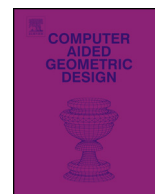




Contents lists available at ScienceDirect

## Computer Aided Geometric Design

www.elsevier.com/locate/cagd



# Conversion of trimmed NURBS surfaces to Catmull–Clark subdivision surfaces

Jingjing Shen<sup>a,\*</sup>, Jiří Kosinka<sup>a</sup>, Malcolm A. Sabin<sup>b</sup>, Neil A. Dodgson<sup>a</sup>

<sup>a</sup> Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom

<sup>b</sup> Numerical Geometry Ltd., 19 John Amner Close, Ely, Cambridge CB6 1DT, United Kingdom

## ARTICLE INFO

### Article history:

Available online xxxx

### Keywords:

Trimmed NURBS surface  
Catmull–Clark subdivision  
Cross field

## ABSTRACT

This paper introduces a novel method to convert trimmed NURBS surfaces to untrimmed subdivision surfaces with Bézier edge conditions. We take a NURBS surface and its trimming curves as input, from this we automatically compute a base mesh, the limit surface of which fits the trimmed NURBS surface to a specified tolerance. We first construct the topology of the base mesh by performing a cross-field based decomposition in parameter space. The number and positions of extraordinary vertices required to represent the trimmed shape can be automatically identified by smoothing a cross field bounded by the parametric trimming curves. After the topology construction, the control point positions in the base mesh are calculated based on the limit stencils of the subdivision scheme and constraints to achieve tangential continuity across the boundary. Our method provides the user with either an editable base mesh or a fine mesh whose limit surface approximates the input within a certain tolerance. By integrating the trimming curve as part of the desired limit surface boundary, our conversion can produce gap-free models. Moreover, since we use tangential continuity across the boundary between adjacent surfaces as constraints, the converted surfaces join with  $G^1$  continuity.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).

## 1. Introduction

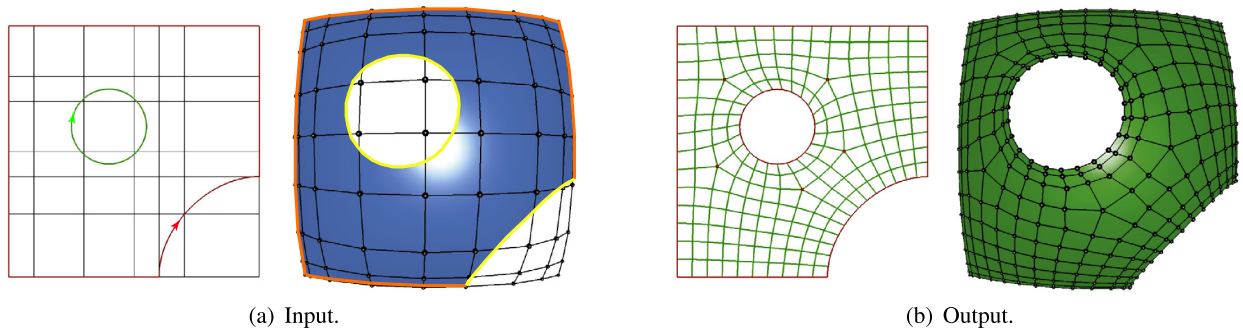
Non-Uniform Rational B-Splines (NURBS) are the standard freeform surface representation in Computer-Aided Design (CAD) applications. Due to their limitation of a strict rectangular topology, trimming is an important operation to create complex objects. However, it introduces unavoidable gaps when stitching two trimmed NURBS patches together (Sederberg et al., 2008). Moreover, as the trimmed NURBS are only visually trimmed by skipping the evaluation of the trimmed part in parameter space (Fig. 1(a)), the rectangular NURBS topology is not altered although the geometric shape has changed. As a consequence, many operations on trimmed shapes may require a time-consuming re-evaluation process of the trimming curves (Farin, 2001), e.g., editing and deformation.

Instead, subdivision, because of its ability to handle arbitrary topology and ease of use, has become an attractive alternative to NURBS, especially for modelling in high-end animation productions (Pixar, DeRose et al., 1998; Stam, 1998). The subdivision representations have the great advantage that if two subdivision surfaces share a boundary edge in base meshes, they both contain exactly that piece of boundary curve. This makes it possible to apply exactly the same trimming curves on two intersecting surfaces and thus provide gap-free models.

\* Corresponding author.

<http://dx.doi.org/10.1016/j.cagd.2014.06.004>

0167-8396/© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).



**Fig. 1.** (a) Input data set. Left: the domain space with black knot lines and oriented trimming loops. Right: the rectangular control mesh of a trimmed NURBS surface. The original NURBS boundary curves marked in orange are referred to as remaining boundary curves (the partly trimmed ones can be obtained by knot insertion at the cut position) and the model-space trimming curves marked in yellow are referred to as  $B$  curves. They compose the final boundary of the target limit surface. (b) Output: the partition in domain space (left) and the resulting base mesh (right) for Catmull–Clark subdivision. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

Although there has been great progress in NURBS-compatible subdivision schemes (Sederberg et al., 1998, 2003; Müller et al., 2006, 2010; Cashman et al., 2009a, 2009b, 2009c), there is little work addressing the conversion algorithms between trimmed NURBS surfaces and subdivision surfaces. The conversion from Catmull–Clark subdivision surfaces to NURBS surfaces can be achieved by viewing each quadrilateral face of the base mesh as a NURBS patch (Loop and Schaefer, 2008). However, conversion in the other direction is nontrivial. This is demanded, for example, in industry scenarios when the designer first converts a Catmull–Clark subdivision surface to NURBS patches and then performs trimming operations to get the desired shape (since trimming NURBS models is well developed in modelling software). The challenge is how to automatically convert the trimmed shape back to the Catmull–Clark representation.

The key motivation behind our work is to develop a framework that automatically converts trimmed NURBS surfaces to subdivision surfaces. The conversion objective is to keep the approximation error within a specified tolerance and maintain the original continuity across the boundary of adjacent surfaces.

### 1.1. Summary of our method

Given a NURBS surface  $S$  and a set of trimming curves  $B$ , the target is to represent the region of  $S$  inside the trimming curves as a Catmull–Clark subdivision surface  $C$  (see Fig. 1). We present a two-step algorithm to achieve the conversion. We first construct the topology of the base quad mesh  $M$  for the desired subdivision surface  $C$  and then calculate the control point positions in  $M$  based on limit stencils of the subdivision scheme.

Inspired by the cross field theory in recent quad remeshing techniques, we automatically identify the EVs (Extraordinary Vertices) required in the mesh  $M$  and decompose the trimmed domain region into quads by computing a boundary aligned cross field in parameter space. The control point positions in  $M$  are chosen to make their corresponding limit points lie exactly on the input NURBS surface. In order to maintain up to  $C^2$  continuity, Bézier edge conditions are applied when constructing the boundary of the base mesh  $M$ .

### 1.2. Advantage of the method

The advantage of our conversion is threefold. First, by specifying the number of refinement steps in the topology construction step, we are able to provide the user with either a fully editable control mesh to manipulate the trimmed surface, or a fine control mesh whose limit surface approximates the input trimmed surface with high accuracy. Second, as an advantage of subdivision representation, we can provide gap-free models after conversion. The converted Catmull–Clark surfaces of two intersecting NURBS surfaces will use the same trimming curve as their shared boundary. Third, the original continuity across the boundary of two trimmed surfaces (tangentially intersecting) can be maintained by setting the near-boundary layers of control points to satisfy Bézier edge conditions on both sides.

## 2. Background and related work

In this section, we explain the related concepts and review prior work in converting trimmed NURBS to other forms of surface representations, trimming subdivision surfaces and quad remeshing.

### 2.1. Trimmed NURBS

A trimmed NURBS surface consists of a regular NURBS patch and a set of trimming curves. In the parameter space, a closed trimming loop identifies a region whose evaluation is skipped. Each trimming operation is defined with NURBS

curve descriptions both in the parameter domain ( $P$  curves) and in model space in  $\mathbb{R}^3$  ( $B$  curves). In this work, we have assumed that both are given and are not concerned with the computation of surface–surface intersections and their projections in parameter space since most commercial geometric modelling programs can do this automatically (Farin et al., 2002; Song and Wang, 2007; Sederberg et al., 2008).

As clearly pointed out by Sederberg et al. (2008), gaps are unavoidable when expressing the intersections of two NURBS surfaces using conventional trimmed NURBS representation. This is because the actual intersection curve in  $\mathbb{R}^3$  is in general not rational. In practice, it is approximated using a B-spline curve, the model-space trimming curve  $B$ . This  $B$  curve is projected back into the parameter space of the two surfaces. These projected curves are then approximated using B-spline curves, which are referred to as the parametric trimming curves  $P$ . The evaluations of the  $P$  curves on the two surfaces are called images of the actual intersection curve. This results in three different curves in  $\mathbb{R}^3$ : the  $B$  curve and the images on two surfaces. None of them is exactly the true intersection. The gap between the surfaces is owing to the two surfaces being trimmed along these slightly different curves in  $\mathbb{R}^3$ . Because the calculations cannot produce exact matches, they are all, instead, performed to within a defined tolerance.

## 2.2. Conversion of trimmed NURBS

Because of the unavoidable gaps in conventional trimmed NURBS models and the complexity in manipulating trimmed surfaces, many approaches have been proposed to convert trimmed NURBS to other forms of surface representations.

One popular way is to decompose and represent trimmed NURBS surfaces with a group of regular surfaces, such as NURBS or Bézier surfaces. Hamann and Tsai (1996) proposed a decomposition method based on a Voronoi diagram in the parameter space. However, their method creates many degenerate patches. Hui and Wu (2005) alleviated this problem to some extent by taking the sharp features of the trimming curves and boundaries into consideration during the construction of the Voronoi diagram. After decomposition in the domain region, both methods then interpolate a group of regular B-spline surfaces, one for each decomposed region. However, as the interpolation is performed locally for each subregion, it is challenging to maintain the continuity of the original surface. Note that these Voronoi based methods split the original corners, which makes it difficult to enforce boundary conditions. They also result in poor global alignment of quads, which makes it inappropriate for the global fitting that we seek.

More recently, Li and Chen (2009) split the trimmed domain in a way that keeps unchanged the regions that are far away from the trimming curves, and then approximated the regions near the trimming curves with  $G^1$  continuity. Schmidt et al. (2012) introduced a similar method to enable isogeometric analysis on trimmed NURBS surfaces. As the local approximation around trimming curves depends on the number and positions of intersections between the trimming curves and knot grids in domain space, both methods are restricted to simple intersecting configurations.

Sederberg et al. (2008) introduced a method to convert each trimmed NURBS to an untrimmed T-spline with approximation error confined to an arbitrarily narrow neighbourhood of trimming curves. However, their method cannot avoid EVs when it exactly matches all of the boundaries.

In contrast to prior conversion work, we propose to convert the trimmed NURBS surfaces to untrimmed subdivision surfaces.

## 2.3. Subdivision schemes

The existence of trimming operations in NURBS surfaces results in the need for careful patch layout and cross-boundary continuity management. This is one reason why subdivision schemes are preferred in the entertainment industry. By repeatedly applying subdivision rules to a base mesh with arbitrary topology, one can model smooth surfaces. For many applications, the limit surface is the preferred choice during the modelling process, rather than the approximated surface after performing the subdivision for several steps. Therefore, for subdivision schemes, the exact evaluation method (the limit stencil) is essential for practical use. Catmull–Clark subdivision (Catmull and Clark, 1978) is the most popular subdivision method used in industry. It can be viewed as midpoint knot insertion on a uniform bicubic B-spline surface (except at extraordinary vertices and faces, whose valency is not equal to four). The evaluation methods proposed by Stam (1998), Lacewell and Burley (2007), Zorin and Kristjansson (2002) are widely used.

As NURBS implementations have the flexibility of higher degrees and non-uniformity, in order to make NURBS-compatible subdivision schemes, Sederberg et al. (1998, 2003) and Müller et al. (2006, 2010) introduced non-uniform subdivision schemes (subdivision surfaces with degree 2 or 3 only) based on the refinement of NURBS with extension to arbitrary topology. NURBS-like knot intervals are assigned to the corresponding vertices or edges of the base mesh. Going beyond degree 3, Cashman et al. (2009b, 2009a, 2009c) proposed a non-uniform subdivision scheme with arbitrarily high degree. Currently, most subdivision implementations in modelling systems are still limited to the uniform bicubic case.

We choose to use the Catmull–Clark subdivision scheme as the conversion target for the following reasons:

- Catmull–Clark is still the standard subdivision method (with crease-rules, DeRose et al., 1998) widely used in modelling, rather than any of the extended non-uniform subdivision methods.

- It has been well studied and the limit stencils are known, although there are additional errors introduced by approximating the input non-uniform boundary curves with uniform curves in the first place.
- Even for the non-uniform case, the approximation is better if the knot intervals are as uniform as possible, especially around EVs, as implied by [Cashman et al. \(2009a\)](#).

#### 2.4. Quadrangulation and cross field

We propose to construct a quadrilateral base mesh  $M$  which can topologically represent the shape of the trimmed surface. This is related to other methods for generating quad meshes.

In recent years, a wealth of quad remeshing techniques has been presented to generate high quality quad meshes from triangular meshes ([Bommes et al., 2012, 2013](#); [Liu et al., 2011](#); [Campen et al., 2012](#); [Zhang et al., 2013](#); [Kobbelt, 2010](#)). The most popular methods are based on periodic parameterisations ([Ray et al., 2006](#)), Morse–Smale theory ([Dong et al., 2006](#); [Huang et al., 2008](#); [Zhang et al., 2010](#)) or mixed-integer optimisation ([Kälberer et al., 2007](#); [Bommes et al., 2009](#)). Most of the parameterisation techniques are guided by vector or cross fields constructed from principal curvature directions or manually specified directions. Cross fields are promising since they can easily capture the positions and types of EVs during quadrangulation.

Our method constructs the topology of the base mesh  $M$  by performing a cross-field based partition in parameter space. The cross field is automatically computed and satisfies the boundary alignment constraints.

A cross field can be viewed as four coupled vector fields. Each cross can be represented as a single unit length vector (red arrows in [Fig. 3](#)) and three other directions by 90 degree rotations of this vector. Due to the 90 degree freedom, smoothing a cross field involves specifying which direction to follow for each cross. Recently, [Zhang et al. \(2006\)](#), [Ray et al. \(2008\)](#), [Palacios and Zhang \(2007\)](#) and [Palacios and Zhang \(2010\)](#) proposed a method to handle N-symmetry direction fields in a linear manner. It separates the direction field into an angle field and an integer period-jump field. Based on this work, [Bommes et al. \(2009, 2013\)](#) solved the quadrangulation problem as a two-step process: cross field generation and global parameterisation. Both can be formulated as mixed integer problems. They proposed an adaptive greedy solver for mixed integer problems. We utilise this solver to compute the cross field in this paper.

#### 2.5. Trimming subdivision surfaces

An alternative approach to ours is to first convert the whole untrimmed NURBS surface to a subdivision surface via degree reduction or elevation, and then introduce trimming into the subdivision method ([Reusche, 2005](#)). While not directly related to our problem, it is useful to be aware of this alternative.

We emphasise that Reusche's work focuses on how to trim a subdivision surface. While in contrast, this paper focuses on how to construct the base mesh for an untrimmed subdivision surface that fits the trimmed NURBS surface. Reusche's method takes the rectangular control mesh of a NURBS surface as the base mesh of an extended subdivision surface (named ESub, published later by [Müller et al., 2006](#)), and attaches the trimming information into the subdivision process. The actual trimming of the faces is done when the complete subdivision is performed and desired level of accuracy is reached. The approximation error can be confined to arbitrary narrow regions around the trimming curves. However, this method cannot guarantee continuity between trimmed patches. More importantly, in the manufacturing industry, when interrogating the surface, instead of performing the subdivision to a finite number of steps, it is preferred to evaluate the limit surface exactly.

### 3. Overview

We propose a novel framework to convert trimmed bicubic NURBS surfaces to Catmull–Clark subdivision surfaces with Bézier edge conditions. The conversion will keep the approximation error within a specified tolerance and maintain tangential continuity across surface boundary. The input and output are illustrated in [Fig. 1](#).

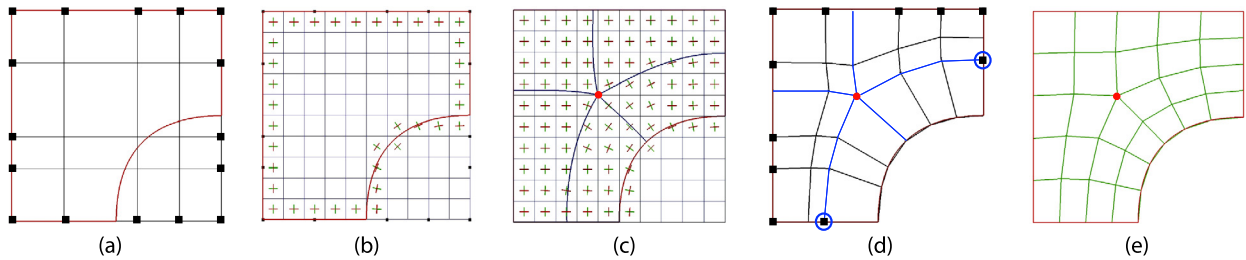
#### Input data set:

- A NURBS surface  $S : (u, v) \rightarrow (x, y, z)$ .
- A set of parametric trimming curves  $P : (t) \rightarrow (u, v)$ .
- A set of model-space trimming curves (also referred to as the intersection curves)  $B : (t) \rightarrow (x, y, z)$ .

Note that  $B(t) \neq S(P(t))$ , but as discussed in [Section 2.1](#), the difference is deemed acceptable and occurs in all commercial CAD systems. Note also that the NURBS representations of  $B$  and  $P$  have the same knot vector and the same degree (achieved by knot insertion and degree elevation if necessary).

#### Target:

- a base mesh  $M$  for Catmull–Clark subdivision surface  $C$  with Bézier edge conditions.



**Fig. 2.** Illustration of quad partition in the domain space. (a) Knot lines (black) and the trimming loop (red). (b) Directional constraints set by the tangent along trimmed domain boundary. (c) Smooth cross field with an EV (marked in red) and traced rays from the EV (blue). (d) Additional rays (black lines) added from original boundary knots (black dots). Note that the intersections of blue rays are snapped to the original boundary knots if they are close enough (marked in blue circles). (e) The final as uniform as possible quad partition ( $M_d$ ). (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

### Main workflow:

**Step 1.** Build the topology of the base mesh  $M$  for Catmull–Clark subdivision surface by performing a cross-field based decomposition in parameter space, as illustrated in Fig. 2.

- Generate a cross field at a selected resolution with boundary constraints in the trimmed domain region.
- Identify the EVs and perform quad partition (referred to as  $M_d$ ) in the trimmed domain region.
- Optimise and refine  $M_d$  to make the quad partition uniform and to keep EVs at least two edges away from the boundary in domain space (required for Bézier edge conditions, see Fig. 4(a)).

**Step 2.** Calculate the control point positions in the base mesh  $M$ .

- Evaluate  $M_d$  on the input NURBS surface to get a mesh consisting of limit points that correspond to the control points (referred to as  $M_l$ ).
- Construct an additional layer of control points along boundaries in base mesh  $M$  for Bézier edge condition (the orange layer in Fig. 4(b)).
- Compute the control points in  $M$  by solving a linear system built from both the limit stencils of the subdivision scheme and the constraints put by tangential continuity requirement across boundary.

The limit stencils of the subdivision scheme specify the linear relation between the limit points and the control points. The tangential continuity constrains control point positions in the near-boundary layer, which can also be formulated as linear equations. Therefore, the control point positions (except the ones at the boundary, which are set up to represent the boundary curves) can be calculated by inverting a linear system.

## 4. Quad partition in trimmed domain region

The domain space of an untrimmed NURBS patch is defined as  $[0, 1] \times [0, 1]$ . The cross field is defined on a 2D grid with user-specified cell size  $s$  inside the trimmed domain region (see Fig. 2(c)). The more tightly curved the trimming curves are, the smaller the cell size should be used;  $s = 0.025$  is small enough for all examples in this paper. As shown in Fig. 2(b–c), after putting the tangent along the boundary of the trimmed domain as directional constraints, a smooth cross field can be computed inside the bounded region (Bommes et al., 2009). EVs with valency 3 (EV3) and 5 (EV5) are automatically identified by local cross field distortions. The coarse quad partition is generated by adding tracing rays from EVs. It is further refined by adding interpolation rays for original boundary knots. Each tracing ray is a set of line segments following the flow of the cross field (blue curves in Fig. 2(c)). The interpolation rays are added proportionally to the knot spacings on boundary (black line segments in Fig. 2(d)). Since the number of boundary knots indicates the number of control points needed to represent the NURBS shape to some extent, refining the partition according to the boundary knots can ensure that we use a reasonable number of control points in the base mesh for the target subdivision surface.

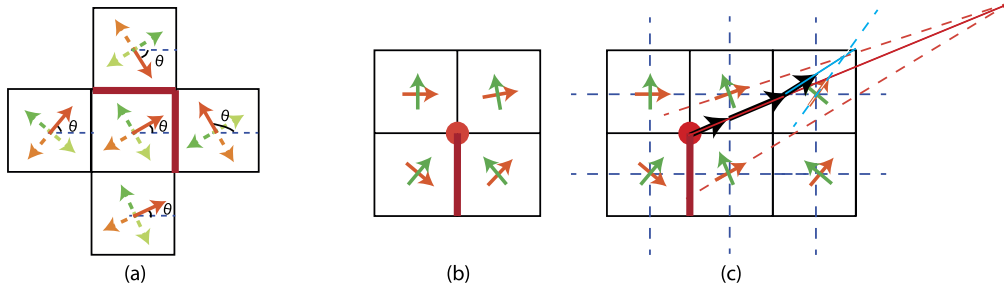
### 4.1. Cross field and EVs

The cross field can be viewed as an angle field plus an integer period-jump field. Each cell  $F_i$  has an angle  $\theta_i$  and each edge  $e_{ij}$  has an integer value  $p_{ij}$  that represents the number of 90 degree jumps between the two cells that share this edge, as illustrated in Fig. 3(a).

After putting directional constraints, a smooth cross field is calculated by minimising total distortion. The distortion between two adjacent cells is defined by the angle difference of the two crosses:

$$D_{ij} = \theta_i - \theta_j + \frac{\pi}{2} p_{ij},$$





**Fig. 3.** (a) Illustration of the cross field on a 2D grid, represented as an angle field in cells ( $\theta$  value) and an integer jump field on edges (non-zero jump marked in red). (b) The cross field around an EV with a non-zero jump edge. (c) Illustration of finding a tracing direction (arrows in black) from the EV (red lines) and then finding a tracing direction from a position inside the cell (blue lines). (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

where  $p_{ij}$  is the integer chosen from  $[0, 1, 2, 3]$  that minimises  $D_{ij}$ .

The smoothing problem is then reduced to a mixed integer problem. The total distortion energy is expressed as:

$$E_{smooth} = \sum_{e_{ij} \in \Gamma} \left( \theta_i - \theta_j + \frac{\pi}{2} p_{ij} \right)^2,$$

where  $\Gamma$  is the set of edges shared by cells in the trimmed domain region.

The cross field index of a vertex can be computed by the sum of the jumps on neighbouring edges:

$$I(v_i) = \sum_{e_{ik} \in N(v_i)} \frac{p_{ik}}{4}.$$

Only the EV has a nonzero index, which is always a multiple of  $1/4$  (Ray et al., 2008). In particular,  $-1/4$  for valency 5 and  $1/4$  for valency 3.

The greedy mixed-integer solver (Bommes et al., 2009) is adopted to solve this minimisation problem. The problem is reduced from quadratic to linear by setting the gradient of the energy to zero:

$$\begin{cases} \frac{\partial E_{smooth}}{\partial \theta_i} = \sum_{e_{ik} \in N(F_i)} 2 \left( \theta_i - \theta_k + \frac{\pi}{2} p_{ik} \right) \doteq 0 \\ \frac{\partial E_{smooth}}{\partial p_{ij}} = \pi \left( \theta_i - \theta_j + \frac{\pi}{2} p_{ij} \right) \doteq 0 \end{cases}$$

The mixed-integer solver first calculates the exact non-integer solution to this problem and then adaptively rounds the integer variables to integers. Therefore,  $\doteq$  is used in the above equations, rather than  $=$ .

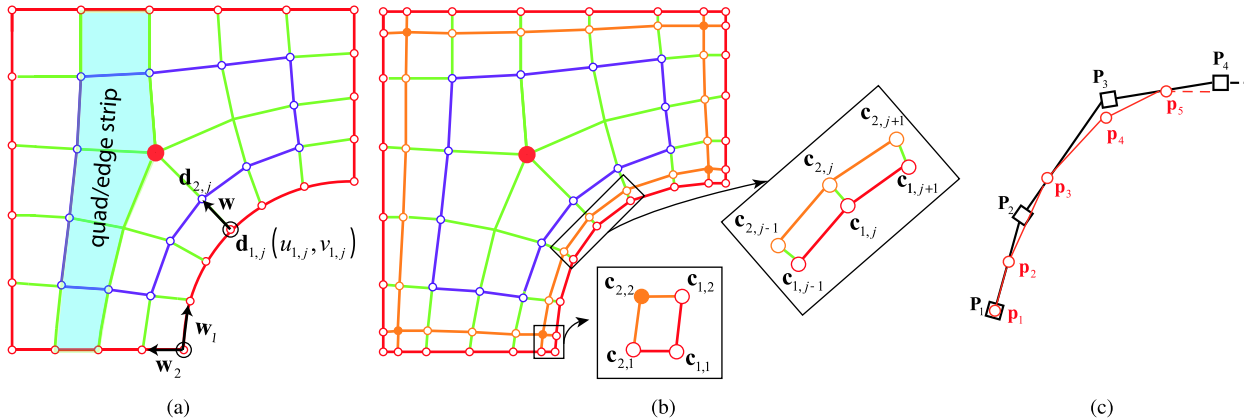
The result is a smooth cross field where the integer jumps define the types and positions of all EVs (see Fig. 2(c) and Fig. 3(b)).

#### 4.2. Tracing rays

The coarse quad partition is generated by tracing rays from EVs following the flow of the cross field (blue curves in Fig. 2(c)). The flow direction at each position in this discrete field is chosen by averaging the flow in its surrounding cells.

- Start from the cell centre, there are two tracing rays (crossed lines shown in orange and green in Fig. 3(b)).
- Start from a point on an edge, the directions of the tracing rays are calculated by taking the average of the centre tracing rays of the cells that share this edge.
- Start from an EV, the number of unique tracing rays is equal to the valency of the EV. These tracing rays are found in the following way (see Fig. 3(c)). For each pair of adjacent cells  $(F_i, F_j)$  around the EV, consider the EV as a point on the edge shared by  $(F_i, F_j)$  and find the two ray candidates. If  $p_{ij}$  is not zero, there is a jump in the following direction between  $F_i$  and  $F_j$ .
- Start from a point inside a cell, choose the closest edge and calculate the tracing direction as if the point was on this edge (the blue line in Fig. 3(c)).

In order to keep the tracing as accurate as possible, middle lines are inserted to the grid (dashed lines in Fig. 3(c)). In each tracing step, after choosing the tracing direction, the ray will follow this direction until it hits a middle line. It will then use this intersection as a new start position for tracing (the black arrows in Fig. 3(c)). The tracing process is repeated



**Fig. 4.** (a) Illustration of layers in the domain partition with EV placement condition satisfied and (b) the corresponding layers in the base mesh with Bézier edge condition. The control points in the first (boundary), second and third are shown in red, orange and purple circles respectively. The control points in the orange layer are introduced by the multiple knots on the boundary, and they have no corresponding knot points in the domain space. The corners of this layer are shown as solid dots. The standard limit stencils are not applicable for these three layers. (c) Illustration of subdivision rules to obtain Bézier end-conditions for a cubic subdivision curve. The control points are shown as black squares and the new points after one step of subdivision are shown as red circles. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

until the ray hits the boundary or becomes close enough to snap with another EV (distance less than the cell size  $s$ ). The tracing rays are snapped to each other if their distance is less than  $s$ . The final ray is a set of line segments with segment length less than  $s$ .

This coarse quad partition in domain space is then refined according to the knot vectors of the boundary curves (black lines in Fig. 2(c)). These refined lines are added by interpolation according to the knot spacings on boundary.

Instead of applying global parameterisation as in MIQ (Mixed Integer Quadrangulation) algorithms (Bommes et al., 2009), we chose to construct the coarse quad partition by adding tracing rays from EVs. This is mainly because the global parameterisation may fail to find a valid solution when a coarse quad mesh is required (specified by edge length, which is an input parameter for MIQ algorithms). Even though this problem is further solved in Bommes et al. (2013), as their input is a dense triangular mesh in 3D, the algorithms to achieve reliable quad meshing are complicated. In contrast, as we work on the cross field on a 2D grid, finding the coarse quad partition by tracing the flow is reliable and much simpler.

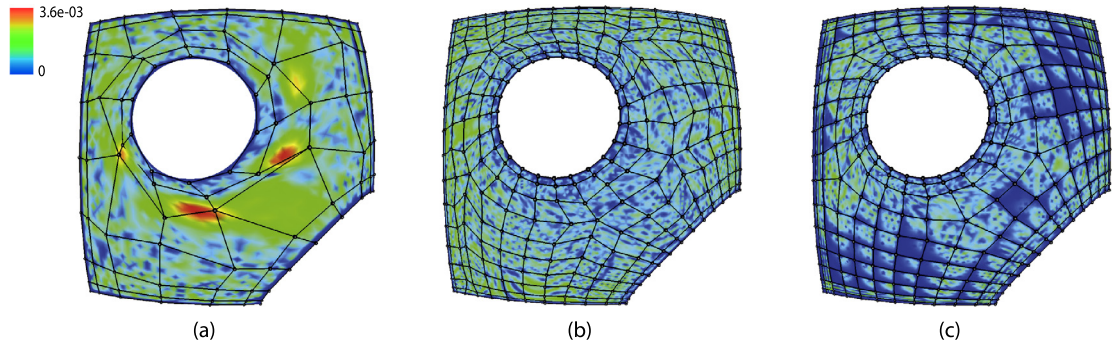
#### 4.3. Uniform knot spacings

Since Catmull–Clark subdivision is a uniform scheme, the knot spacings in the domain partition  $M_d$  should be as uniform as possible. For each boundary edge, this can be done by adjusting the vertices on it to have uniform spacings. If the original input is a non-uniform B-spline surface, the curves  $B$  are non-uniform or the trimming curves  $P$  cut the original boundary arbitrarily, the knot spacings on the boundary are inevitably non-uniform. After moving the positions of original boundary knots (black dots in Fig. 2(a) and (d)) to make the knot spacings uniform for Catmull–Clark subdivision, we cannot reproduce the exact surface boundary curve. In fact, the boundary curve is approximated using the uniform knot vector defined by the boundary vertices of  $M_d$ . This approximation is due to the limitation of Catmull–Clark subdivision scheme.

For interior edges, a simple Laplacian smoothing process is performed. Each vertex is iteratively moved to the average position of its neighbours. The number of steps applied in all our examples is 40.

In order to utilise Bézier edge conditions to guarantee continuity across boundary, the first four layers (counting from boundary, each layer is a chain of vertices) in the base mesh  $M$  should form a topologically rectangular array. Thus in the base mesh, the outermost layer in which EVs can exist is the fourth layer (Fig. 4(b)). As multiple knot lines (multiplicity 4 for bicubic surface) are applied, topologically the EVs should be at least two edges away from the boundary in parameter space (Fig. 4(a)). This EV placement condition is satisfied by two simple steps in our algorithm. First, when generating the cross field, we put more directional constraints in near-boundary cells to keep the singularity away from the boundary. This is done by first computing an offset of the boundary curves inside the trimmed region, and then setting these new crossed cells with tangent direction of the associated offset curve segment. A proper offset distance is twice the grid size in cross field. Second, after making the partition uniform in domain space, a refinement step is applied to make the EVs two edges away from the boundary. Note that this might lead to a denser mesh if the trimming curves are highly curved.

Fig. 4(a) illustrates the definition of a strip: the quads covered in blue form a quad strip. Similarly, the edges inside this strip form an edge strip. Refining the partition  $M_d$  once will insert a middle line in each edge strip. The user can use the number of refinement steps to control the density of the partition and thus achieve the desired approximation accuracy.



**Fig. 5.** The resulting subdivision surfaces with approximation error plots (relative to the bounding box diagonal of the surface): without satisfying the EV placement condition (left), without uniforming step (middle) and with uniforming and EV placement condition satisfied (right).

Fig. 5 illustrates the resulting base meshes and their approximation errors in the following cases. Fig. 5(a) is the result with EVs placed in the third layer of the control mesh. Fig. 5(b) is the result without uniforming the domain partition. Fig. 5(c) is the result with uniforming and EV placement condition satisfied. It has the smallest approximation error.

### 5. Base mesh for Catmull–Clark subdivision

The limit point mesh  $M_l$  is calculated by evaluating  $M_d$  on the input NURBS surface. For Catmull–Clark subdivision scheme, the mesh  $M_l$  and the target base mesh  $M$  have the same topology (except the special orange layer marked in Fig. 4(b), which is introduced by the multiple knot lines on the boundary). Each point in  $M_l$  can be viewed as the limit position of the corresponding control point in  $M$ , formulated by the limit stencils of the subdivision scheme. The control points in base mesh  $M$  can be calculated by solving a square linear system set by

$$\mathbf{V}_l = \mathbf{L}\mathbf{V}_c,$$

where  $\mathbf{V}_l$  is the vector that contains the limit points in  $M_l$ ,  $\mathbf{V}_c$  is the vector that contains the control points in the base mesh  $M$ , and the matrix  $\mathbf{L}$  is built from the coefficients in the equations which link limit point positions to control point positions. For Catmull–Clark subdivision with Bézier edge conditions, the equations are defined by the limit stencils and the tangential continuity requirement across the boundary, which are fully explained in following sections. If the matrix  $\mathbf{L}$  is singular (we have not encountered singular matrices in our examples), a least-squares solution can be adopted (Halstead et al., 1993).

#### 5.1. Catmull–Clark subdivision with Bézier edge conditions

In order to create Bézier edge conditions for the subdivision surface  $C$ , special rules need to be used for boundary (or near boundary) elements, which are naturally derived from multiple knots at the end of the B-splines (Sabin, 2010, Ch. 32). Fig. 4(c) illustrates the special rules at the end of a cubic subdivision curve. To obtain a Catmull–Clark subdivision surface with Bézier edge conditions, these end conditions are applied in a tensor-product manner across all boundary pieces (separated by corners). More specifically, special subdivision stencils are used to calculate the new points in the first four rows during subdivision. Fig. 6 illustrates this process and shows the stencil for  $\mathbf{c}'_{3,4}$ .

#### 5.2. Limit stencils for Catmull–Clark with Bézier edge condition

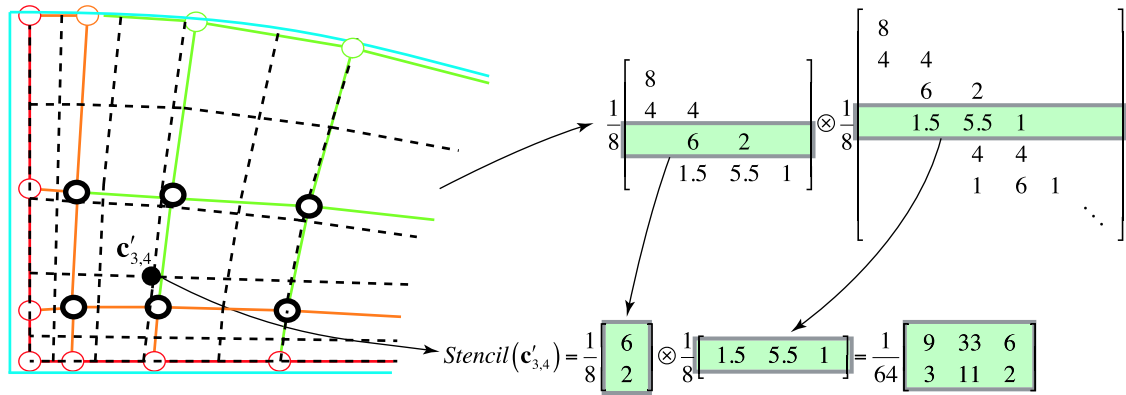
Each subdivision step can be viewed as multiplying the subdivision matrix with the current mesh points once to get a new set of points. The limit surface can be obtained by iteratively multiplying the matrix infinitely. Thus, the standard limit stencil for Catmull–Clark subdivision can be found by analysing the eigenstructure of its subdivision matrix. It corresponds to the dominant left eigenvector of its subdivision matrix (Halstead et al., 1993; Loop et al., 2009). For a control point  $\mathbf{c}$  with arbitrary valency  $n$ , the limit position  $\mathbf{l}$  is then given by (Loop et al., 2009)

$$\mathbf{l}(\mathbf{c}) = \mathbf{l} = \frac{n}{n+5}\mathbf{c} + \frac{4}{n(n+5)}\sum_{k=0}^{n-1}\mathbf{e}_k + \frac{1}{n(n+5)}\sum_{k=0}^{n-1}\mathbf{f}_k,$$

where  $\mathbf{e}_k$  is the edge vertex touching  $\mathbf{c}$  and  $\mathbf{f}_k$  is the diagonal vertex on the face touching  $\mathbf{c}$ . Due to the use of special subdivision rules for Bézier edge condition in near-boundary layers, this linear relation is only valid to the points that are not in the first three layers in the base mesh (marked in red, orange and purple in Fig. 4(b)).

Using the subdivision curve for illustration (Fig. 4(c)), the limit stencil of the third control point  $\mathbf{P}_3$  can be derived by first subdividing the control polygon once and then applying  $\mathbf{l}(\mathbf{P}_3) = \mathbf{l}(\mathbf{p}_4)$ . Since the standard rule will be used to calculate





**Fig. 6.** Illustration of Catmull-Clark subdivision with Bézier edge conditions. The mesh after one step of subdivision is shown in dashed lines. The special subdivision stencils are used to calculate the new points in the first four rows along each boundary piece. The subdivision matrices involved in the tensor product calculation are those for univariate Bézier end conditions, and are shown in the upper right. An example of the derived stencil for  $c'_{3,4}$  is shown at the bottom. The old control points involved in this stencil are marked by black circles.

the new point for  $p_4$  in following subdivision steps, the standard limit stencil can be used to compute  $l(p_4)$ . Therefore, the linear relation can be formulated as  $l(p_3) = l(p_4) = (p_3 + 4p_4 + p_5)/6 = (1.5p_2 + 3.5p_3 + p_4)/6$ .

### 5.3. Continuity across boundary

In order to use the input model-space trimming curve  $B$  as part of the subdivision surface boundary (for a gap-free join) and maintain the first-derivative conditions across the boundary, the first two layers of the base mesh (the red and orange control points in Fig. 4(b)) are set up in the following way. We use  $c_{i,j}$  to denote the  $j$ th control point in the  $i$ th layer locally along a boundary piece (separated by corners).

The boundary points  $c_{1,j}$  of the base mesh (red points in Fig. 4(b)) are set to be the actual control points of the control polygons of the boundary curves of the trimmed NURBS surface. Note that these control polygons are not exactly the original input ones (as discussed in Section 4.3).

The second layer control points (orange points in Fig. 4(b))  $c_{2,j}$  are set to lie on the tangent plane along the boundary of the trimmed NURBS surface. For each boundary point  $d_{1,j} = (u_{1,j}, v_{1,j})$  in the domain partition  $M_d$ , its tangent plane on the NURBS surface is decided by its neighbouring control points in the first two rows in the base mesh  $M$  (see Fig. 4(b)). Therefore, for control points in the orange layer which are not corners, we have

$$\nabla S(u_{1,j}, v_{1,j}) \cdot \mathbf{w} = (c_{2,j-1} + 4c_{2,j} + c_{2,j+1} - c_{1,j-1} - 4c_{1,j} - c_{1,j+1})/2,$$

where the direction of vector  $\mathbf{w} = (w_u, w_v)$  is chosen to follow the knot line emanating from  $d_{1,j}$  in  $M_d$  (see Fig. 4(a)). For corners (shown as solid orange dots), we have

$$\mathbf{w}_1 \mathbf{H} \mathbf{w}_2^T = 9(c_{2,2} + c_{1,1} - c_{1,2} - c_{2,1}),$$

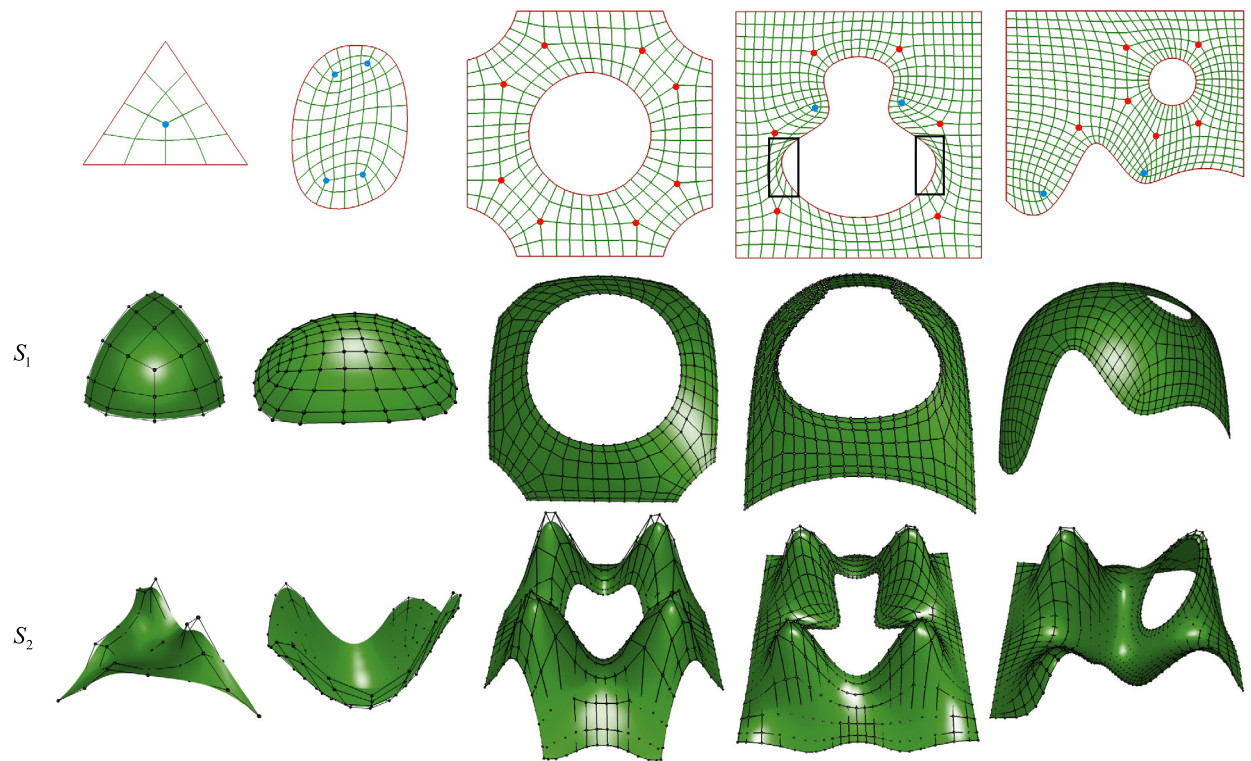
where  $\mathbf{H}$  is the Hessian matrix of input surface  $S$  at a new corner point  $(u_{1,1}, v_{1,1})$  in the domain partition  $M_d$ , and  $\mathbf{w}_1, \mathbf{w}_2$  follow the directions of the two boundary knot lines emanating from this corner (see Fig. 4(a)). The left part of this equation is a generalisation of  $\frac{\partial^2 S}{\partial u \partial v}$  for directions of  $(1, 0), (0, 1)$  to  $\mathbf{w}_1, \mathbf{w}_2$ .

The magnitude of vectors  $\mathbf{w}, \mathbf{w}_1, \mathbf{w}_2$  is related to the rescaling of the knot spacings of boundary curves before and after conversion. We observed that a good choice is the first non-zero knot interval along the new knot line, i.e., the magnitude of  $\mathbf{w}$  is  $|d_{2,j} - d_{1,j}|$ . Note that, these vectors are chosen to make the control point positions vary smoothly along the surface boundary. The result is exactly  $G^1$  at those shared control points. This is also the typical way that NURBS modelling systems (e.g., Rhinoceros) approximate  $G^1$  in a simple way. Future research will consider where to put the second-layer control points on the local tangent planes to achieve  $G^1$  continuity at all points along the shared boundary curve (Che et al., 2005).

In this way, along each boundary of the subdivision surface, the boundary curve and the first derivative across the boundary are set up using both the original NURBS surface and intersection curves  $B$  as input data. If the same data is used on both sides along the boundary, there will be a  $G^1$  join between the adjacent surfaces.

## 6. Results

The evaluation of the conversion is performed by computing the mesh difference between the input trimmed NURBS surface  $S$  and the limit surface  $C$  of the output base mesh  $M$  (for Catmull-Clark subdivision with Bézier edge conditions). Two measurements are used. One is the distance error scaled by the bounding box diagonal of the surface  $S$ . The other is the normal-vector error using the angle difference in degrees. The mesh of surface  $S$  is obtained by tessellating the



**Fig. 7.** The conversion results. The first row shows different trimming loops and the corresponding decomposition in the trimmed domain region with EV5 marked in red and EV3 in blue. The second and third rows show the converted subdivision surfaces with NURBS surface  $S_1$  and  $S_2$  as input surface, respectively. Note that they use the same decomposition in parameter space (the first row).  $S_2$  has high curvature. The distorted knot lines highlighted in the black boxes in the first row are due to the requirement of uniform knot spacings for boundary curve. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

**Table 1**

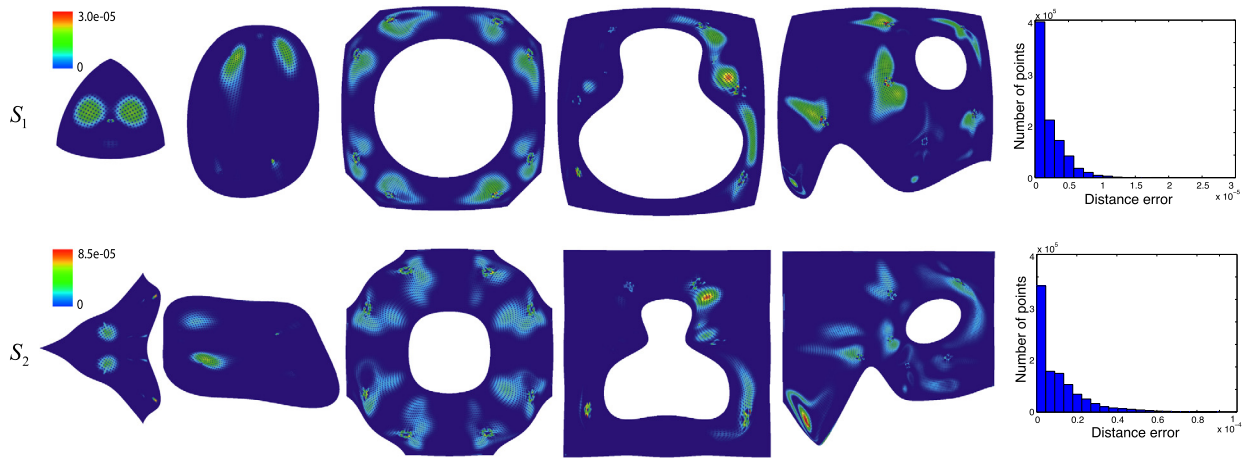
The refinement step used in the base mesh topology construction vs. maximum error (for examples in Fig. 8, 5th column).

Number of refinement steps	Number of control points	Max distance error				Max normal error	
		$S_1$	ratio	$S_2$	ratio	$S_1$	$S_2$
0	718	$2.01 \times 10^{-3}$	–	$5.63 \times 10^{-3}$	–	1.783	9.628
1	2681	$6.11 \times 10^{-4}$	3.3	$1.82 \times 10^{-3}$	3.1	1.037	7.445
2	10868	$1.05 \times 10^{-4}$	5.8	$2.31 \times 10^{-4}$	7.8	0.538	4.758
3	43096	$2.94 \times 10^{-5}$	3.6	$8.24 \times 10^{-5}$	2.8	0.273	2.01

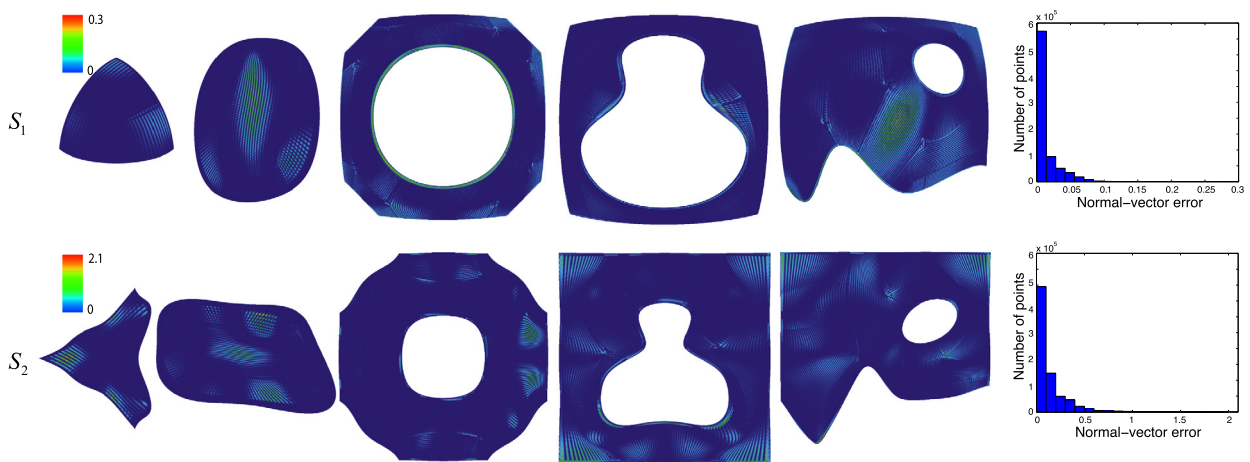
trimmed NURBS surface with a specified threshold (the largest edge length relative to the bounding box diagonal is less than  $1.0 \times 10^{-5}$  in our examples). The mesh of surface  $C$  is obtained by first subdividing the base mesh  $k$  times ( $k = 2$  in all examples), and then applying limit-point rules to calculate the limit positions of the points in the subdivided mesh. The normal vector for each limit point is calculated using the tangent stencils described in Andersson and Stewart (2010, Ch. 6.1).

We examine the behaviour of our algorithms on several kinds of trimming loops in domain space. The partitions in the trimmed domain region are shown in the first row of Fig. 7. We use two NURBS surfaces with the same parametric trimming curves as input surface  $S$ , referred to as  $S_1$  and  $S_2$ .  $S_2$  has high curvature. The conversion results are shown in the second and third row of Fig. 7, respectively. These are the results if coarse base meshes are required and their approximation errors are of the order of  $10^{-3}$ . The unnatural alignment marked in black boxes in the left corner of Fig. 7 is caused by the fact that we move the boundary vertices around to get uniform knot spacings for the boundary curve. It will be improved if converting to non-uniform subdivision schemes.

The approximation error is totally dependent on the number of control points used to represent the subdivision surface. As shown in Table 1, the maximum error drops when increasing the number of refinements in the topology construction step. The ratio by which it drops from one step to the next is around 4. Since we measure the maximum of the error over the entire surface, we can expect only quadratic convergence because of EVs (the surface is only  $C^1$  and has only linear precision at EVs). There is a tradeoff between accuracy and number of control points. For entertainment applications, the designer will mostly opt for an editable coarse control mesh for Catmull–Clark subdivision. Our method can produce



**Fig. 8.** The distance error plots of conversion results in Fig. 7. The error is scaled by the bounding box diagonal of the trimmed surface. The number of refinement steps is set to 3 to keep the approximation error below  $1.0 \times 10^{-4}$ . The histograms in the last column explicitly illustrate the error distribution of the examples in the fifth column.



**Fig. 9.** The normal-vector error plots of the conversion results (angle difference measured in degrees).

an editable coarse mesh with the EV placement condition satisfied (Fig. 7). For applications with stringent accuracy requirement, the approximation error should not be seriously worse than the error between the trimming curve  $B$  and its images on original NURBS surfaces. This mismatch can be used as the tolerance of the conversion.  $1.0 \times 10^{-4}$  is used as the conversion tolerance (the number of refinement steps is 3) in order to produce the evaluation results in Figs. 8 and 9.

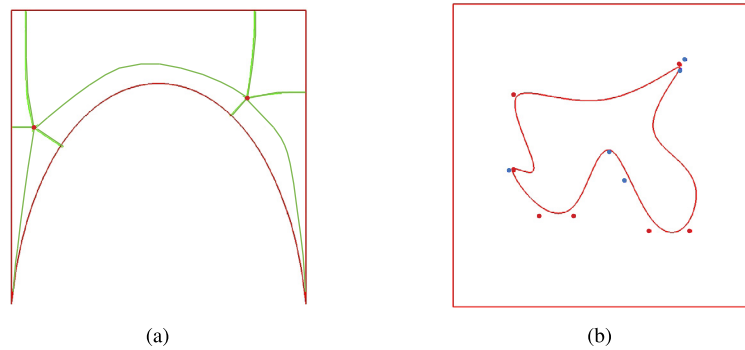
We plot the approximation error over the limit surface  $C$  measured in distance (Fig. 8) and in normal-vector angle difference (Fig. 9). The error distribution histograms in the last column of Figs. 8 and 9 explicitly illustrate the error distribution of the examples in the fifth column. It can be seen that over 90% of approximation errors are below one fifth of the maximum value.

The conversion can be done in seconds when coarse base mesh is required (the number of control points is of the order of  $10^3$ ) and in minutes when accuracy is required (the number of control points is of the order of  $10^4$ ).

## 7. Conclusion and future work

This paper presents a novel approach for the conversion of trimmed NURBS surfaces to Catmull–Clark subdivision surfaces with Bézier edge conditions. We first construct the topology of the base mesh for subdivision by performing cross-field based decomposition in the parameter space. The control point positions are then calculated by inverting a linear system built from limit stencils and cross-boundary continuity constraints. Our method can achieve the conversion within a specified tolerance and maintain the tangential continuity of adjacent trimmed surfaces.

Although in this paper we only exhibit the conversion of trimmed bicubic NURBS surfaces to Catmull–Clark subdivision surfaces, the framework can be directly applied to non-uniform subdivision schemes, except that different limit stencils need to be applied in the second step. The ultimate goal of this work is to provide tools for both entertainment applications



**Fig. 10.** Limitations of our method. (a) Sharp corners become EVs with valency 3. (b) In domain space, the trimming curve with high curvature results in many EVs close to the trimming curve.

(Catmull–Clark subdivision) and CAD applications (trimmed NURBS and high-degree non-uniform subdivision) which require exactness. In future, we will further investigate the conversion of trimmed NURBS surfaces to the subdivision scheme of [Cashman et al. \(2009a\)](#), which is non-uniform and high degree.

As demanded for most industry applications, the original continuity across the shared boundary should be maintained when the information of the adjacent surface is provided. Currently, we use Bézier edge conditions and guarantee the tangent plane continuity across the boundary. Our future work will investigate the method to achieve up to  $C^2$  with Catmull–Clark subdivision and even higher continuity across the shared boundary of two NURBS patches with Cashman's subdivision framework.

Moreover, the current decomposition is based on cross field, which is orthogonal. Corners become EV3 when the angle is small (as shown in [Fig. 10\(a\)](#)). A future challenge will be investigating the non-orthogonal cross field. With more degrees of freedom, better alignment could be achieved at sharp corners. Also as shown in [Fig. 10\(b\)](#), the parametric trimming curve with high curvature will result in many EVs around the trimming curve. The quad partition will be quite dense if we keep the uniformity of the knot spacings after satisfying the EVs placement condition. This problem could be improved by using non-orthogonal cross field to reduce the number of EVs around high curvature points on the trimming curves.

## References

- Andersson, L.E., Stewart, N.F., 2010. *Introduction to the Mathematics of Subdivision Surfaces*. Society for Industrial and Applied Mathematics.
- Bommes, D., Campen, M., Ebke, H.C., Alliez, P., Kobbelt, L., 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 98.
- Bommes, D., Lévy, B., Pietroni, N., Puppo, E., Silva, T.C., Tarini, M., Zorin, D., 2012. State of the art in quad meshing. In: *Eurographics STARS*.
- Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 77.
- Campen, M., Bommes, D., Kobbelt, L., 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Trans. Graph.* 31, 110.
- Cashman, T.J., Augsdörfer, U.H., Dodgson, N.A., Sabin, M.A., 2009a. NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Trans. Graph.* 28, 46.
- Cashman, T.J., Dodgson, N.A., Sabin, M.A., 2009b. A symmetric, non-uniform, refine and smooth subdivision algorithm for general degree B-splines. *Comput. Aided Geom. Des.* 26, 94–104.
- Cashman, T.J., Dodgson, N.A., Sabin, M.A., 2009c. Selective knot insertion for symmetric, non-uniform refine and smooth B-spline subdivision. *Comput. Aided Geom. Des.* 26, 472–479.
- Catmull, E., Clark, J., 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Des.* 10, 350–355.
- Che, X., Liang, X., Li, Q., 2005. Continuity conditions of adjacent NURBS surfaces. *Comput. Aided Geom. Des.* 22, 285–298.
- DeRose, T., Kass, M., Truong, T., 1998. Subdivision surfaces in character animation. In: *SIGGRAPH '98*. ACM Press, New York, USA, pp. 85–94.
- Dong, S., Bremer, P.T., Garland, M., Pascucci, V., Hart, J.C., 2006. Spectral surface quadrangulation. *ACM Trans. Graph.* 25, 1057–1066.
- Farin, G., 2001. *Curves and Surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc.
- Farin, G., Hoschek, J., Kim, M.S., 2002. Intersection problems. In: *Handbook of Computer Aided Geometric Design*, pp. 623–649.
- Halstead, M., Kass, M., DeRose, T., 1993. Efficient, fair interpolation using Catmull–Clark surfaces. In: *SIGGRAPH '93*. ACM Press, New York, USA, pp. 35–44.
- Hamann, B., Tsai, P.Y., 1996. A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves. *Comput. Aided Des.* 28, 461–472.
- Huang, J., Zhang, M., Ma, J., Liu, X., Kobbelt, L., Bao, H., 2008. Spectral quadrangulation with orientation and alignment control. *ACM Trans. Graph.* 27, 147.
- Hui, K., Wu, Y.B., 2005. Feature-based decomposition of trimmed surface. *Comput. Aided Des.* 37, 859–867.
- Kälberer, F., Nieser, M., Polthier, K., 2007. QuadCover – surface parameterization using branched coverings. *Comput. Graph. Forum* 26, 375–384.
- Kobbelt, L., 2010. *Polygon Mesh Processing*. A K Peters Limited.
- Lacewell, D., Burley, B., 2007. Exact evaluation of Catmull–Clark subdivision surfaces near B-spline boundaries. *J. Graph. GPU Game Tools* 12, 7–15.
- Li, X., Chen, F., 2009. Exact and approximate representations of trimmed surfaces with NURBS and Bézier surfaces. In: *CAD/Graphics*, pp. 286–291.
- Liu, Y., Xu, W., Wang, J., Zhu, L., Guo, B., Chen, F., Wang, G., 2011. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.* 30, 140.
- Loop, C., Schaefer, S., 2008. Approximating Catmull–Clark subdivision surfaces with bicubic patches. *ACM Trans. Graph.* 27, 8.
- Loop, C., Schaefer, S., Ni, T., Castaño, I., 2009. Approximating subdivision surfaces with Gregory patches for hardware tessellation. *ACM Trans. Graph.* 28, 151.
- Müller, K., Fünzig, C., Reusche, L., Hansford, D., Farin, G., Hagen, H., 2010. Dinus: double insertion, nonuniform, stationary subdivision surfaces. *ACM Trans. Graph.* 29, 25.
- Müller, K., Reusche, L., Fellner, D., 2006. Extended subdivision surfaces: building a bridge between NURBS and Catmull–Clark surfaces. *ACM Trans. Graph.* 25, 268–292.

- Palacios, J., Zhang, E., 2007. Rotational symmetry field design on surfaces. *ACM Trans. Graph.* 26, 55.
- Palacios, J., Zhang, E., 2010. Interactive visualization of rotational symmetry fields on surfaces. *IEEE Trans. Vis. Comput. Graph.* 17, 947–955.
- Ray, N., Li, W.C., Lévy, B., Sheffer, A., Alliez, P., 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 1460–1485.
- Ray, N., Vallet, B., Li, W.C., Lévy, B., 2008. N-symmetry direction field design. *ACM Trans. Graph.* 27, 10.
- Reusche, L., 2005. Conversion of trimmed NURBS surfaces into subdivision surfaces. Ph.D. thesis. Technical University Braunschweig, Germany.
- Sabin, M.A., 2010. *Analysis and Design of Univariate Subdivision Schemes*. Springer.
- Schmidt, R., Wüchner, R., Bletzinger, K.U., 2012. Isogeometric analysis of trimmed NURBS geometries. *Comput. Methods Appl. Mech. Eng.* 241–244, 93–111.
- Sederberg, T.W., Finnigan, G.T., Li, X., Lin, H., Ipson, H., 2008. Watertight trimmed NURBS. *ACM Trans. Graph.* 27, 79.
- Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A., 2003. T-splines and T-NURCCs. *ACM Trans. Graph.* 22, 477–484.
- Sederberg, T.W., Zheng, J., Sewell, D., Sabin, M., 1998. Non-uniform recursive subdivision surfaces. In: *SIGGRAPH '98*. ACM Press, New York, USA, pp. 387–394.
- Song, Q., Wang, J., 2007. Generating parametric blending surfaces based on partial reparameterization of base surfaces. *Comput. Aided Des.* 39, 953–963.
- Stam, J., 1998. Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values. In: *SIGGRAPH '98*. ACM Press, New York, USA, pp. 395–404.
- Zhang, E., Mischaikow, K., Turk, G., 2006. Vector field design on surfaces. *ACM Trans. Graph.* 25, 1294–1326.
- Zhang, M., Huang, J., Liu, X., Bao, H., 2010. A wave-based anisotropic quadrangulation method. *ACM Trans. Graph.* 29, 118.
- Zhang, M., Huang, J., Liu, X., Bao, H., 2013. A divide-and-conquer approach to quad remeshing. *IEEE Trans. Vis. Comput. Graph.* 19, 941–952.
- Zorin, D., Kristjansson, D., 2002. Evaluation of piecewise smooth subdivision surfaces. *Vis. Comput.* 18, 299–315.