C¹ triangular Coons surface construction and image interpolation based on new Side-Side and Side-Vertex interpolation operators

Mingshan Qiu, Yuanpeng Zhu

School of Mathematics, South China University of Technology, Guangzhou, China

* ypzhu@scut.edu.cn

Abstract

In this paper, a new class of rational quadratic/linear trigonometric Hermite functions with two shape parameters is proposed. Based on these Hermite functions, new improved first class of Side-Side (FCSS), second class of Side-Side (SCSS), first class of Side-Vertex (FCSV) and second class of Side-Vertex (SCSV) interpolation operators are proposed respectively, which can be used to construct C¹ Coons surfaces over triangular domain. By altering the values of two shape parameters, the shape of the Coons surface patch can be adjusted flexibly, but without affecting the function values and partial derivatives of the boundaries. For constructing the triangular surface patches with the center of mass passing through a fixed point, we also give a center of mass function value control method, by which we can solve the corresponding shape parameter values. Moreover, we also apply these four improved interpolation operators to image interpolation. Compared with some widely used image interpolation methods, our methods achieve competitive performance.

1. Introduction

The problem of constructing surfaces through a given set of scattered data points has a wide range of applications in many fields, such as medical imaging, scientific visualization, architectural design and aerospace materials manufacturing. In order to satisfy various requirements and obtain pleasing surfaces, many methods have been proposed, such as NURBS, Bézier-like surfaces, and so on. These methods can accurately construct quadrangular surfaces and free form surfaces. However, they still have disadvantages in some applications. One of them is that when adjusting the shape inside the surface, the boundary function values or partial derivatives will be affected simultaneously, which cannot satisfy the requirement of maintaining the shape of the boundaries.

In order to solve this problem, Coons [1] proposed a new surface construction method. If we need to construct a quadrangular surface that satisfies the boundary interpolation requirement, we only need to provide the data points with four boundaries. In real applications, bilinear Coons patches and bicubic Coons patches are widely used. However, the shape of the classic Coons patch is fixed and related to the given data points. Wang et al. [2] made an
improvement and used second-order trigonometric blending functions to construct biquadratic trigonometric polynomial Coons surface patches with adjustable parameters. Zou et al. [3] constructed a rational Coons surface over rectangular domain, which inherits the good properties of bicubic Coons patches. In her method, R-Hermit basis functions are applied to the interpolation operators, and the parameters in basis functions are used to adjust the shape of the patch. The rational Coons surface can accurately represent ellipsoidal surface, elliptical cone surface, and elliptic cylinder surface.

In computer-aided design (CAD), surfaces are often defined over rectangular domain, for CAD is originally applied to the design of objects with rectangular structures such as cars and aircraft fuselages. However, as the application range of surface geometric modeling technology becomes wider, and the shape of the design surface becomes more complex, rectangular surfaces and rectangular topology appear some limitations. Many scholars have begun to study the surface patches of non-rectangular topologies, such as the triangular surface patches. One of the methods used to construct Coons surface patches over triangular domain is transfinite interpolation, which constructs surface patches by interpolating the boundary values. The two main methods of transfinite interpolation are Side-Side method and Side-Vertex method. Barnhill et al. [4] first presented the Side-Side method, which uses Boolean sum to derive rational blending interpolation functions for constructing triangular surface patches that interpolate the boundary values and partial derivatives. Their method requires the cross partial derivatives to be compatible. If not, Gregory’s square [5] can be used to solve this problem. Barnhill et al. [6] used the polynomial blending functions to replace the rational blending functions, and proposed a polynomial Boolean sum interpolant. In [7], Side-Vertex method was proposed by Nielson, which is based upon the convex combinations of three interpolation operators consisting of univariate interpolation along lines segments joining a vertex and its opposing side. In [8], Nielson proposed a new method for constructing G2 surfaces. This new method utilizes the surface normals rather than the partial derivatives on the boundaries. In [9], Hagen generalized the Side-Vertex method using the geometric Hermite operators. The constructed surface patch guarantees to interpolate the function values, derivatives and curvature values of the triangle boundaries. In [10], this result has been further generalized to open and closed G2 triangular surface patches by Hagen. Zhang et al. showed a new method to construct triangular surface patches in [11], which is the combination of an interior interpolation operator and three Nielson’s Side-Vertex interpolation operators. This method shows better approximation precision than Nielson’s Side-Vertex method.

The Coons surface patches constructed by the above methods have advantages in continuity and interpolation properties. However, once the boundary values are given, designers cannot further adjust the shape of the surface because it is already fixed. In order to construct adjustable curves and surfaces, researchers have proposed some convenient and effective methods, one of them is to apply shape parameters to the basis functions. In [12], Zhang gave four cubic C-Hermite polynomials depending on parameter $\alpha > 0$, and they can exactly reproduce circles and cylinders. In [13], Han presented quadratic trigonometric polynomial curves with a shape parameter, and those curves satisfy $C^1$ continuity. In [14], Li et al. introduced two kinds of Coons surface patches generated by the hyperbolic polynomial functions and trigonometric polynomial functions respectively. Both two improved schemes achieve adjustable shape through combining the shape parameters in basis functions. In [15], Wu et al. used two classes of trigonometric Hermite polynomial functions to construct the Coons surface patches over triangular domain. By changing the shape parameters, the shape of the Coons surface patch can be adjusted. In [16], Ibraheem et al. developed a scheme for constructing positive surfaces via parameter control.

Interpolation is also an important tool in image magnification and visual processing. Image interpolation is an image processing technique that uses low resolution images (LR) to obtain...
high resolution images (HR). Nearest-neighbor interpolation, bilinear interpolation and bicubic interpolation are classic image interpolation algorithms. Bicubic interpolation was proposed by Keys in [17], and it shows an order of accuracy between that of linear interpolation and that of cubic spline. An algorithm for image enlargement and reduction with arbitrary accuracy using the scaling function of B-spline wavelet was offered by Yang et al. in [18]. An algorithm of image smooth interpolation using cubic B-spline was proposed by Hou et al. in [19], which can perfectly implement image enlargement, reduction and noise smoothing. In recent years, many new image interpolation algorithms have been proposed. In [20], Takeda et al. applied kernel regression to image interpolation and reconstruction. This algorithm is ideal for processing noisy images and sampling irregular images. In [21], Li et al. showed a novel edge-orientation adaptive interpolation scheme. Their idea is to use the geometric duality of LR and HR covariance coefficients for image interpolation. In [22], Zhang et al. applied the fractal analysis method to the field of image magnification, and proposed a novel single-image super-resolution procedure based upon the rational fractal interpolation. In [23], Hussain et al. used a class of quadratic triangle B-spline with control parameters to implement image interpolation, and creatively applied genetic algorithms to optimize the selection of parameters, so as to obtain better quality interpolation images.

The purpose of this paper is to propose the improved first class of Side-Side (FCSS), second class of Side-Side (SCSS), first class of Side-Vertex (FCSV) and second class of Side-Vertex (SCSV) interpolation operators using a new class of rational quadratic/linear trigonometric Hermite functions. We use these methods to construct Coons surface patches and implement $C^1$ surface stitching. Based on the property that the surface shape can be adjusted through parameters, we introduce a center of mass function value control method, by which we can construct surface patches with the center of mass passing through a fixed point. Finally, we apply these proposed interpolation methods to image interpolation for upscaling the images with factor 2, 3 and 4, and make comparisons between our methods and some existing image interpolation methods.

The rest of this paper is organized as follows. The definition and properties of a new class of rational quadratic/linear trigonometric Hermite functions are described in Section 2. In Section 3, we discuss the four improved interpolation operators: FCSS, SCSS, FCSV and SCSV. The constructed Coons surface patches and stitched surfaces are showed in Section 4. Section 5 shows a center of mass function value control method. In Section 6, four interpolation methods are applied to image interpolation. Section 7 summarizes our work.

2. Trigonometric Hermite functions with shape parameters

For $u \in [0, 1]$, we construct a new class of rational quadratic/linear trigonometric Hermite functions with two shape parameters $\alpha, \beta \in (0, +\infty)$ as follow

$$
\begin{align*}
T_0(u) &= \frac{\alpha(1 - \sin \frac{\pi u}{2})}{\alpha(1 - \sin \frac{\pi u}{2}) + \beta(1 - \cos \frac{\pi u}{2})}, \\
T_1(u) &= \frac{\beta(1 - \cos \frac{\pi u}{2})}{\alpha(1 - \sin \frac{\pi u}{2}) + \beta(1 - \cos \frac{\pi u}{2})}, \\
T_2(u) &= 2\pi \sin \frac{\pi u}{2} (1 - \sin \frac{\pi u}{2}) \\
&\quad \pi \alpha(1 - \sin \frac{\pi u}{2}) + \pi \beta(1 - \cos \frac{\pi u}{2}), \\
T_3(u) &= \frac{2\beta \cos \frac{\pi u}{2} (1 - \cos \frac{\pi u}{2})}{\pi \alpha(1 - \sin \frac{\pi u}{2}) + \pi \beta(1 - \cos \frac{\pi u}{2})}.
\end{align*}
$$

(1)
It is easy to check that $T_0(u)+T_i(u) \equiv 1$. And the trigonometric Hermite functions $T_i(u)(i = 0, 1, 2, 3)$ have the following important end-point properties

\[
\begin{align*}
T_0(0) &= 1, T_i(1) = 0, T_0(0) = 0, T_0(1) = 0, \\
T_i(0) &= 0, T_i(1) = 1, T_i(0) = 0, T_i(1) = 0, \\
T_2(0) &= 0, T_2(1) = 0, T_2(0) = 1, T_2(1) = 0, \\
T_3(0) &= 0, T_3(1) = 0, T_3(0) = 0, T_3(1) = 1.
\end{align*}
\]

3. The improved Side-Side methods and Side-Vertex methods

Over triangular domain, we use transfinite interpolant to construct Coons surface patches. In order to facilitate the discussion over triangular domain, we first need to convert the Cartesian coordinates to the barycentric coordinates.

3.1 Barycentric coordinates

Let the non-degenerate triangle be $T$, and its vertices be $V_i(x_i, y_i)(i = 1, 2, 3)$ counterclockwise (see Fig 1). $\partial T$ represents the boundary of $T$. The vectors of three boundaries are marked as $e_1 = V_3 - V_2, e_2 = V_1 - V_3, e_3 = V_2 - V_1$. For any point $P$ inside $T$, the barycentric coordinates is $(b_1, b_2, b_3)$, where $b_i = \frac{A_i}{A}$, $A$ is the area of $T$, and $A_i$ is the area of $\Delta PV_j V_k$. $F$ represents the continuous function on $T$.

The conversion relationship between the Cartesian coordinates $(x, y)$ and the barycentric coordinates $(b_1, b_2, b_3)$ is as follows

\[
\begin{align*}
x &= b_1x_1 + b_2x_2 + b_3x_3, \\
y &= b_1y_1 + b_2y_2 + b_3y_3, \\
1 &= b_1 + b_2 + b_3.
\end{align*}
\]

3.2 The First Class of Side-Side (FCSS) interpolation operations

The classic Side-Side method, proposed by Barnhill, Birkhoff and Gordon [4], is also called the parallel projection method. Each interpolation operator $P_i[F]$ represents the linear
interpolation along segments parallel to the \( i \)th side of \( T \). The geometric principle of the Side-Side method is shown in Fig 2.

Now we make an improvement to the first class of Side-Side interpolation operators. The Hermite functions in the operators are replaced by the new class of rational quadratic/linear trigonometric Hermite functions given in Eq (1), and the first class of Side-Side (FCSS for short) interpolation operators \( P_i[F](i = 1, 2, 3) \) are defined as follows

\[
P_1[F] = T_0 \left( \frac{b_3}{1 - b_1} \right) F(M_1) + T_1 \left( \frac{b_3}{1 - b_1} \right) F(N_1) + T_2 \left( \frac{b_3}{1 - b_1} \right) (1 - b_1) \frac{\partial F}{\partial e_1}(M_1) + T_3 \left( \frac{b_3}{1 - b_1} \right) (1 - b_1) \frac{\partial F}{\partial e_1}(N_1),
\]

where

\[
M_1 = b_1V_1 + (1 - b_1)V_2, \quad N_1 = b_1V_1 + (1 - b_1)V_3,
\]

Fig 2. The geometric principle of Side-Side method. (a) The geometric principle of interpolation operator \( P_1[F] \). (b) The geometric principle of interpolation operator \( P_2[F] \). (c) The geometric principle of interpolation operator \( P_3[F] \). (d) The Coons surface patch obtained by Side-Side method.
Hermite functions with two trigonometric Hermite functions

3.3 The second class of Side-Side (SCSS) interpolation operations

and \( \frac{\partial F}{\partial e_1}(x) \) is the partial derivative of \( F \) along the direction of \( e_1 \) at point \( x \).

\[
P_i[F] = T_0 \left( \frac{b_1}{1 - b_2} \right) F(M_1) + T_1 \left( \frac{b_1}{1 - b_2} \right) F(N_1)
+ T_2 \left( \frac{b_1}{1 - b_2} \right) \left( 1 - b_2 \right) \frac{\partial F}{\partial e_1} (M_1) + T_3 \left( \frac{b_1}{1 - b_2} \right) \left( 1 - b_2 \right) \frac{\partial F}{\partial e_1} (N_1),
\]

where\[ M_1 = b_1 V_1 + (1 - b_2) V_2, \quad N_1 = b_2 V_1 + (1 - b_2) V_2, \]

and \( \frac{\partial F}{\partial e_2}(x) \) is the partial derivative of \( F \) along the direction of \( e_2 \) at point \( x \).

\[
P_i[F] = T_0 \left( \frac{b_2}{1 - b_1} \right) F(M_2) + T_1 \left( \frac{b_2}{1 - b_1} \right) F(N_2)
+ T_2 \left( \frac{b_2}{1 - b_1} \right) \left( 1 - b_1 \right) \frac{\partial F}{\partial e_2} (M_2) + T_3 \left( \frac{b_2}{1 - b_1} \right) \left( 1 - b_1 \right) \frac{\partial F}{\partial e_2} (N_2),
\]

where\[ M_2 = b_1 V_1 + (1 - b_2) V_3, \quad N_2 = b_2 V_1 + (1 - b_2) V_3, \]

and \( \frac{\partial F}{\partial e_3}(x) \) is the partial derivative of \( F \) along the direction of \( e_3 \) at point \( x \).

3.3 The second class of Side-Side (SCSS) interpolation operations

The second class of Side-Side interpolation operators also can be obtained by replacing the Hermite functions with two trigonometric Hermite functions \( T_i(u) \) and \( T_j(u) \) given in Eq (1), respectively. And the second class of Side-Side (SCSS for short) interpolation operators \( P_i[F] \) \((i = 1, 2, 3)\) are defined as follows

\[
P_i[F] = T_0 \left( \frac{b_1}{1 - b_1} \right) F(N_1) + T_1 \left( \frac{b_1}{1 - b_1} \right) F(M_1)
+ T_2 \left( \frac{b_1}{1 - b_1} \right) \left( 1 - b_1 \right) \frac{\partial F}{\partial e_1} (N_1) - T_3 \left( \frac{b_1}{1 - b_1} \right) \left( 1 - b_1 \right) \frac{\partial F}{\partial e_1} (M_1),
\]

\[
P_i[F] = T_0 \left( \frac{b_2}{1 - b_2} \right) F(N_2) + T_1 \left( \frac{b_2}{1 - b_2} \right) F(M_2)
+ T_2 \left( \frac{b_2}{1 - b_2} \right) \left( 1 - b_2 \right) \frac{\partial F}{\partial e_2} (N_2) - T_3 \left( \frac{b_2}{1 - b_2} \right) \left( 1 - b_2 \right) \frac{\partial F}{\partial e_2} (M_2),
\]

\[
P_i[F] = T_0 \left( \frac{b_3}{1 - b_3} \right) F(N_3) + T_1 \left( \frac{b_3}{1 - b_3} \right) F(M_3)
+ T_2 \left( \frac{b_3}{1 - b_3} \right) \left( 1 - b_3 \right) \frac{\partial F}{\partial e_3} (N_3) - T_3 \left( \frac{b_3}{1 - b_3} \right) \left( 1 - b_3 \right) \frac{\partial F}{\partial e_3} (M_3),
\]

where the definitions of \( M_i, N_i, \frac{\partial F}{\partial e_i} (i = 1, 2, 3) \) are the same as (2), (3), and (4).
3.4 The First Class of Side-Vertex (FCSV) interpolation operations

The Side-Vertex method proposed by Nielson [7] is also called the radial projection method. Each operator represents the interpolation along line joining a vertex and its opposing side. The principle of Side-Vertex method is shown in Fig 3.

Set $S_i$ as the point on the boundary corresponding to vertex $V_i$, and the coordinates of $S_i$ is

$$S_i = S_i(x, y) = \left( \frac{x - b_i x_i}{1 - b_i}, \frac{y - b_i y_i}{1 - b_i} \right) = \left( \frac{b_j x_j + b_k x_k}{b_j + b_k}, \frac{b_j y_j + b_k y_k}{b_j + b_k} \right).$$

We also make an improvement to the first class of Side-Vertex interpolation operators. The Hermite functions are replaced by the trigonometric Hermite functions given in Eq (1), and the first class of Side-Vertex (FCSV for short) interpolation operators $Q_i[F](i = 1, 2, 3)$ are showed as follows

$$Q_1[F] = T_0(b_1) F(S_1) + T_1(b_1) R_1(1) + T_1(b_1) F(V_1) + T_1(b_1) R_1(0),$$

$$Q_2[F] = T_0(b_2) F(S_2) + T_2(b_2) R_2(1) + T_2(b_2) F(V_2) + T_2(b_2) R_2(0),$$

where $R_1(1) = \frac{b_j \frac{dF}{dx}(V_1) + b_k \frac{dF}{dx}(V_1)}{b_j + b_k}$, $R_1(0) = \frac{b_j \frac{dF}{dx}(V_1) + b_k \frac{dF}{dx}(V_1)}{b_j + b_k}$.
Given the FCSS and SCSS interpolation operators $C$

By using the new trigonometric Hermite functions

3.5 The second class of Side-Vertex (SCSV) interpolation operations

where $Q_i[F] = T_0(b_1)F(S_1) + T_2(b_2)R_i''(1) + T_i(b_1)F(V_i) + T_3(b_3)R_i'(0)$,

where $R_i'(1) = \frac{b_1 \frac{\partial F}{\partial s}(S_1) + b_2 \frac{\partial F}{\partial v}(S_1)}{b_1 + b_2}$, $R_i'(0) = \frac{b_1 \frac{\partial F}{\partial s}(V_i) + b_2 \frac{\partial F}{\partial v}(V_i)}{b_1 + b_2}$.

3.5 The second class of Side-Vertex (SCSV) interpolation operations

By using the new trigonometric Hermite functions $T_i(u)$ and $T_j(u)$ given in Eq (1), the second class of Side-Vertex (SCSV for short) interpolation operators $Q_i[F](i = 1, 2, 3)$ are defined as follows

$Q_1[F] = T_1(1 - b_1)F(S_1) + T_3(1 - b_3)R_1'(1) + T_1(b_1)F(V_1) - T_2(b_2)R_1'(0)$,

$Q_2[F] = T_1(1 - b_2)F(S_2) + T_3(1 - b_3)R_2'(1) + T_1(b_2)F(V_2) - T_2(b_2)R_2'(0)$,

$Q_3[F] = T_1(1 - b_1)F(S_3) + T_3(1 - b_3)R_3'(1) + T_1(b_1)F(V_3) - T_2(b_2)R_3'(0)$.

where the definitions of $R_i'(1), R_i'(0) (i = 1, 2, 3)$ are the same as Eqs (9)–(11).

4.C1 Coons surfaces over triangular domain

4.1 Construction of Coons surfaces using Side-Side methods

Given the FCSS and SCSS interpolation operators $P_i[F](i = 1, 2, 3)$, the Coons surface patch is defined by the Boolean sum of three operators

$P[F] = \omega_1 P_1[F] + \omega_2 P_2[F] + \omega_3 P_3[F]$, \hspace{1cm} (15)

where

$\omega_i = \frac{b_i^2}{\sum_{m=1}^{3} b_m^2}$.

\hspace{1cm} (16)

**Theorem 1.** The weight function $\omega_i (i = 1, 2, 3)$ given in Eq (16) has the following properties:

1. $\omega_1 + \omega_2 + \omega_3 = 1$.
2. $\omega_i |_{e_i} = 0$.
3. For $i = 1, 2, 3, j = 1, 2, 3, \frac{\partial \omega_i}{\partial \theta_j} |_{e_i} = 0$.

**Proof of Theorem 1.** Through simple calculations, we have

1. $\omega_1 + \omega_2 + \omega_3 = \frac{b_1^2 + b_2^2 + b_3^2}{\sum_{m=1}^{3} b_m^2} = 1$.

2. Obviously, $b_i = 0$ on the side $e_i$, so

$\omega_i |_{e_i} = \frac{b_i}{\sum_{m=1}^{3} b_m^2} |_{e_i} = 0$. 
3. Taking $i = 1$ as an example. Since in Barycentric coordinates, \( e' = V_3 - V_2 = (0, -1, 1) \), \( e' = V_1 - V_3 = (1, 0, -1) \), \( e' = V_2 - V_1 = (1, 1, 0) \) we have

\[
\frac{\partial \omega_i}{\partial e'_1} = \nabla \omega_i \cdot e'_1
\]

\[
= \left( \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \right) \cdot (0, -1, 1)
\]

\[
= \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \times 0 + \left[ -\frac{2b_1^2b_2}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times (-1)
\]

\[
+ \left[ -\frac{2b_1^2b_3}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times 1
\]

\[
= \frac{2b_1^2(b_2 - b_3)}{(b_1^2 + b_2^2 + b_3^2)^2}.
\]

\[
\frac{\partial \omega_i}{\partial e'_2} = \nabla \omega_i \cdot e'_2
\]

\[
= \left( \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \right) \cdot (1, 0, -1)
\]

\[
= \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \times 1 + \left[ -\frac{2b_1^2b_2}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times 0
\]

\[
+ \left[ -\frac{2b_1^2b_3}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times (-1)
\]

\[
= \frac{2b_1b_2^2 + 2b_1b_3^2 + 2b_1b_2^2}{(b_1^2 + b_2^2 + b_3^2)^2}.
\]

\[
\frac{\partial \omega_i}{\partial e'_3} = \nabla \omega_i \cdot e'_3
\]

\[
= \left( \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \right) \cdot (-1, 1, 0)
\]

\[
= \frac{2b_1(b_1^2 + b_2^2 + b_3^2) - 2b_1^2b_1}{(b_1^2 + b_2^2 + b_3^2)^2} \times (-1) + \left[ -\frac{2b_1^2b_2}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times 1
\]

\[
+ \left[ -\frac{2b_1^2b_3}{(b_1^2 + b_2^2 + b_3^2)^2} \right] \times 0
\]

\[
= -2b_1b_2 - 2b_1b_3 - 2b_1b_2.
\]
From this together with $b_1 = 0$ on the side $e_1$, it follows immediately that $\frac{\partial F}{\partial e_1} |_{e_1} = 0$. The proofs for $\frac{\partial F}{\partial e_j} |_{e_j} = 0$ and $\frac{\partial F}{\partial e_k} |_{e_k} = 0$ are similar.

**Theorem 2.** Let $F(x, y) \in C^1(T)$, when $(x, y) \in \partial T$, $P[F]$ will interpolate $F(x, y)$ and its first-order partial derivatives, so $P[F] \in C^1(T)$.

**Proof of Theorem 2.** The following proof is based on the FCSS interpolation operators over triangular domain $T$. For $b_i = 0$ and $b_j = 1 - b_k$ on the side $e_i$, $i = 1, 2, 3$; $j = 2, 3, 1$; $k = 3, 1, 2$, the Boolean sum of three interpolation operators is

$$P[F]|_{e_i} = \frac{b^2_j}{b^2_j + b^2_k} P_j[F] + \frac{b^2_k}{b^2_j + b^2_k} P_k[F],$$

and

$$P_j[F]|_{e_i} = T_0 \left( \frac{b_j}{1 - b_j} \right) F(M_j) + T_1 \left( \frac{b_j}{1 - b_j} \right) F(N_j) + T_2 \left( \frac{b_j}{1 - b_j} \right) (1 - b_j) \frac{\partial F}{\partial e_j} (M_j) + T_3 \left( \frac{b_j}{1 - b_j} \right) (1 - b_j) \frac{\partial F}{\partial e_j} (N_j) = F(M_j),$$

$$P_k[F]|_{e_i} = T_0 \left( \frac{b_j}{1 - b_j} \right) F(M_k) + T_1 \left( \frac{b_j}{1 - b_j} \right) F(N_k) + T_2 \left( \frac{b_j}{1 - b_j} \right) (1 - b_k) \frac{\partial F}{\partial e_k} (M_k) + T_3 \left( \frac{b_j}{1 - b_j} \right) (1 - b_k) \frac{\partial F}{\partial e_k} (N_k) = F(N_k).$$

According to the geometric principle of Side-Side method shown in Fig 2, We find that $M_j$ and $N_k$ are points on the same boundary, thus

$$P[F]|_{e_i} = \frac{b^2_j}{b^2_j + b^2_k} F(M_j) + \frac{b^2_k}{b^2_j + b^2_k} F(N_k) = F(S).$$

Therefore, we have proved the following properties

$$P[F]|_{\partial T} = F(x, y)|_{\partial T}.$$
Next, we prove that $P[F]$ interpolates the first derivative values of the triangle boundaries. Taking $i = 1$ as an example, according to Theorem 1 (3), we have

$$\frac{\partial P[F]}{\partial \varepsilon_i} \bigg|_{e_i} = \frac{\partial (\omega_1 P_1[F])}{\partial \varepsilon_i} \bigg|_{e_i} + \frac{\partial (\omega_2 P_2[F])}{\partial \varepsilon_i} \bigg|_{e_i} + \frac{\partial (\omega_3 P_3[F])}{\partial \varepsilon_i} \bigg|_{e_i}$$

$$= \left( \frac{\partial \omega_1}{\partial \varepsilon_i} P_1[F] \right) \bigg|_{e_i} + \left( \frac{\partial \omega_2}{\partial \varepsilon_i} P_2[F] \right) \bigg|_{e_i} + \left( \frac{\partial \omega_3}{\partial \varepsilon_i} P_3[F] \right) \bigg|_{e_i}$$

$$= -2b_2b_3 \left( \frac{\partial F(M_2)}{(b_2 + b_3)^2} \right) + \left( \frac{b_2^2}{(b_2 + b_3)^2} \right) \frac{\partial F(N_s)}{\partial \varepsilon_i}$$

$$= \frac{\partial F(S_i)}{\partial \varepsilon_i}.$$
thus, we prove that

\[
\frac{\partial P[F]}{\partial e_j} \bigg|_{c_i} = \frac{\partial F(S_i)}{\partial e_j}.
\]

Similarly, both on the side \(e_2\) and \(e_3\) of \(T\), we can verify that

\[
\frac{\partial P[F]}{\partial e_j} \bigg|_{c_2} = \frac{\partial F(S_2)}{\partial e_j}, \quad \frac{\partial P[F]}{\partial e_j} \bigg|_{c_3} = \frac{\partial F(S_3)}{\partial e_j}.
\]

The proof of the SCSS interpolation operators is analogy to the FCSS interpolation operators. Therefore, we have verified that \(P[F]\) interpolates \(F(x, y)\) and its first-order partial derivatives of the triangle boundaries, which means that \(P[F] \in C^1(T)\).

Both FCSS and SCSS interpolation operators contain two shape parameters \(\alpha\) and \(\beta\). Through adjusting the values of two shape parameters, we can obtain Coons surfaces of different shapes. Comparing to the classic Side-Side interpolation operators based on cubic Hermite basis functions, which construct the surface patches with fixed shape, both FCSS and SCSS interpolation operators show advantages in shape adjustment. Besides, Theorem 2 illustrates that no matter how the Coons surface is adjusted, function values and first-order partial derivatives of the triangle boundaries are always interpolated.

Given the interpolating function \(f(x, y) = 10 - x^2 - y^2\), the interpolating region \(T\) is a triangular domain with vertex \(V_1(0, 0), V_2(1, 0), V_3(0, 1)\). For FCSS, when the shape parameters \(\alpha > \beta\), Fig 4d shows convex shape in the middle of the Coons surface patch. For SCSS, the effect of parameters on the shape is more obvious. When \(\alpha < \beta\), the surface patch in Fig 4e shows convex shape. When \(\alpha > \beta\), the surface patch Fig 4f presents concave shape. Fig 4 demonstrates that the interpolation surfaces can be adjusted conveniently by shape parameters. However, no matter how the shape changes, the Coons surface patches constructed by FCSS and SCSS still keep the interpolating property of Theorem 2.

Complex surface design can usually be achieved by splicing multiple small patches with simple shapes. Based on this technology, we first triangulate the area, use FCSS and SCSS interpolation operators to perform segmented transfinite interpolation, and finally stitch all the Coons patches to obtain a new surface.

Given the triangulation on Fig 5a and the surface function \(g(x, y) = 5.2 \exp \left(\frac{-x^2 - (y - 0.5)^2}{4}\right)\), \((x, y) \in \mathbb{R}^2\), Fig 5b and 5c display the splice surfaces constructed by the FCSS and SCSS interpolations operators respectively with shape parameters \(\alpha = 1, \beta = 1\). According to Theorem 2, the splice surfaces are \(C^1\) continuous. From Fig 5, we can see that the obtained stitching surfaces fit well.

### 4.2 Construction of Coons surfaces using Side-Vertex methods

Given the FCSV and SCSV interpolation operators \(Q_i[F]\) \((i = 1, 2, 3; j = 2, 3, 1; k = 3, 1, 2)\), the construction of Coons surface patches is based upon the Boolean sum of three operators

\[
Q[F] = \bar{\omega}_i Q_i[F] + \bar{\omega}_j Q_j[F] + \bar{\omega}_k Q_k[F],
\]

where

\[
\bar{\omega}_i = \frac{b_i^2 b_k^2}{b_i^2 b_k^2 + b_i^2 b_j^2 + b_j^2 b_k^2}.
\]

**Theorem 3.** The weight function \(\bar{\omega}_i\) \((i = 1, 2, 3)\) given in Eq (18) has the following properties:
Fig 4. Comparison between the classic Side-Side interpolation operators and the improved parametric Side-Side methods to construct Coons surface patches. (a) The given function $f(x, y)$. (b) The classic Side-Side interpolation operators. (c) FCSS with $\alpha = 1$, $\beta = 10$. (d) FCSS with $\alpha = 10$, $\beta = 1$. (e) SCSS with $\alpha = 1$, $\beta = 1.03$. (f) SCSS with $\alpha = 1.03$, $\beta = 1$.

https://doi.org/10.1371/journal.pone.0231617.g004
1. \( \bar{\omega}_1 + \bar{\omega}_2 + \bar{\omega}_3 = 1 \).
2. \( \bar{\omega}_i|_{i,j} = \delta_{i,j} \).
3. For \( i = 1, 2, 3, j = 1, 2, 3, \frac{\partial \bar{\omega}_i}{\partial y_j}|_{i,j} = 0 \).

Proof of Theorem 3. The proof of Theorem 3 is analogy to Theorem 1.

Theorem 4. Let \( F(x, y) \in C^1(T) \), when \( (x, y) \in \partial T \), \( Q[F] \) will interpolate \( F(x, y) \) and its first-order partial derivatives, so \( Q[F] \in C^1(T) \).

Proof of Theorem 4. The proof of Theorem 4 is analogy to Theorem 2.

The basis functions of two improved Side-Vertex methods also have shape parameters \( \alpha \) and \( \beta \) analogous to the improved Side-Side methods, so we can adjust the shape by simply...
altering the parameter values, while the Coons surface patches still keep the interpolation property the same. Comparing to Nielson’s Side-Vertex method which is only suitable to construct fixed shape surfaces, the improved Side-Vertex methods have advantages.

Given the function \( f(x, y) = 10 - x^2 - y^2 \), and the interpolating region \( T \) with \( V_1(0, 0), V_2(1, 0) V_3(0, 1) \) the same as Fig 4, we can see that both FCSV and SCSV interpolations operators have good performances in changing the shape of the Coons surface patches from Fig 6. For FCSV, when the shape parameters \( \alpha < \beta \), the Coons surface patch presents concave shape (see Fig 6c). For SCSV, when \( \alpha < \beta \), the Coons surface patch presents convex shape (see Fig 6e). However, when \( \alpha > \beta \), it presents concave shape (see Fig 6f).

Given the triangulation (see Fig 7a) and the interpolation function
\[
g(x, y) = 5.2\exp \left[ \frac{-x^2-(y-0.5)^2}{4} \right], \ (x, y) \in R^2,\ \text{Fig 7 shows the splice surfaces constructed by FCSV and SCSV with shape parameters } \alpha = 1, \beta = 1.\] When constructing surfaces with complex shape, we can apply this surface stitching method to achieve accurate reproduction.

### 4.3 Approximation precision of the improved Side-Side methods and Side-Vertex methods

In order to demonstrate that the improved four interpolation methods have good approximation precision, we perform the following experiments. Given a test surface \( F(x, y) \), we first sample the boundary data, then apply the four improved interpolation methods to reconstruct interpolation surfaces and calculate the approximation accuracy of these restructured surfaces. We will also compare the approximation accuracy of the improved methods with the classic Side-Side method and Nielson’s Side-Vertex method. The four test surfaces in this experiment are constructed by the following functions

\[
h_1(x, y) = 0.75 \exp \left[ -\frac{(9x - 2)^2 + (9y - 2)^2}{4} \right] + 0.75 \exp \left[ -\frac{(9x + 1)^2}{49} - \frac{9y + 1}{10} \right] + 0.5 \exp \left[ -\frac{(9x - 7)^2 + (9y - 3)^2}{4} \right] - 0.2 \exp[-(9x - 4)^2 - (9y - 7)^2],
\]
\[
h_2(x, y) = \frac{\tanh(5 - 9x - 9y) + 1}{9},
\]
\[
h_3(x, y) = \frac{1.25 + \cos(5.4y)}{6 + 6(3x - 1)^2},
\]
\[
h_4(x, y) = \frac{5.2 \exp(18y - 18x)}{9 \exp(18y - 18x) + 9}.
\]

where \((x, y) \in T\) with \( V_1(0, 0), V_2(1, 0), V_3(0, 1) \). The ranges of these functions are \( h_1 \in [0, 1.220], h_2 \in [0, 0.223], h_3 \in [0, 0.375], h_4 \in [0, 0.578] \). Error between the reconstructed surface and the original surface is measured by root mean square (RMS)

\[
RMS = \sqrt{\frac{\sum_{j=0}^{M} \sum_{i=0}^{N} (p(x_i, y_j) - h(x_i, y_j))^2}{(M + 1)(N + 1)}},
\]

where \( M = N = 20, p(x_i, y_j) \) is the function value of interpolation surface, \( h(x_i, y_j) \) is the function value of original surface. To normalize the results, we divide RMS by the difference between the maximum and minimum values of \( h \) over \( T \). In the following experiments, all \( \alpha = 1, \beta = 1. \)
Fig 6. Comparison between Nielson’s Side-Vertex method and the improved parametric Side-Vertex methods to construct Coons surface patches. (a) The given function $f(x, y)$. (b) Nielson’s Side-Vertex method. (c) FCSV with $\alpha = 1, \beta = 10$. (d) FCSV with $\alpha = 10, \beta = 1$. (e) SCSV with $\alpha = 1, \beta = 2$. (f) SCSV with $\alpha = 2, \beta = 1$.

https://doi.org/10.1371/journal.pone.0231617.g006
Table 1 is the RMS errors of six interpolation methods: the classic Side-Side method, Nielson’s Side-Vertex method, FCSS, SCSS, FCSV and SCSV. Fig 8 is the original surface and reconstructed surfaces of $h_3$.

Comparing the results of different interpolation methods in Table 1, regardless any shape of the test surface, the RMS errors of the improved Side-Side methods and Side-Vertex methods are less than the classic Side-Side method and Nielson’s Side-Vertex method, and less than 6 percent of the range of the function values. These results demonstrate that the four improved interpolation methods have higher approximation accuracy, and are more competitive than the classic Side-Side method and Nielson’s Side-Vertex method.
Table 1. RMS errors of six interpolation methods.

|            | $h_1$ | $h_2$ | $h_3$ | $h_4$ |
|------------|-------|-------|-------|-------|
| The classic Side-Side | 0.078 | 0.049 | 0.021 | 0.066 |
| Nielson’s Side-Vertex | 0.126 | 0.080 | 0.071 | 0.066 |
| FCSS       | 0.054 | 0.047 | 0.011 | 0.052 |
| SCSS       | 0.054 | 0.047 | 0.011 | 0.052 |
| FCSV       | 0.048 | 0.023 | 0.015 | 0.056 |
| SCSV       | 0.048 | 0.023 | 0.011 | 0.056 |

https://doi.org/10.1371/journal.pone.0231617.t001

5. Center of mass function value control method

In practical applications, the center of mass over triangular domain is more important than other points for shape control. Since we can use shape parameters $\alpha$, $\beta$ to adjust the surface, we expect the control of the center of mass can also be achieved through parameters. The method introduced below can solve this issue perfectly.

If we require that the function value of the interpolation surface at the center of mass is equal to $M$, where $M \in \{ \min_{1 \leq i \leq 3} F(V_i), \max_{1 \leq i \leq 3} F(V_i) \}$, we only need the shape parameters to satisfy:

$$P[F] = \omega_1 P_1[F] + \omega_2 P_2[F] + \omega_3 P[F] = M,$$

or

$$Q[F] = \omega_1 Q_1[F] + \omega_2 Q_2[F] + \omega_3 Q_3[F] = M.$$  \hspace{1cm} (19)

For convenience, we use a new variable $k$ to represent the shape parameters $\alpha, \beta$, where $k = \frac{\alpha}{\beta}$. The center of mass coordinates of the triangle is known to be $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. By substituting the shape parameter ratio $k$ and the coordinates into (19) and making a simplification, we can derive the following equations about $k$.

For FCSS, the equation of $k$ is

$$\left\{ \pi [F(M_1) + F(M_2) + F(M_3)] + \frac{2\sqrt{2}}{3} \left[ \frac{\partial F}{\partial e_1}(M_1) + \frac{\partial F}{\partial e_2}(M_2) + \frac{\partial F}{\partial e_3}(M_3) \right] - 3\pi M \right\} k = - \pi [F(N_1) + F(N_2) + F(N_3)] + \frac{2\sqrt{2}}{3} \left[ \frac{\partial F}{\partial e_1}(N_1) + \frac{\partial F}{\partial e_2}(N_2) + \frac{\partial F}{\partial e_3}(N_3) \right] + 3\pi M.$$  \hspace{1cm} (20)

For SCSS, the equation of $k$ is

$$3\pi Mk = \pi [F(N_1) + F(N_2) + F(N_3) + F(M_1) + F(M_2) + F(M_3)] + \frac{2\sqrt{2}}{3} \left[ - \frac{\partial F}{\partial e_1}(N_1) - \frac{\partial F}{\partial e_2}(N_2) - \frac{\partial F}{\partial e_3}(N_3) + \frac{\partial F}{\partial e_1}(M_1) + \frac{\partial F}{\partial e_2}(M_2) + \frac{\partial F}{\partial e_3}(M_3) \right] - 3\pi M.$$  \hspace{1cm} (21)

For FCSV, the equation of $k$ is

$$\left\{ \pi [F(S_1) + F(S_2) + F(S_3)] + R_1(1) + R_2(1) + R_3(1) - 3\pi M \right\} k = (\sqrt{3} - 2)\pi [F(V_1) + F(V_2) + F(V_3)] + \sqrt{3}(2 - \sqrt{3})[R_1(0) + R_2(0) + R_3(0)] + 3\pi(2 - \sqrt{3})M.$$  \hspace{1cm} (22)
Fig 8. The original surface and reconstructed surfaces of $h_3$. (a) Original surface. (b) The classic Side-Side method. (c) Nielson’s Side-Vertex method. (d) FCSS. (e) SCSS. (f) FCSV. (g) SCSV.

https://doi.org/10.1371/journal.pone.0231617.g008
Table 2. Center of Mass function values and the parameter ratios $k$.

|       | FCSS | SCSS_1 | SCSS_2 | FCSV | SCSV_1 | SCSV_2 |
|-------|------|--------|--------|------|--------|--------|
| $M$   | 5.11 | 4.87   | 5.14   | 4.80 | 4.39   | 5.22   |
| $k$   | 1.01 | 1.10   | 0.99   | 1.05 | 1.32   | 0.78   |

For SCSS, the equation of $k$ is

$$
\pi k[F(S_1) + F(S_2) + F(S_3)] - k[R_1(1) + R_2(1) + R_3(1)] \\
+ (7 - 4\sqrt{3})\pi k[F(V_1) + F(V_2) + F(V_3)] + (7\sqrt{3} - 12)k[R_1(0) + R_2(0) + R_3(0)] \\
- \frac{3[(2 - \sqrt{3})k^2 + (8 - 4\sqrt{3})] \pi M}{k} \\
= -\pi(2 - \sqrt{3})[F(S_1) + F(S_2) + F(S_3)] + (2 - \sqrt{3})[R_1(1) + R_2(1) + R_3(1)] \\
- (2 - \sqrt{3})\pi[R(V_1) + F(V_2) + F(V_3)] - (2\sqrt{3} - 3)[R_1(0) + R_2(0) + R_3(0)] + 3(2 - \sqrt{3})\pi M. (23)
$$

By solving the above simple first or quadratic equation, the shape parameter ratio $k$ of FCSS, SCSS, FCSV or SCSV can be obtained.

Set three vertices of the triangle as $V_1(0, 0)$, $V_2(1, 0)$, $V_3(0, 1)$, and the value of center of mass as $M$. The boundaries of the surface patch are constructed by function $g(x, y) = 5.2exp\left[\frac{-x^2-(y-0.5)^2}{4}\right], (x, y) \in R^2$. We use these four methods to construct Coons surface patches respectively. The given values $M$ and the calculated shape parameter ratios are shown in Table 2. Fig 9 shows the improved Side-Side methods and Side-Vertex methods Coons surface patches with the corresponding $M$ displayed in Table 2.

### 6. Apply four new interpolation operators to image interpolation

In this section, we apply four new constructed FCSS, SCSS, FCSV and SCSV interpolation operators to image interpolation and achieve image magnification. The proposed methods are tested with six different standard images, which are gray scale and named as 'Airplane', 'Baboon', 'Flower', 'Peppers', 'Rail' (kodim19) and 'Truck' respectively. They are obtained from SIPI image database, University of Southern California [24] and the classic Kodak test image data set (http://r0k.us/graphics/kodak/). Besides, these methods will be compared with some classic schemes: bilinear interpolation, bicubic interpolation, kernel regression (KR) [20], new edge-directed interpolation (NEDI) [21], cubic trigonometric B-spline image interpolation scheme (CTB) [25] and rational cubic B-spline image interpolation scheme (RBC) [26]. The following algorithm illustrates the steps to implement image interpolation using the methods of this paper.

**Algorithm 1.**

**Input.** Standard test image, parameters $\alpha$ and $\beta$.

**Output.** Upscaling image, PSNR value and SSIM value.

**Step 1. Convert the input image to a gray matrix $G$.** The elements $g_{ij}(i = 1, 2, \ldots m, j = 1, 2, \ldots, n)$ of matrix $G$ represent the pixel values of corresponding pixel points in the $i$ row and $j$ column of the image.

**Step 2. Down-sample the original image to obtain low-resolution image.** Depending on whether the required interpolation factor is 2, 3 or 3, the pixel matrix of low resolution image is obtained by down-sampling the original image with interpolation factor 1/2 (sample one pixel every two pixels), 1/3 (sample one pixel every three pixels) or 1/4 (sample one pixel every four pixels). The grayscale matrix corresponding to the low resolution image is recorded as $G_{d}$. 
Fig 9. Coons surface patches with fixed function values on the center of mass. (a) FCSS. (b) SCSS_1. (c) SCSS_2. (d) FCSV. (e) FCSV_1. (f) FCSV_2.

https://doi.org/10.1371/journal.pone.0231617.g009
Step 3. Use FCSS, SCSS, FCSV and SCSV for fragment interpolation. Subdivide the pixel matrix $G_d$ into small pixel matrices. Each small pixel matrix is $2 \times 2$, and stores the pixel values of a rectangular region in the down-sampled image. We divide the rectangular region into two triangle regions by triangulation and perform transfinite interpolation on each triangle patch. In order to implement the interpolation scheme, an approximation is made that the boundary values are considered as piecewise linear. After transfinite interpolation is done on all small matrices, we get the high resolution image. The gray matrix calculated by the interpolation method is recorded as $G = \bar{g}_{ij}(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$.

Step 4. Results analysis. Compare the results of our four interpolation methods against some existing methods by analyzing the visual quality and calculating the values of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM). PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right).$$

where MSE is the mean square error defined as $\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (g_{ij} - \bar{g}_{ij})^2$.

SSIM is defined as

$$\text{SSIM} = \frac{(2\mu_1 \mu_2 + c_1)(2\sigma_{12} + c_2)}{((\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2))}.$$

where $\mu_1$ is the mean of $\bar{g}_{ij}$, $\sigma_1^2$ is the variance of $\bar{g}_{ij}$, $\sigma_{12}$ is the covariance of $g_{ij}$ and $\bar{g}_{ij}$, $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are the constant used to maintain stability. $L$ is the dynamic range of the pixel values. Generally $k_1 = 0.01, k_2 = 0.03$. Higher PSNR and SSIM mean better image quality.

When the image interpolation is performed by FCSS, SCSS, FCSV and SCSV, different interpolation results can be obtained by altering the values of parameters $\alpha$ and $\beta$, and interpolated images will have different qualities. However, based on Theorems 2 and 4, for the boundary values do not change with the shape parameters, the adjustment of the shape by parameters is limited to a reasonable range, which means that the adjusted image always maintains a high degree of fit with the original image, and a fine adjustment will result in a high quality image. We perform experiments on image enlargement with interpolation factors 2, 3 and 4, respectively. In the experiment of image magnification with interpolation factor 2, we compare our four methods with bilinear interpolation, bicubic interpolation, KR, NEDI, and CTB. Table 3 lists the values of parameters $\alpha, \beta$ used in all experiments. Tables 4 and 5 list the PSNR and SSIM values for each image interpolation scheme with interpolation factor 2. PSNR and SSIM data for KR and NEDI are referenced from [27].

In terms of the data shown in Tables 4 and 5, the PSNR values of our four methods are the highest among all image interpolation schemes. In general, the PSNR values of our interpolation methods have pleasing results, that is, the error between corresponding pixels is small. The SSIM values of our methods are better than bilinear interpolation, bicubic interpolation and CTB. Although our methods are not as good as KR and NEDI in terms of SSIM, we can
see the average SSIM values of our methods differ from KR by less than 0.05, and differ from NEDI by less than 0.08. Therefore, our methods still give pleasing results overall.

Figs 10 and 11 are upscaling images with interpolation factor 2 of seven methods: bilinear interpolation, bicubic interpolation, CTB, FCSS, SCSS, FCSV and SCSV. Comparing the visual quality of different interpolation schemes, for Fig 10b, bilinear interpolation shows the problem of damaging the texture of the image. In addition, bilinear interpolation is poor at edge preservation, where jaggy artifacts are observed alone the shadow edges, see Fig 11b. The image obtained by bicubic interpolation is obviously better than bilinear interpolation, and bicubic interpolation works well in maintaining edges. But comparing the texture direction in Fig 11a and 11c, we find that bicubic interpolation fails in preserving the texture direction. In terms of CTB, it has made a breakthrough in keeping the edges of the image. But looking at Fig 10d, we find that the reconstruction of the texture regions is still not perfect. The four methods we proposed overcome the problem of jagged edges to a certain extent. It can also be seen from Fig 10e–10h that these four improved interpolation methods preserve the texture direction and reproduce the texture details.

Furthermore, we also use FCSS, SCSS, FCSV and SCSV to upscale images with interpolation factor 3 and 4, and compare with NEDI and RBC. The standard test images are ‘Peppers’, ‘Airplane’ and ‘Flower’. In order to fully evaluate the quality of high resolution images, in addition to PSNR and SSIM, we have also added two image quality assessment metrics, which are MS-SSIM [28] and FSIM [29]. Similarly, the higher values of MS-SSIM and FSIM indicate the better quality of upscaling images. The PSNR, SSIM, MS-SSIM, FSIM results of NEDI and RBC are referenced from [26].

Tables 6–9 show PSNR, SSIM, MS-SSIM, and FSIM values of different image interpolation methods. For Tables 6–8, it can be noted that the PSNR, SSIM, and MS-SSIM of our four image interpolation methods are the highest. In Table 9, the FSIM of our method is higher than NEDI and slightly lower than RBC.

Figs 12 and 13 show the upscaling images of ‘Peppers’, and ‘Flower’, which are produced by our image interpolation methods with interpolation factor 4. Comparing the visual quality of two standard test images, when the interpolation factor increases, our methods are prone to jagged edges, see Fig 12. Therefore, in terms of processing the images with long edges, our
Fig 10. Comparison of Baboon images (Interpolation Factor 2). (a) Original image. (b) Bilinear interpolation. (c) Bicubic interpolation. (d) CTB. (e) FCSS. (f) SCSS. (g) PCSV. (h) SCSV.

https://doi.org/10.1371/journal.pone.0231617.g010
Fig 11. Comparison of Truck images (Interpolation Factor 2). (a) Original image. (b) Bilinear interpolation. (c) Bicubic interpolation. (d) CTB. (e) FCSS. (f) SCSS. (g) FCSV. (h) SCSV.

https://doi.org/10.1371/journal.pone.0231617.g011
methods are not as good as RBC. But for the images containing texture regions, our methods have advantages, see Fig 13. This result is also reflected in Table 9. For pepper and airplane images, the FSIM values of our methods are lower than RBC. But the difference between the FSIM value of each of our method and RBC is significantly smaller, and even the results of our methods are better.

7. Conclusions

In order to overcome the shape defect of the classic Side-Side method and Nielson’s Side-Vertex method, this paper proposes a new class of rational quadratic/linear trigonometric Hermite functions with two shape parameters. Applying them we obtain the new improved first class of Side-Side (FCSS), second class of Side-Side (SCSS), first class of Side-Vertex (FCSV) and

Table 6. PSNR values of different interpolation schemes (Interpolation Factor 3 and 4).

| Images | NEDI | RBC | FCSS | SCSS | FCSV | SCSV | NEDI | RBC | FCSS | SCSS | FCSV | SCSV |
|--------|------|-----|------|------|------|------|------|-----|------|------|------|------|
| Peppers | 24.58 | 25.16 | 28.51 | 28.51 | 28.47 | 28.50 | 22.69 | 22.98 | 26.68 | 26.68 | 26.64 | 26.65 |
| Airplane | 26.58 | 27.43 | 27.74 | 27.74 | 27.63 | 27.68 | 24.88 | 25.79 | 25.33 | 25.33 | 25.21 | 25.28 |
| Flower | 30.76 | 31.48 | 32.56 | 32.56 | 32.42 | 32.47 | 28.59 | 29.64 | 29.64 | 29.64 | 29.44 | 29.54 |
| Average | 27.31 | 28.02 | 29.60 | 29.60 | 29.51 | 29.55 | 25.39 | 26.14 | 27.22 | 27.22 | 27.10 | 27.16 |

https://doi.org/10.1371/journal.pone.0231617.t006

Table 7. SSIM values of different interpolation schemes (Interpolation Factor 3 and 4).

| Images | NEDI | RBC | FCSS | SCSS | FCSV | SCSV | NEDI | RBC | FCSS | SCSS | FCSV | SCSV |
|--------|------|-----|------|------|------|------|------|-----|------|------|------|------|
| Peppers | 0.726 | 0.770 | 0.820 | 0.820 | 0.818 | 0.818 | 0.669 | 0.699 | 0.783 | 0.783 | 0.780 | 0.780 |
| Airplane | 0.845 | 0.862 | 0.876 | 0.876 | 0.874 | 0.874 | 0.806 | 0.823 | 0.818 | 0.818 | 0.812 | 0.815 |
| Flower | 0.838 | 0.854 | 0.924 | 0.924 | 0.923 | 0.923 | 0.789 | 0.812 | 0.881 | 0.881 | 0.876 | 0.879 |
| Average | 0.803 | 0.829 | 0.873 | 0.873 | 0.872 | 0.872 | 0.755 | 0.778 | 0.827 | 0.827 | 0.823 | 0.825 |

https://doi.org/10.1371/journal.pone.0231617.t007

Table 8. MS-SSIM values of different interpolation schemes (Interpolation Factor 3 and 4).

| Images | NEDI | RBC | FCSS | SCSS | FCSV | SCSV | NEDI | RBC | FCSS | SCSS | FCSV | SCSV |
|--------|------|-----|------|------|------|------|------|-----|------|------|------|------|
| Peppers | 0.929 | 0.931 | 0.965 | 0.965 | 0.964 | 0.964 | 0.869 | 0.881 | 0.948 | 0.947 | 0.946 | 0.946 |
| Airplane | 0.946 | 0.959 | 0.971 | 0.971 | 0.970 | 0.971 | 0.718 | 0.940 | 0.940 | 0.940 | 0.938 | 0.939 |
| Flower | 0.947 | 0.954 | 0.983 | 0.983 | 0.982 | 0.982 | 0.915 | 0.926 | 0.960 | 0.960 | 0.959 | 0.959 |
| Average | 0.941 | 0.948 | 0.973 | 0.973 | 0.972 | 0.972 | 0.834 | 0.916 | 0.949 | 0.949 | 0.948 | 0.948 |

https://doi.org/10.1371/journal.pone.0231617.t008

Table 9. FSIM values of different interpolation schemes (Interpolation Factor 3 and 4).

| Images | NEDI | RBC | FCSS | SCSS | FCSV | SCSV | NEDI | RBC | FCSS | SCSS | FCSV | SCSV |
|--------|------|-----|------|------|------|------|------|-----|------|------|------|------|
| Peppers | 0.965 | 0.983 | 0.957 | 0.957 | 0.956 | 0.956 | 0.958 | 0.970 | 0.933 | 0.933 | 0.932 | 0.933 |
| Airplane | 0.941 | 0.971 | 0.952 | 0.952 | 0.951 | 0.951 | 0.943 | 0.952 | 0.910 | 0.910 | 0.907 | 0.909 |
| Flower | 0.936 | 0.961 | 0.975 | 0.975 | 0.974 | 0.975 | 0.936 | 0.954 | 0.952 | 0.952 | 0.950 | 0.951 |
| Average | 0.947 | 0.972 | 0.961 | 0.961 | 0.960 | 0.961 | 0.946 | 0.959 | 0.932 | 0.932 | 0.930 | 0.931 |

https://doi.org/10.1371/journal.pone.0231617.t009
Fig 12. Comparison of Peppers images (Interpolation Factor 4). (a) Original image. (b) FCSS. (c) SCSS. (d) FSCV (e) SCV.

https://doi.org/10.1371/journal.pone.0231617.g012
Fig 13. Comparison of Flower images (Interpolation Factor 4). (a) Original image. (b) FCSS. (c) SCSS. (d) FCSV. (e) SCSV.

https://doi.org/10.1371/journal.pone.0231617.g013
second class of Side-Vertex (SCSV) interpolation operators. These improved interpolation methods can be used to construct Coons surface patches and stitch surfaces that meet the following properties:

1. The shape of the Coons surface patch can be flexibly adjusted by altering the values of shape parameters.
2. Always maintain interpolating boundary values and partial derivatives.
3. $C^1$ continuity.

For the properties that we can adjust the shape of the surface patches using the shape parameters, a center of mass function value control method is proposed. When constructing a triangular patch that requires the center of mass passing through a fixed point, the corresponding parameter values can be calculated by this method. Moreover, FCSS, SCSS, FCSV and SCSV are applied to image interpolation. By adjusting the parameter values, it is possible to obtain images with better quality. The reason is that the Coons surface patches constructed by these proposed methods always satisfy the interpolation of boundary values and partial derivatives. Therefore, these methods ensure that the adjustment of the surface shape is limited to a reasonable range without destroying some of the properties required by the original image. Future work will concentrate on constructing $C^2$ Coons surfaces over triangular domain with shape parameters.

**Author Contributions**

**Conceptualization:** Mingshan Qiu, Yuanpeng Zhu.

**Data curation:** Mingshan Qiu, Yuanpeng Zhu.

**Formal analysis:** Mingshan Qiu, Yuanpeng Zhu.

**Funding acquisition:** Mingshan Qiu, Yuanpeng Zhu.

**Investigation:** Mingshan Qiu, Yuanpeng Zhu.

**Methodology:** Mingshan Qiu, Yuanpeng Zhu.

**Project administration:** Mingshan Qiu, Yuanpeng Zhu.

**Resources:** Mingshan Qiu, Yuanpeng Zhu.

**Software:** Mingshan Qiu, Yuanpeng Zhu.

**Supervision:** Mingshan Qiu, Yuanpeng Zhu.

**Validation:** Mingshan Qiu, Yuanpeng Zhu.

**Writing – original draft:** Mingshan Qiu, Yuanpeng Zhu.

**Writing – review & editing:** Mingshan Qiu, Yuanpeng Zhu.

**References**

1. Coons SA. Surfaces for computer-aided design of space forms. Technical Report MAC-TR-41, Project MAC, MIT. 1967.
2. Wang JX, Dong Y, Ni J. Biquadrate trigonometric polynomial coons patch with adjustable parameters. J. Dalian Jiaotong Univ. 2013; 34:110–112.
3. Zou J, Han XL. R-Coons surface with shape parameters. Journal of Graphics. 2019; 40:388–394.
