplicities (built up by the variable μ).

The paper suggests the use of arrays for the control points, indexed by the lexicographical order of the multi-index labels. A conceptually simpler alternative is the use of a dictionary data structure (a hash table) that maps control point positions to their labels. Our experiments show that this is actually even faster, as the hash function is cheaper to compute than the lexicographical index.

4.1.4. Change of coordinates

The homogenized control points computed above have the form $(wx, wy, wz, w(1 - x - y - z))$. To convert these to the usual homogeneous coordinates (wx, wy, wz, w), we just need to replace the last coordinate by the sum of all coordinates.

4.2. Conversion to tensor product form

A (rational or non-rational) S-patch of depth d has the tensor product Bézier patch form

$$\hat{S}(u, v) = \sum_{i=0}^{d} \sum_{j=0}^{d} C_{ij} B_i^d(u) B_j^d(v) \qquad (16)$$

with the control points

$$C_{ij} = \sum_{\substack{\mathbf{s} \\ s_2+s_3=i \\ s_3+s_4=j}} \frac{\binom{d}{\mathbf{s}}}{\binom{d}{i}\binom{d}{j}} P_{\mathbf{s}}. \qquad (17)$$

5. Example

Figure 1 shows a 5-sided S-patch of depth 5, converted into a 15×15-degree tensor product rational Bézier patch. The example model was generated by the ribbon-based algorithm of the author.[7]

6. Discussion

We have seen how S-patches are transformed into trimmed Bézier patches. Let us now look at some practical issues.

6.1. Efficiency

The algorithm in Section 4.1.3 is still computationally demanding. Converting a 5-sided S-patch of depth 8 took more than 5 minutes on a 2.8GHz processor. There is a much faster approach – see the upcoming paper of the author.[8]

6.2. Triangles

Three-sided S-patches, i.e., Bézier triangles, are converted into simple polynomial patches, since computing the barycentric coordinates do not involve rational polynomials. Note however, that there are alternative methods for the quadrilateral transformation of a triangular patch, see e.g. Warren's domain deformation method.[10]
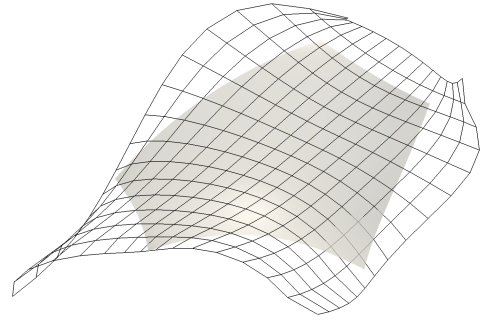
6.3. Control net quality

One issue with the conversion is that the quadrilateral control grid may have outlier control points or spikes near the corners. This is because the denominator of Wachspress coordinates vanish on a circle around the domain, and these singularities undermine the stability of these areas.
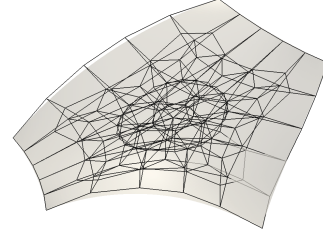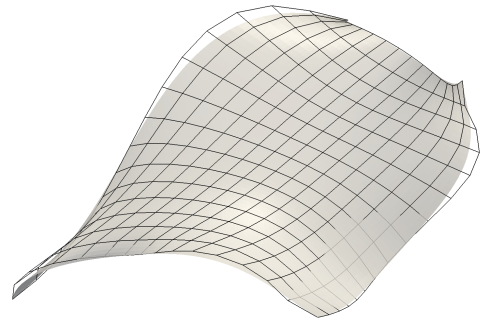
(a) S-patch control network
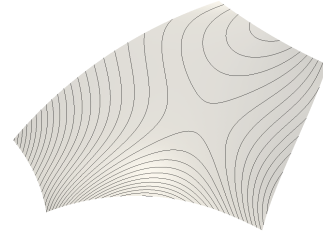
(b) Contouring

(c) Trimmed patch with control net

(d) Full tensor product patch

Figure 1: A 5-sided S-patch of depth 5.

Conclusion

The S-patch representation of multi-sided surfaces can be an important asset in the designer's toolbox. It can be used to fill holes or create smooth vertex blends, and then it can be exported to CAD/CAM systems as a trimmed Bézier patch, without losing precision, thus making it possible to create perfectly watertight models. In this paper we have reviewed the steps required for the quadrilateral conversion, and discussed some related questions.

Acknowledgements

References

1. Tony D. DeRose. Composing Bézier simplexes. *ACM Transactions on Graphics*, 7(3):198–221, 1988.

2. Tony D. DeRose, Ronald N. Goldman, Hans Hagen, and Stephen Mann. Functional composition algorithms via blossoming. *ACM Transactions on Graphics*, 12(2):113–135, 1993.

3. Kai Hormann and Michael S. Floater. Mean value coordinates for arbitrary planar polygons. *Transactions on Graphics*, 25(4):1424–1441, 2006.

4. Charles T. Loop and Tony D. DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, 1989.

5. Les A. Piegl and Wayne Tiller. Filling *n*-sided regions with NURBS patches. *The Visual Computer*, 15(2):77–89, 1999.

6. Lyle Ramshaw. *Blossoming: A connect-the-dots approach to splines*. Digital Equipment Corporation, Palo Alto, 1987.

7. Péter Salvi. G^1 hole filling with S-patches made easy. In *Proceedings of the 12th Conference of the Hungarian Association for Image Processing and Pattern Recognition (submitted)*, 2019.

8. Péter Salvi, Alyn Rockwood, and Tamás Várady. Exact conversion of multi-sided patches into trimmed Bézier surfaces. *Computer Aided Geometric Design (submitted)*, 2019.

9. Márton Vaitkus and Tamás Várady. Parameterizing and extending trimmed regions for tensor-product surface fitting. *Computer-Aided Design*, 104:125–140, 2018.

10. Joe Warren. Creating multisided rational Bézier surfaces using base points. *ACM Transactions on Graphics*, 11(2):127–139, 1992.