Better Pasting via Quasi-Interpolation

Blair Conrad and Stephen Mann

Abstract. Surface pasting is a hierarchical modeling technique that places feature surface atop base surface with approximate continuity. In this paper, we describe the use of quasi-interpolation to set the boundary control points of the pasted feature. To achieve interactive performance for surface pasting, we modify the coefficients and weights of quasi-interpolation and devise an efficient sampling scheme. The new surface pasting technique is actually faster than the original surface pasting method when using the same number of feature control points, with a much lower discontinuity between the feature and base surface.

§1. Introduction

Spline curves and surfaces are used in many areas of computer graphics and computer aided geometric design. In particular, tensor product B-spline surfaces are commonly used in modeling and computer animation because they have many attractive properties, such as a compact representation and adjustable levels of internal continuity [7].

Frequently, the user of a piece of modeling or animation software will want to add a region of local detail to a B-spline tensor product surface, but the knot structure will be too coarse to allow the fine-grained control that the user desires. Traditional methods of increasing the complexity of the surface include inserting knots using either Boehm's algorithm [3] or the Oslo algorithm [6]. The insertion of a knot into either of a surface's knot vectors causes an entire row or column of subpatches to be split — rather than increasing the number of subpatches locally, extra subpatches are created across the width or breadth of the surface.

Forsey and Bartels [8] developed hierarchical B-splines to allow the user to add local detail to a tensor product B-spline surface while maintaining B-spline continuity. Hierarchical B-splines suffer from several drawbacks: the regions of added detail must remain parametrically aligned with the base, and it is impractical to slide the features or to maintain a library of overlays that may be added to a base surface. Wavelets [10] can be used in a similar

Curve and Surface Design: Saint-Malo 1999
Pierre-Jean Laurent, Paul Sablonnière, and Larry L. Schumaker (eds.), pp. 27–36.

ISBN 0-8265-1356-5.

hierarchical modeling technique, but they also require the detail regions to be parametrically aligned with the base.

Displacement mapping is another technique for adding local detail to a surface while adding as few extra control points as possible. Every point on the feature is defined by a displacement vector relative to a certain point in a reference plane. To evaluate the surface at the detail, you add the displacement vector of the feature to the corresponding point on the base surface. The greatest disadvantage to using displacement mapped surfaces is the computational cost. Multiple surface evaluations must be used to determine each point to be rendered on the composite surface.

Surface pasting, suggested by Forsey and Bartels [2], is a generalization of hierarchical B-splines that was intended to combine the flexibility of displacement mapped surfaces with the speed of evaluation enjoyed by hierarchical B-splines. This technique has the flexibility of displacement mapping, but it is much cheaper since only the control points of the feature must be mapped, rather than the larger number of surface points to be rendered. This combination of flexibility and speed has drawn the attention of the modeling industry to surface pasting — recent versions of *Houdini*, a commercial animation tool produced by Side Effects Software, have included support for surface pasting.

However, surface pasting is only an approximation, and as such it does not have the same continuity properties as displacement mapping or hierarchical B-splines. In general, there is no guaranteed continuity between the feature and the base surfaces. By inserting knots into the feature surface, the error between the feature boundary and the base surface can be reduced to any desired tolerance, but many knot insertions may be required to get the desired approximation, and the resulting additional control points in the feature dramatically increase the cost of the pasting operation.

In this paper, we suggest altering the surface pasting technique to improve the approximate continuity between the feature and base surfaces by using quasi-interpolation to set one or more of the outer rings of feature control points.

§2. Surface Pasting

Since the work in this paper is an extension of surface pasting, we will give a quick review of how pasting works. For details on standard surface pasting, see any of the earlier works on the subject [1,2].

The pasting process is illustrated in Figure 1. Surface pasting combines a base surface and a feature surface, each of which is in tensor-product B-spline form. The feature's control points are adjusted so that the boundary of the pasted feature lies on or near the base surface, and the shape of the pasted feature reflects the original shape of the feature imposed as a displacement of the base surface.

To map the feature's control points, the feature's domain is embedded in the feature's range (upper left of Figure 1). Next, a local coordinate frame $\mathcal{F}_{i,j} = \{u_{i,j}, v_{i,j}, w_{i,j}, \mathcal{O}_{i,j}\}$ is constructed for each feature control point $P_{i,j}$

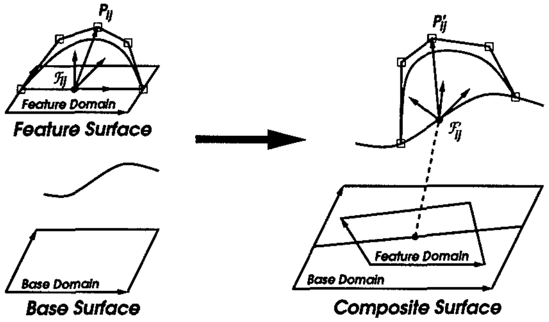

Fig. 1. Surface Pasting.

where the origin $\mathcal{O}_{i,j}$ of each frame is the Greville point corresponding to $P_{i,j}$, two of the frame's basis vectors are the parametric domain directions and the third basis vector is the direction perpendicular to the domain. Each control point $P_{i,j}$ is then expressed relative to its local coordinate frame $\mathcal{F}_{i,j}$ as $P_{i,j} = \alpha u_{i,j} + \beta v_{i,j} + \gamma w_{i,j} + \mathcal{O}_{i,j}$.

Next, we associate the feature's domain with a region in the base's domain (right half of Figure 1). This gives the location on the base surface where we will place the feature. We now map each coordinate frame $\mathcal{F}_{i,j}$ onto the base surface, giving a new coordinate frame $\mathcal{F}'_{i,j} = \{u'_{i,j}, v'_{i,j}, w'_{i,j}, \mathcal{O}'_{i,j}\}$ whose origin $\mathcal{O}'_{i,j}$ is the evaluation of the base surface at $\mathcal{O}_{i,j}$, and two of its basis vectors lie in the tangent plane of the base surface at that point, the third being perpendicular to the tangent plane. We then use the coordinates of each feature control point $P_{i,j}$ relative to $\mathcal{F}_{i,j}$ as coordinates of the elements of the frame $\mathcal{F}'_{i,j}$. This gives us the location of the pasted feature control point, $P'_{i,j} = \alpha u'_{i,j} + \beta v'_{i,j} + \gamma w'_{i,j} + \mathcal{O}'_{i,j}$.

§3. Feature Boundaries

With standard surface pasting, the boundary control points of the pasted feature surface (Figure 2) are conventionally placed on the base surface, resulting in an almost C^0 join. Likewise, the second layer of feature control points are also conventionally located on the base surface, giving an almost C^1 join. By inserting knots into the feature surface, the discontinuity between the feature and the base can be made as small as desired.

One disadvantage of using knot insertion to decrease the discontinuity between the feature surface and the base surface is that it greatly increases the number of control points in the feature surface. In addition to increasing the storage requirements of the feature surface, this increases the computational cost for interactive modeling, as each feature control point needs to be mapped with the pasting process described in the previous section.

The goal of this work is to find better settings of the boundary layers of the feature control points to minimize the C^0 and C^1 discontinuities. Thus, we will find better settings of the control points illustrated in Figure 2. In this figure,

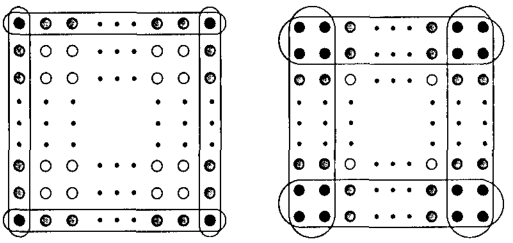

Fig. 2. Approximate C^0 boundary points and C^1 boundary points.

the circled points are set using quasi-interpolation: the black control points will be set to interpolate position and derivative information sampled from the base surface; the gray control points will be set using quasi-interpolation; and the remaining control points will be set using the standard surface pasting process.

§4. Quasi-Interpolation

Quasi-interpolation is a spline approximation technique developed by de Boor and Fix [4]. Their method, given a function f defined over a region of $\mathcal{R}$ and a partition π of $\mathcal{R}$, constructs a degree m spline, $F_\pi f$, that approximates f. $F_\pi f$ is called the quasi-interpolant of f. The quasi-interpolant is a local approximation in that its value at u depends only on the values of f in a small neighborhood around u, it reproduces polynomials of degree m or less, and it provides a high order approximation to f, with $|F_\pi f - f|$ being $\mathcal{O}(|\pi|^{m+1})$.

Lyche and Schumaker [9] describe a family of quasi-interpolation operators Q which include the one of de Boor and Fix. They take

$$Qf = \sum_{i=0}^{M} \lambda_i f B_i^m, \tag{1}$$

where the $\{B_i^m\}_{i=0}^M$ are the B-splines and $\{\lambda_i\}_{i=0}^M$ are linear functionals. The linear functionals can be chosen so that Q is applicable to a wide class of functions, Q is local, and Qf approximates smooth functions with a high order of accuracy. Then the operator Q can be applied to a real-valued function f to produce a B-spline curve Qf that approximates f.

The quasi-interpolant Qf approximates a smooth function f with a high order of accuracy because Q is constructed specifically to reproduce polynomials. The Lyche-Schumaker quasi-interpolants approximate smooth functions with accuracy of up to $\mathcal{O}(h^{m+1})$ for a function with a sufficiently high degree of continuity. In addition, quasi-interpolants can be constructed using linear functionals that can be quickly evaluated, resulting in a fast approximation operator. The combination of high accuracy and speed make quasi-interpolation a fitting tool to improve the approximate continuity around the boundary of pasted features.

The quasi-interpolant Qf is in B-spline form. Each control point of Qf is a weighted sum of linear functionals applied to f:

$$P_i = \sum_{j=0}^{m} \alpha_{i,j} \lambda_{i,j} f.$$

One particular choice of the Lyche-Schumaker quasi-interpolants corresponds to

$$\lambda_{i,j} f = [\tau_{i,0}, \tau_{i,1}, ..., \tau_{i,j}] f,$$

where the τ's are locations at which to evaluate f, and $\alpha_{i,j}$ is the blossom [11] of

$$p_{i,j}(u) = (u - \tau_{i,0})(u - \tau_{i,1}) \cdots (u - \tau_{i,j-1})$$

evaluated at $u_i, ..., u_{i+m-1}$, where the u_i are the knots of Qf. This results in a quasi-interpolant where $Qf = f$ when f is a degree m or lower polynomial and if $Qf \neq f$, then the error has the best possible order.

§5. Modified Quasi-interpolant

The Lyche-Schumaker quasi-interpolant uses cheap coefficients and expensive linear functionals. For interactive modeling using surface pasting, the linear functionals are recalculated frequently (i.e., when we slide the feature across the base) and the coefficients are recalculated less frequently (only when we perform knot insertion to reduce the C^0 discontinuity). Thus, to match the quasi-interpolant to our application, we made new, cheaper linear functionals,

$$\lambda_{i,j} f = f(\tau_{i,j}).$$

This choice of $\lambda_{i,j}$ results in more expensive $\alpha_{i,j}$, which are now the blossom of

$$p_{i,j}(u) = \prod_{k \neq j} \frac{u - \tau_{i,k}}{\tau_{i,j} - \tau_{i,k}} \tag{2}$$

evaluated at $u_i, ..., u_{i+m-1}$, where the u_i are the knots of Qf. Note that the Lyche-Schumaker linear functionals require computing a divided difference of the samples while ours do not, but the denominator term of (2) makes our α coefficients more expensive than theirs.

To use the quasi-interpolant on all four boundaries of a feature, the ends of the connecting boundary curves must match. This is most easily achieved by modifying Q to reproduce position and (to use the quasi-interpolant on the cross-boundary derivatives) d derivatives at its endpoints.

Thus, we devised a quasi-interpolant that we call Q^d, where the linear functionals for the control points near the ends of the curves are derivatives of the original curve at the endpoints. This quasi-interpolant is actually a special case of the Lyche-Schumaker quasi-interpolant. The Q^d quasi-interpolant requires different λ's and p's at the endpoints, which in turn require new α's.

For $0 \leq i \leq M$, and $0 \leq j \leq m$, we define Q^d as

$$Q^d f = \sum_{i=0}^{M} P_i B_i^m = \sum_{i=0}^{M} \Big(\sum_{j=0}^{m} \alpha_{i,j} \lambda_{i,j} f \Big) B_i^m, \tag{3}$$

where

$$\lambda_{i,j} f = \begin{cases} f(\tau_{i,j}), & \text{if } d < i < M - d \\ D^j f(u_{m-1}), & \text{if } i \leq d \text{ and } j \leq i \\ D^j f(u_M), & \text{if } i \geq M - d \text{ and } j \leq M - i \\ 0, & \text{otherwise} \end{cases}$$

$$p_{i,j}(u) = \begin{cases} \displaystyle\prod_{k \neq j} \frac{u - \tau_{i,k}}{\tau_{i,j} - \tau_{i,k}}, & \text{if } \; d < i < M - d \\ \dfrac{(u - u_{m-1})^j}{j!}, & \text{if } \; i \leq d \; \text{ and } \; j \leq i \\ \dfrac{(u - u_M)^j}{j!}, & \text{if } \; i \geq M - d \; \text{ and } j \leq M - i \end{cases}$$

$$\alpha_{i,j} = \begin{cases} 0, & \text{if } \; i \leq d \; \text{ and } \; j > i \\ 0, & \text{if } \; i \geq M - d \; \text{ and } \; j > M - i \\ p_{i,j}(u_i, ..., u_{i+m-1}), & \text{otherwise} \end{cases}$$

where the $\tau_{i,j}$s are as defined in the next section. Note that for $d < i < M - d$, the $p_{i,j}$s are merely Lagrange polynomials. This quasi-interpolant Q^d interpolates the position and derivatives at the endpoints and has optimal error bounds as discussed by Conrad [5].

Sampling discipline

The above equations require us to sample the function f to be approximated at some values $\tau_{i,j}$, and (3) suggests that we require about $m + 1$ samples of the base surface for each boundary control point of the feature. Since the quasi-interpolant error bounds hold for most choices of τ's, we managed to reduce the number of samples of the base surface by selecting the $\tau_{i,j}$ in the following way:

- Place the first d and last d control points in groups of their own; divide the remaining control points into groups as indicated by the pseudo-code of Figure 3; each group k will contain control points $P_{N_k}, ..., P_{N_k+n_k-1}$, where $N_k = \sum_{\ell=0}^{k-1} n_\ell$ and n_k is the number of control points in group k;
- Let γ_i be the Greville point associated with control point P_i; then for each group k, choose the interval from which to sample as
$$[a_k, b_k] = [(\gamma_{N_k-1} + \gamma_{N_k})/2, (\gamma_{N_{k+1}-1} + \gamma_{N_{k+1}})/2];$$
- Set $\tau_{i,j} = a_k + j(b_k - a_k)/m$, for $i = N_k, ..., N_{k+1} - 1$ and $j = 0, ..., m$;
- Sample f at the $\tau_{i,j}$; note that you can share the first and last samples between groups,

```
$I := M - 2 * d + 1$
$numFullGroups := \lfloor I/m \rfloor$
$L := I - m * numFullGroups$
if $L = 0$
    make $numFullGroups$ groups of $m$
else if $numFullGroups \equiv 0 \pmod 2$
    put leftovers in middle group; all others have $m$
else there are an odd number of full groups
    if $L \equiv 0 \pmod 2$
        first and last groups contain $L/2$ each; all others
    else
        move one control vertex from middle full group to the leftovers
        the first and last groups each contain $(L + 1)/2$,
        the middle group contains $m - 1$, and the others contain $m$
```

Fig. 3. Pseudo-code for grouping.

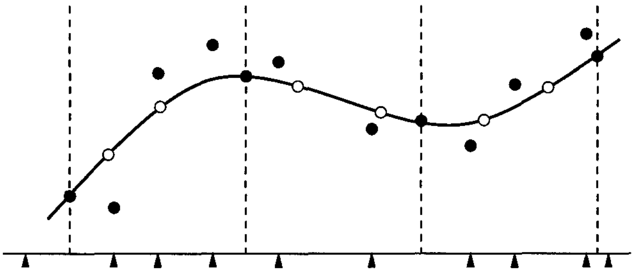

Fig. 4. Sampling discipline for quasi-interpolated surface pasting.

Pseudo-code for our grouping scheme appears in Figure 3, and the sampling is illustrated in Figure 4. The white and gray points are samples of the base surface; the black points represent the control points for the quasi-interpolant (the actual quasi-interpolant control points will lie much closer to the curve; we have moved them away from their actual position for illustrative purposes); the dotted lines indicate the boundaries between the groups, with the gray points being shared by two groups; and the triangles indicate the position of the Greville points.

This sampling method reduces the number of base surface evaluations to about one sample per control point.

Quasi-interpolated surface pasting

Ideally, we would be able to integrate quasi-interpolation into surface pasting by running four independent quasi-interpolations to set the boundary control points, then four more independent quasi-interpolations to set the second layer of control points, and so on. However, as seen in Figure 2, the boundary layers of control points overlap, and while the gray control points in this figure can be set independently, the black ones are shared by two boundaries. To build a tensor-product patch, we need the two quasi-interpolation solutions that compute a group of black points to give these points the same settings.

Our solution is to sample the base surface at the corners of the embedded feature domain for position and as many derivatives as we are trying to approximate along the boundary. In our implementation, we were only trying to achieve approximate C^1 continuity, so we sampled for position and first derivatives (including mixed partial derivatives).

Next, we used the Q^d operator to set $d+1$ outermost rings of control points. With $d=1$ (the right side of Figure 2), this means that we make eight applications of the Q^1 operator, four for the outer layer of control points, and four for the next layer of control points. In each application, the Q^1 operator interpolates four values (those indicated by the black points) and quasi-interpolates the remaining values.

Note that while we quasi-interpolate points to compute the first layer of control points, to compute the second layer of control points, we first quasi-interpolate cross-boundary derivative vectors. The vector coefficients of this second quasi-interpolant are then scaled and added to the first layer control points to produce the second layer of control points.

We set the remaining interior control points, which do not affect the cross-boundary derivatives, using the standard pasting method.

Results

We integrated the Q^0 and Q^1 quasi-interpolants into surface pasting. In our particular implementation, this results in a cubic quasi-interpolant being used to approximate a C^2 curve. Theoretically, the quasi-interpolant described in this paper should have $\mathcal{O}(h^3)$ order of accuracy, which matched our empirical tests. This is an improvement over standard pasting, which empirically had $\mathcal{O}(h^2)$ accuracy on the same data.

We achieved a similar improvement for the cross-boundary derivatives (improving from $\mathcal{O}(h^1)$ to $\mathcal{O}(h^2)$), and the Q^1 method has an additional advantage over standard pasting: for standard pasting, you insert knots in one parametric direction to decrease the C^0 discontinuity, and insert knots in the other parametric direction to decrease the C^1 discontinuity. With the Q^1 method, inserting knots in one parametric direction will decrease both the C^0 and C^1 discontinuity along the boundary.

Visually, the quasi-interpolation surface pasting method gives significant improvements in the approximate C^d continuity around the feature boundaries. An example appears in Figure 5. In all three images, the feature surface has the same number of control points. The large gap that appears in the standard surface pasting example has almost disappeared with the Q^0 and Q^1 pasted surfaces, but a "corner" has appeared in the Q^0 surface. However, Q^1 pasted surfaces have better approximate continuity (both C^0 and C^1) and are cheaper to paste than standard pasting, assuming we use feature surfaces having the same number of control points for both standard and Q^1 pasting.

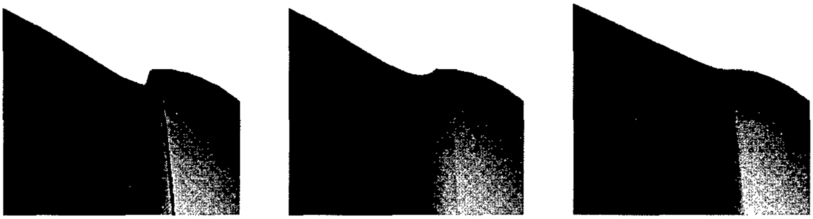

Fig. 5. Standard pasted feature, Q^0 pasting, and Q^1 pasting.

This computational gain for Q^1 pasting is a result of quasi-interpolating the cross-boundary vector field when computing the second layer of control points. We obtain the vectors to quasi-interpolate when we evaluate the base surface for quasi-interpolating the boundary curve. Thus, we avoid having to evaluate the base surface when computing the second layer of control points.

The biggest computational gain, however, comes from needing fewer control points for Q^1 feature surfaces than for standard pasting surfaces to achieve the same tolerance along the boundaries. In particular, in the examples we tested, the Q^1 method required one third the number of boundary points as the standard surface pasting method to achieve the same error along the boundary. This results in one-ninth the total number of control points in the feature, resulting in roughly a factor of ten speedup for quasi-interpolated surface pasting over standard surface pasting.

Conclusion

Surface pasting is a flexible modeling technique that allows the creation of multi-resolution surfaces by hierarchically composing tensor product B-spline surfaces. The resulting composite surface is only approximately C^0 and approximately C^1. In this paper, we developed quasi-interpolation operators to reduce the C^0 and C^1 discontinuities between the feature and base surfaces. By reducing the C^0 and C^1 discontinuities, we can use fewer control points in the representation of the feature surfaces, resulting in a significant speedup of the pasting technique.

We implemented the technique described in this paper for pasted surfaces of arbitrary degree, and tested the method on cubic and quartic surfaces. While we have only reduced C^0 and C^1 errors, the extension to higher levels of approximate continuity is straightforward, but calculating the coefficients that weight the linear functionals will be more complicated.

Acknowledgments. Many thanks to Richard Bartels and Kirk Haller, who have provided many valuable insights during the course of our research. We also thank Tom Lyche, who originally suggested the topic of this research. Financial support for this research was provided by NSERC, CITO, and the University of Waterloo.

References

1. Barghiel, C., R. Bartels, and D. Forsey, Pasting spline surfaces, in *Mathematical Methods for Curves and Surfaces,* Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds), Vanderbilt University Press, Nashville & London, 1995, 31-40.
2. Bartels, R. and D. Forsey, Spline overlay surfaces, University of Waterloo Report CS-92-08, 1991.
3. Boehm, Wolfgang, Inserting new knots into B-spline curves, Computer-Aided Design **12(4)** (1980), 199–201.
4. Boor, C. de and G. J. Fix, Spline approximation by quasi-interpolants, J. Approx. Theory **8** (1973), 19–45.
5. Conrad, B., Better pasting through quasi-interpolation, University of Waterloo Report CS-99-14, 1999.
6. Cohen, E., T. Lyche, and R. Riesenfeld, Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, Comp. Graphics and Image Proc. **14** (1980), 87–111.
7. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design,* Fourth Edition, Academic Press, NY, 1994.
8. Forsey, D. and R. Bartels, Hierarchical B-spline refinement, Computer Graphics **22(4)** (1988), 205–212.
9. Lyche, T. and L. L. Schumaker, Local spline approximation methods, J. Approx. Theory **15** (1975), 294–325.
10. Stollnitz, E., T. DeRose, and D. Salesin, *Wavelets for Computers Graphics,* Academic Press, Boston, 1996.
11. Ramshaw, L., Blossoming: a connect-the-dots approach to splines, Techn. Rep., Digital Systems Research Center, Palo Alto, 1987.

Blair Conrad and Stephen Mann
Computer Science Department
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
`beconrad@cgl.uwaterloo.ca`
`smann@cgl.uwaterloo.ca`