*Applied Mathematics*, **2011, 2, 369-378**
doi:10.4236/am.2011.23044 Published Online March 2011 (http://www.SciRP.org/journal/am)

# Optimal Interpolatory Wavelets Transform for Multiresolution Triangular Meshes

**Chong Zhao, Hanqiu Sun**
*Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China*
*E-mail*: *czhao@cse.cuhk.edu.hk*

*Received November* 7, 2010; *revised January* 27, 2011; *accepted January* 30, 2011

## Abstract

In recent years, several matrix-valued subdivisions have been proposed for triangular meshes. The matrix-valued subdivisions can simulate and extend the traditional scalar-valued subdivision, such as loop and $\sqrt{3}$ subdivision. In this paper, we study how to construct the matrix-valued subdivision wavelets, and propose the novel biorthogonal wavelet based on matrix-valued subdivisions on multiresolution triangular meshes. The new wavelets transform not only inherits the advantages of subdivision, but also offers more resolutions of complex models. Based on the matrix-valued wavelets proposed, we further optimize the wavelets transform for interactive modeling and visualization applications, and develop the efficient interpolatory loop subdivision wavelets transform. The transform simplifies the computation, and reduces the memory usage of matrix-valued wavelets transform. Our experiments showed that the novel wavelets transform is sufficiently stable, and performs well for noise reduction and fitting quality especially for multiresolution semi-regular triangular meshes.

**Keywords:** Lifting Scheme, Subdivision Wavelets, Matrix-Valued Subdivision

## 1. Introduction

The development of graphics applications and virtual reality demand complex models often with millions of vertices, which need large resources to process and also transmission through the networks. Since the subdivision based wavelets can efficiently represent highly detailed geometric models in resolutions, they are widely used in geometry compression and multiresolution editing. In addition, subdivision wavelets can be further customized to possess some desired properties, such as stability, orthogonalization and vanishing moment, using the lifting scheme. Since the lifting steps can be converted into local in-place operations in wavelet transform, if they are locally conducted, the reconstruction and decomposition of the resulting wavelets are not necessary to allocate auxiliary memory or solve a global system of linear equations. The fast wavelet transform based on the lifting scheme is usually simple, and can be performed in linear time.

Subdivision wavelets can be constructed based on subdivision schemes. Since the wavelets based on approximate subdivision have good shape preserve ability, efficient subdivision wavelets are constructed based on the approximate subdivision. In some applications, such as reversed engineering of scattered data and study of point clouds, where control points are data points, surface interpolation is an important requirement. In this paper, we propose an efficient wavelets construction for the matrix-valued interpolatory loop subdivision, for triangular meshes based on the lifting scheme. The resulting biorthogonal wavelet inherits the attractive advantage of having the most resolution levels of interpolatory loop refinement. As the analysis and synthesis transforms of the resulting wavelet are composed of only local lifting operations, they can be performed very efficiently in linear time using fully in-place calculations. In the rest of the paper, we first briefly review other work most related to our approach in Section 2. We introduce the background of matrix-valued subdivision, in particular the matrix-valued interpolatory loop subdivision in Section 3. We describe the lifted wavelets based on the matrix-valued subdivision in Section 4, and the algorithm to optimize the matrix-valued wavelets transform for multiresolution modeling/rendering applications in Section 5. The experimental results for the interpolatory wavelet analysis and their performance evaluation are given in Section 6. Finally, the summary of our work is given in Section 7.

## 2. Related Work

Wavelets based on subdivision surfaces have been proposed for geometry mesh processing for years. Lounsbery *et al.* [1] presented a new type of wavelets based on subdivision, generally on the surface of arbitrary topology. By generalizing the uniform subdivision in topology to a new irregular subdivision scheme, Valette and Prost [2,3] extended the work of Lounsbery and proposed a wavelet-based multiresolution analysis, to be applied directly to irregular meshes whose connectivity is unchanged in the wavelet analysis. Samavati *et al.* [4] showed how to use least-squares data fitting to reverse subdivision rules and constructed the wavelets by straightforward matrix observations. Samavati *et al.* [5] constructed multiresolution surfaces of arbitrary topologies by locally reversing the Doo-Sabin subdivision scheme. Since the lifting scheme proposed by Swelden [6] can generate new biorthogonal wavelets from the classic wavelets and lazy wavelets, it is an important tool to construct subdivision wavelets. Based on lifting scheme, Schroder and Sweldens showed how to construct lifting wavelets on the sphere with customized properties [7]. Using local lifting operations performed on polygonal meshes, Bertram *et al.* [8,9] gave a new construction of lifted biorthogonal wavelets on surfaces of arbitrary two manifold topology, and introduced the generalized B-spline subdivision-surface wavelets.

Using local lifting and the discrete inner production, Bertram [10] constructed a biorthogonal wavelet on the Loop subdivision. Li *et al.* [11] proposed unlifted Loop subdivision wavelet by optimizing free parameters in the extended subdivisions. Wang *et al.* [12] developed an effective wavelet construction based on general Catmull-Clark subdivisions and the resulted wavelets have better fitting quality than the previous Catmull-Clark like subdivision wavelets. They also constructed several new biorthogonal wavelets based on $\sqrt{3}$ subdivision over triangular meshes, and approximate and interpolatory $\sqrt{2}$ subdivision over quadrilateral meshes [13-15]. Zhang *et al.* presented a biorthogonal wavelet approach based on dual Doo-Sabin subdivision with the aid of the barycenter of the V-faces corresponding to old vertices [16]. The initial work on computing matrixvalued subdivision wavelets is presented by Zhao *et al.* in [17].

Chui and Jiang proposed a new approach to construct subdivision schemes, called matrix-valued subdivision [18,19]. Different with the scalar-valued subdivision, the dilation coefficients is not a number, but a matrix. He constructed the matrix-valued loop subdivision and $\sqrt{3}$ subdivision on the triangular mesh and Catmull-Clark subdivision on quadrilateral mesh. The data processed by the matrix-valued subdivision is a row vector including geometry information and other parameters for shape control. For matrix-valued subdivisions, versions of interpolatory subdivisions were introduced particularly for the purpose of Hermite interpolation. These considerations, however, are too restrictive to be useful for the construction of interpolatory matrix-valued templates in general. The most general extension of interpolatory loop and $\sqrt{3}$ surface subdivisions, from scalar to matrix considerations and without any restriction, for constructing symmetric interpolatory matrix-valued templates is formulated in [20].

## 3. Matrix-Valued Subdivision

I computer graphics, surface subdivision schemes are designed to generate visually continuous and smooth surfaces from some initial triangulations in the 3D domains iteratorly. For each iterative step, the subdivision has two simple operations: generating a new set of vertices, and connecting the vertices for new triangulation of higher resolution. The former is decided by the topological rule, which determines how the new vertices connect to the existing vertices. And the later is decided by the local averaging rule, which is designed to generate the new vertices by taking some weighted averages of the positions of the neighboring vertices. Local averaging rules can be designed by considering the refinement equation:

$$\phi(\mathrm{x}) = \sum_{\mathrm{k}} p_k \phi(2x-k), \quad x \in R^2$$

Here, $\phi(x)$ is called a refinable function, and the finite sequence $p_k$ called its corresponding refinement sequence or subdivision mask. For a control net with control points $v_k^m$, the subdivision mask $p_k$ provides the local averaging rule:

$$v_j^{m+1} = \sum_k v_k^m \mathrm{p}_{j\text{-}2k}, \quad m = 0,1,\cdots$$

where, for each $m = 1,2,\cdots$, the set $v_k^m$ denotes the set of vertices obtained after taking *m* iterations. The smoothness of the limiting 3-D subdivision surfaces is derived from the smoothness and polynomial preservation properties of the refinable basis function.

Matrix-valued subdivision derives local averaging rules for the subdivision; and the refinement equation is naturally modified to be:

$$\phi(\mathrm{x}) = \sum_{\mathrm{k}} p_k \phi(\mathrm{A}x-k), \quad x \in R^2$$

where A is the dilation matrix, the refinable function $\phi$ is extended to an r-dimensional vector valued refinable function $\phi = [\phi_0, \cdots, \phi_{r\text{-}1}]$ (called refinable function vector). And refinement sequence of r-dimensional square matrices $p_k$ are the subdivision mask. For some finite mask to be constructed, A is a $2 \times 2$ matrix with integer entries such that $|\ \det A\ | = 3 \text{ or } 4$. Examples of such matrices A are:

$$A_0=\begin{bmatrix}1 & 2\\ -2 & -1\end{bmatrix},\ A_1=\begin{bmatrix}2 & -1\\ 1 & -2\end{bmatrix},\ 2I_2=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$$

The local averaging rule is then given by:

$$v_j^{m+1}=\sum_{\mathrm{k}} v_k^m p_{j\text{-}Ak},\ \ m=0,1,\cdots$$

where $v_k^m=\left[v_{1,k}^m,\cdots,v_{n,k}^m\right]$ is a "row-vector" whose $l^{th}$ component $v_{l,k}^m$ is a parameter relative to the point in 3D domain, for $l=1,\cdots,n$. The first component $v_{l,k}^m$ is used as the geometry coordinates of the vertices of the triangulation resulting from the $m^{th}$ iterative step. $v_{l,k}^m$, $l=2,\cdots,n$ can provide the parameters for shape control of the smooth subdivision surfaces, if necessary.

For matrix-valued loop subdivision scheme [18], the matrix-valued refinable function vector is $\phi=\left[\phi_0,\phi_1\right]$, where $\phi_0$ is showed as Bezier net in **Figure 1** and $\phi_1(x)=j_0\left(\left(A_1^{-1}\right)^T x\right)$. The dilation matrix is $2I_2$. The matrix-valued subdivision provides a free parameter (shape control parameter), for adjusting shapes of surface geometry. So if we set the control parameters to be zero at each iterative step, the matrix-valued loop subdivision can generate the surface identical to the surface generated by loop subdivision scheme.

By selecting the function vector as:

$$\left[\tilde{\phi}_0+\frac{1}{2}\tilde{\phi}_1,\tilde{\phi}_1\right]^{\mathrm{T}}$$

where

$$\tilde{\phi}_0=\phi_0$$

$$\tilde{\phi}_1=\phi_1+\phi_0(-1,1)+\phi_0(-1,0)+\phi_0(0,-1)+\phi_0(1,-1)+\phi_0(1,0)+\phi_0(0,1)$$

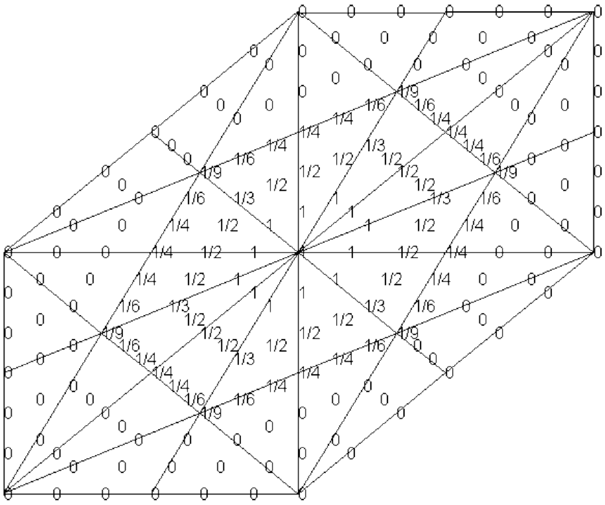


**Figure 1. The Bezier net of basis function $\phi_0$.**

Jiang *et al.* [20] extended the matrix-valued loop subdivision scheme to the interpolatory loop subdivision scheme. The refinement stencils for ordinary points are showed in the first row of **Figure 2**, where

$$G=\begin{bmatrix}1 & \frac{3}{8}\\ 0 & -\frac{1}{2}\end{bmatrix},J=\begin{bmatrix}\frac{31}{72} & -\frac{1}{36}\\ \frac{13}{36} & \frac{7}{36}\end{bmatrix},K=\begin{bmatrix}\frac{7}{72} & \frac{1}{72}\\ \frac{7}{36} & \frac{1}{36}\end{bmatrix}$$

$$L=\begin{bmatrix}0 & -\frac{1}{16}\\ 0 & -\frac{1}{8}\end{bmatrix},M=\begin{bmatrix}-\frac{1}{144} & \frac{1}{288}\\ \frac{1}{72} & \frac{1}{144}\end{bmatrix},N=\begin{bmatrix}-\frac{1}{72} & \frac{1}{144}\\ -\frac{1}{36} & \frac{1}{72}\end{bmatrix}$$

The refinement stencils for extraordinary points are showed in the second row of **Figure 2**, where

$$J_3=\begin{bmatrix}\frac{31}{72}s & 0\\ \frac{1}{4} & \frac{7}{36}\end{bmatrix},K_3=\begin{bmatrix}\frac{1}{2}-\frac{31}{72}s & 0\\ \frac{1}{4} & \frac{1}{36}\end{bmatrix}$$

$$H_n=\begin{bmatrix}1 & \frac{1}{16}\alpha\\ 0 & x_2\end{bmatrix},H=\begin{bmatrix}0 & -\frac{1}{16}\alpha\\ 0 & -\frac{27}{8(54\text{-}31s)}\alpha\end{bmatrix}$$

The value of $s$ should be chosen between 0 and $\frac{36}{31}$ and the values of $x_2$ and $\alpha$ depend on $s$. In this paper, we choose $s=\frac{36}{31}$, $x_2=-\frac{1}{4}$ and $\alpha=\frac{64}{29}$ in theory analysis and experiments.

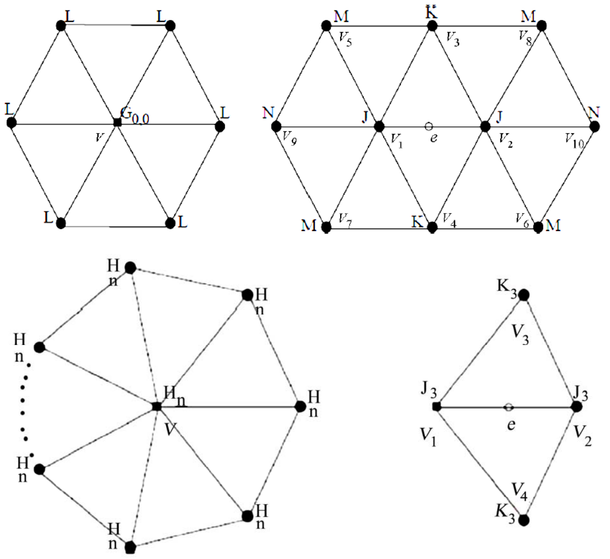


**Figure 2. The refinement stencils of interpolatory loop subdivision. The first row shows the masks for the ordinary points, and the second row shows the masks for extraordinary points.**

## 4. Biorthogonal Wavelets

### 4.1. Lazy Wavelet Transform

From the subdivision theory, we know that the subdivision can generate a series of smooth meshes from the initial mesh. For given mesh $C^n$ and fixed subdivision masks $P$, we can get a new mesh $C^{n+1}$, which has more points than $C^n$. It can be described as:

$$C^{n+1} = PC^n$$

By reversing the process, the meshes generated by the subdivision can also be easily decomposed to the simpler ones. But for an arbitrary mesh, which may be not generated by the subdivision, there will be some redundancy that cannot be decomposed after reversed subdivision. Supposing the given mesh is $C^n$, we can describe the mesh as:

$$C^n = PC^{n\text{-}1} + QD^{n\text{-}1}$$

where $C^{n-1}$ is the points in coarser level, Q is the decomposition masks, and $D^{n\text{-}1}$ are the redundancy in level $n-1$ (wavelets). How to optimize the decomposition algorithm and reduce the redundancy is the key problem we will consider in this section.

For the interpolatory subdivision, each point in $C^{n-1}$ is identical to the one in $C^n$. So it is natural to consider the "distance" between the edge points generated by subdivision and relative points in $C^{n-1}$ as the redundancy.

So, we group the points of mesh at level *n* into 2 sets: the points $V^n$, which are the vertices of triangles of mesh at level n-1 and the edge points $E^n$, which are generated from the edge of mesh at level $n-1$. Suppose $\Phi^{\mathrm{n}}$ is the basis function of subdivision scheme and $\Phi_v^n$ is the basis function relative to the point of $V^n$, $\Phi_e^n$ is the basis function relative to the point of $E^n$. The surface generated by subdivision can be described as:

$$S^n = V^n\Phi_v^n + E^n\Phi_e^n$$

For a given surface $T$, which is not generated by subdivision, just as **Figure 3** showed, we can group the points of given surface into 2 sets: $V$ (yellow points) and $E'$ (red points). We have:

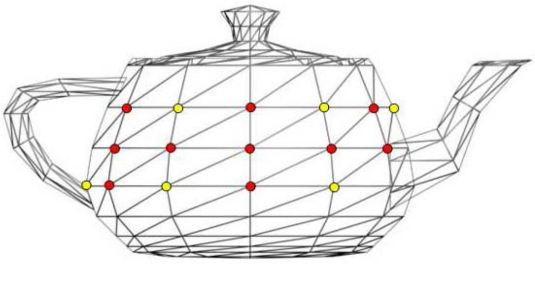


**Figure 3. The points of given surface are grouped into two sets.**

$$T = V'\Phi_v + E'\Phi_e^n$$

We consider $V$ is the set of vertices of $C^{n-1}$ and sub-divide $V$ and get the set of edge points *E*. Then:

$$T = V\mathit{\Phi}_v + E'\mathit{\Phi}_e^n = S^n + \left(E' - E^n\right)\mathit{\Phi}_e^n \qquad (1)$$

From the subdivision theory, we know that:

$$S^n = V^{n\text{-}1}\mathit{\Phi}^{n\text{-}1} \qquad (2)$$

From both (1) and (2), we get:

$$S^n = V^{n\text{-}1}\mathit{\Phi}^{n\text{-}1} + \left(E' - E^n\right)\mathit{\Phi}_e^n \qquad (3)$$

By this way, we have constructed a wavelet, which is called "lazy wavelet", to decompose the mesh from $C^n$ to $C^{n-1}$.

Thought the lazy wavelet can be used to decompose the meshes, in most cases, the lazy wavelet analysis is unstable, which means the shape of result meshes may be affected greatly by little errors after several times decomposition. The reason of instability is that the wavelet function $\phi_e^n$ of lazy wavelet and scaling function $\phi^{n-1}$ are not sure to be orthogonal. In fact the $\phi^{n-1}$ is the linear combination of $\phi_e^n$ and $\phi_v^n$. So we need to perform additional operations for the lazy wavelet transform to increase the orthogonality of wavelet function and scaling function. We construct a new wavelet $\psi'$ by accumulating the scaling functions $\phi^{n-1}$ around the edge point to the corresponding lazy wavelet $\phi_e^n$. To improve the orthogonality of wavelet transform, the new wavelet $\psi'$ should be orthogonalized with the scaling functions around it.

$$\psi' = \phi_e^n + \sum_{i=0}^{n} w_i\phi_i^{n\text{-}1} \text{ and } \left\langle \psi', \phi_i^{n\text{-}1} \right\rangle = 0. \qquad (4)$$

### 4.2. Lifting Scheme

The weights $w_i$ for lifting can be got by solving (4) if we know the inner product of lazy wavelet. There are several ways to define the inner product of wavelets. Most widely used one in classical wavelets is the continuous inner product which is often defined as:

$$\left\langle \psi, \phi \right\rangle = \sum_{\delta\hat{I}V(M)} \frac{1}{ares\left(\delta\right)} \int_{x\hat{I}\delta} \psi\left(x\right) j\left(x\right) dx$$

where $V\left(M\right)$ is the set of triangular faces of mesh $M$. From the view of mathematic theory, it is clear and convincible. But it is hard to be computed; sometimes it cannot be computed as the numeric value, especially for the basis function of subdivision.

Since the scaling function and wavelet function have been given in previous sub section, we adopt the continuous inner product for our wavelet. The basis function of matrix-valued subdivision is often represented as a Bezier net on triangular mesh. **Figure 1** showed the Bezier net of $\phi_0$, which is a basis function of matrix-valued interpolatory loop subdivision. From the Bezier net of basis function, we can write the mathematic representation of basis function. Here, since we only want to calculate the inner product, which is an integral. We can partition the triangle into thousands of blocks, multiply $\phi$ and $\psi$ at each block, then sum the results. It is more convenient to compute numeric value.

Though the computation of continuous inner product is more complex than the discrete inner product, the inner product can be precomputed as the computation is invariant of the geometry mesh. So it won't reduce the efficiency of wavelet transform.

The lifting template depends on what we need, if we want the wavelet transform more efficiency, we can select a small template. In general, it ensures the orthogonality of wavelets if we choose the lifting template which can cover the support of wavelet function. **Figure 4** shows the lifting template covers 10 points.

## 5. Efficient Wavelets Transform

### 5.1. Ordinary Points Treatment

In previous section, we have constructed a biorthogonal wavelet in theory. While, directly computing the wavelet analysis via the matrix-valued wavelets will cost a lot of computation. Here we develop an algorithm to compute fast matrix-valued subdivision wavelets for purpose of mesh simplification. The base idea of this algorithm follows the steps: 1) constructing fast algorithm for lazy wavelet analysis by reversing subdivision; 2) lifting the lazy wavelet with the weights introduced in previous section.

The most important step to construct efficient subdivision wavelets is the first one. The general method to construct the lazy wavelet needs to reform the subdivision rules and make it satisfy 2 constrains: 1) the reformed subdivision should be equal to the original subdivision when the wavelet coefficients are zero; 2) thereformed subdivision could be directly reversed, which enables us to decompose the mesh to the version of lower resolution efficiently. But, in comparison with the scalar-valued subdivision, this reforming is very hard to be applied in matrix-valued subdivision wavelets in vector space. Despite of the affection of the elements of vector to each other, the larger size of stencil to generate edge point e (comparing to the vertex v) make it hardly to be calculated without solving a global system. Since our research focus on the application on the mesh simplification and the geometry coordinates of vertex are most important results of wavelet analysis, we can simplify the subdivision by setting the initial vector as $\left[v,0\right]$, where $v$ is the geometry coordinates of vertex and the rest element of vector are zero. The matrix-valued subdivision can be reformed as:

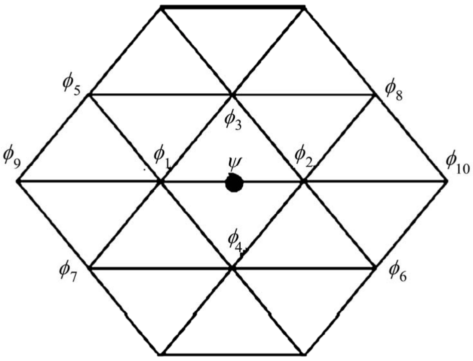


**Figure 4. Constructing a wavelet as linear combination of a lazy wavelet $\psi$ and the scaling functions $\phi_i$ (i = 0,⋯,9).**

$$\left[v',v'_x\right]=\left[v,0\right]*G+\sum_{i=1}^{6}\left[v_i,0\right]*L$$

$$\left[e',e'_x\right]=\left[e,0\right]+\sum_{i=1}^{2}\left[v_i,0\right]*K+\sum_{i=3}^{4}\left[v_i,0\right]*J \qquad (5)$$

$$+\sum_{i=5}^{8}\left[v_i,0\right]*M+\sum_{i=9}^{10}\left[v_i,0\right]*N$$

where the meaning of subscribe i are the same as they are shown in **Figure 2**. From the definition of matrix $\mathrm{G,J,K,L,M,N}$ and (4), we can get:

$$v'=v$$

$$v'_x=\frac{3}{8}v-\frac{1}{16}\sum_{i=1}^{6}v_i \qquad (6)$$

$$e'=e+\frac{31}{72}\sum_{i=1}^{2}v_i+\frac{7}{72}\sum_{i=3}^{4}v_i-\frac{1}{144}\sum_{i=5}^{8}v_i-\frac{1}{72}\sum_{i=9}^{10}v_i$$

$$e'_x=e_x+\frac{1}{72}\sum_{i=1}^{2}v_i-\frac{1}{36}\sum_{i=3}^{4}v_i+\frac{1}{288}\sum_{i=5}^{8}v_i+\frac{1}{144}\sum_{i=9}^{10}v_i$$

If $e$ and $e_x$ are initialized as zero, the expression of reformed rules (6) equals to original rules (5).

Since we focus on the geometry coordinates of points, which are the first elements, we don't care about the rest elements in mesh simplification and in most case; we even don't know the values of rest ones. We can get each $v$ directly from the fist step of (6), and $e$ can calculated from the third step of (6):

$$v=v' \qquad (7)$$

$$e=e'-\frac{31}{72}\sum_{i=1}^{2}v_i-\frac{7}{72}\sum_{i=3}^{4}v_i+\frac{1}{144}\sum_{i=5}^{8}v_i+\frac{1}{72}\sum_{i=9}^{10}v_i$$

This rule can be reverted efficiently. The first and third step of subdivision and their reverted version form a lazy wavelet, where $e$ is the wavelet coefficient and $v$ is the coefficient of scaling function. From the definition, it is can be considered as a simplified version of lazy wavelet defined by (3) in previous sub section, which effects when the data vector are $\left[v,0\right]$ at lower resolution and only compute the first element of vector. It is enough for the mesh simplification because in this situation, we only know the geometry coordinates of points, which is the first element of vector. Based on this lazy wavelet, we can develop the efficient biorthogonal wavelet decomposition algorithm for mesh simplification by performing lifting operation introduced in previous section after lazy wavelet analysis (7):

$$v = v - \sum_{i=1}^{n} w_i e_i$$

where $n$ is the size of lifting template, $e_i$ is the coefficients relative to the wavelet function in lifting template, $w_i$ is the lifting weights introduced in previous sub section.

## 5.2. Extraordinary and Boundary Points Treatment

The wavelet transform for the extraordinary points is similar to the ordinary points. Since we only focus on the geometry coordinates of points, we can only calculate the first element of vector. Then considering the data vector $\left[v,0\right]$, the subdivision can be reformed as:

$$v' = v$$

$$e' = e + \frac{31}{72} s \sum_{i=1}^{2} v_i + \left(\frac{1}{2} - \frac{31}{72} s\right) \sum_{i=3}^{4} v_i \qquad (8)$$

where $s$ can be chosen between 0 and $\frac{31}{36}$. This expression is equal to original subdivision if we set the $e$ as zero. So expression (8) and its inverse version can be considered as a lazy wavelet transform for the extraordinary points.

For the boundary points, there is no special treatment offered by the matrix-valued interpolatory subdivision. But we can consider they are special extraordinary points by setting the coefficient $s$ as $\frac{31}{36}$. In this case, the expression (8) degenerates into a simpler expression:

$$v' = v$$

$$e' = e + \frac{1}{2} \sum_{i=1}^{2} v_i \qquad (9)$$

It can be used to process the boundary points. So expression (9) and its inverse version can be considered as a lazy wavelet transform for the boundary points.

Based on these lazy wavelets, we can construct the biorthogonal matrix-valued wavelet on extraordinary points and boundary points via the lifting scheme. But, since the basis functions of these points cannot be explicit computed in mathematics, we can deploy the discrete inner product, instead of the continuous inner product. The inner product is introduced by Bertram [10] and used by several efficient subdivision based lifting wavelets to simplify the computation. The idea of discrete inner product used in subdivision wavelets is based on the assumption that the scaling functions of finer resolution form an orthogonal basis without considering all correlation of finer level coefficients. With this assumption, the mutual inner product of wavelets and scaling functions is defined as the sum of multiplications of corresponding coefficients (geometry coordinates of points) at finer resolution, and calculated directly from the subdivision template. It is maybe not accurate, but works well in many works [10-15].

**Figure 5** shows the discrete mask of extraordinary masks, where $\alpha = \frac{31}{72} s$ and $\gamma = \frac{1}{2} - \frac{31}{72} s$. Based on these discrete masks we can construct the discrete inner product between extraordinary points (or boundary points) and ordinary points, and get the weights for lifting operation. The final wavelet decomposition rules for extraordinary points can be got by performing lifting operation in addition to the lazy wavelet analysis:

$$v = v'$$

$$e = e' - \frac{31}{72} s \sum_{i=1}^{2} v_i - \left(\frac{1}{2} - \frac{31}{72} s\right) \sum_{i=3}^{4} v_i$$

$$v = v - \sum_{i=1}^{n} w_i e_i$$

where $n$ is the size of lifting template, $e_i$ is the coefficients relative to the wavelet function in lifting template, $w_i$ is the lifting weights. If we set $s = \frac{36}{31}$ and select different lifting template, it can be used to process boundary points too.

## 6. Experimental Results

In general, we can apply the different lifting template for

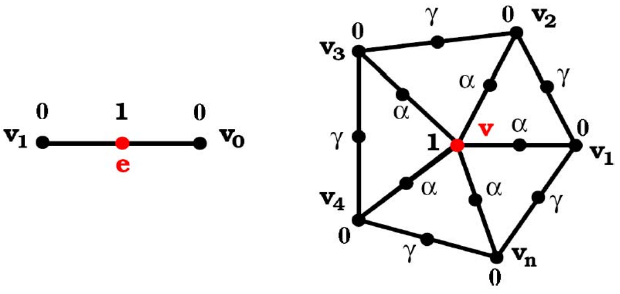


**Figure 5. The discrete masks of extraordinary points.**

various purposes. In this paper we adopt the template showed in **Figure 6**, because it exactly covers the the points to generate edge point. The weights of lifting operations template can be precomputed, which greatly increase the efficiency of transformation.

To be consistence with the optimized wavelet transform algorithm, we adopt the full version of basis function instead of the box spline used in [17]. **Table 1** shows precomputed lifting weights for ordinary points. With these weights, we test the proposed wavelet transform algorithm on the models. **Figure 7** shows the result sequence of surfaces generated by the wavelet transform.

For testing the stability of wavelet transform, we made a noise-filtering experiment, which is often used to examine the stability of approximate subdivision wavelets [10-15]. We first perturb all vertices of the mesh at highest resolution with white noise. The perturbed mesh is decomposed step by step using the wavelet analysis 5 times. At each resolution, we subdivide the mesh to level 5 (the highest resolution) again without considering the wavelet coefficients of any higher resolution. Thus, we get a sequence of low-pass filtered versions of the noisy mesh. We compare the low-pass filtered versions of noisy mesh with the original unperturbed mesh by calculating the corresponding $L_2$-norm errors. The results of experiment are showed in **Figure 7** and **Figure 8**. From these figures, we can find that that the error rates don't increase much after few wavelet analysis operations. So the decomposition is quite stable.

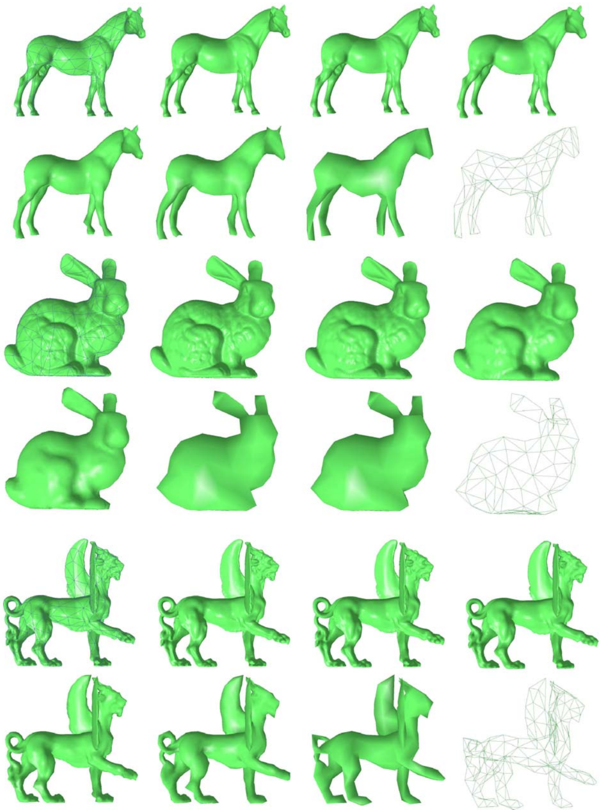


**Figure 6. Multiresolution mesh models (horse, bunny, feline): the 1st surfaces show the original models rendered with wireframe. Others are surfaces generated by performing wavelet analysis 0-5 times, and the last one shows the lowest-resolution model in wireframe rendering.**

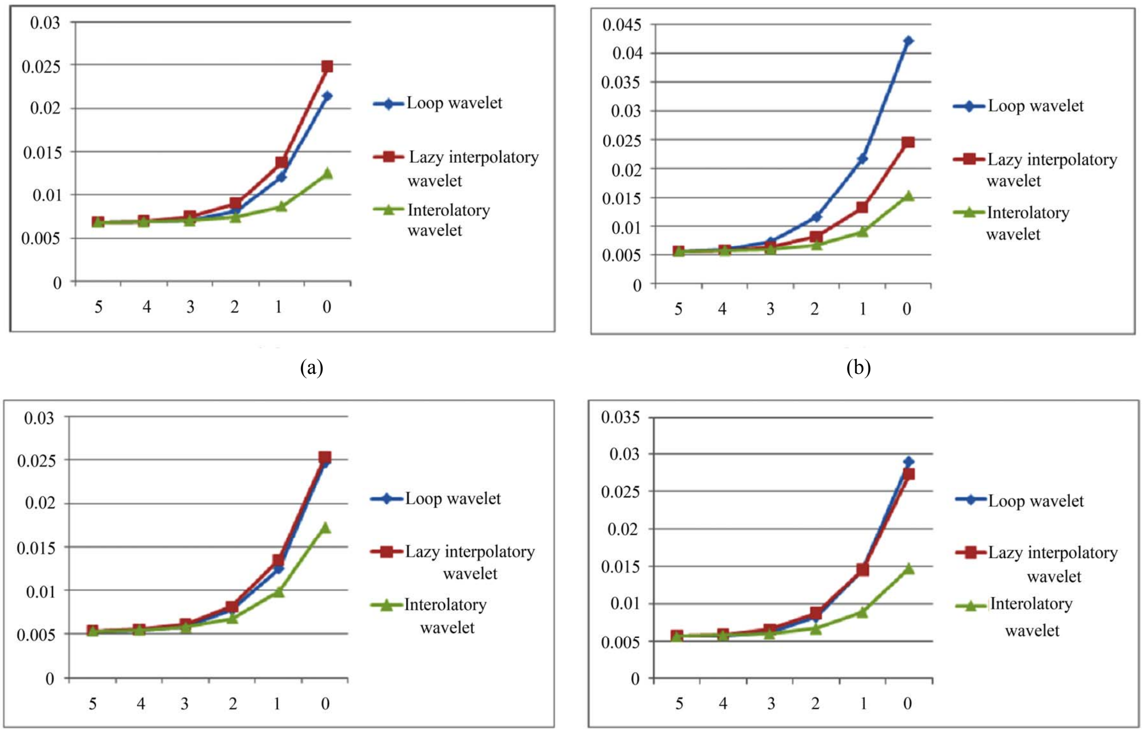
**Table 1. The precomputed weights of local lifting operations, when the valence n = 6.**

| w2 | w3 | w4 | w5 | w6 |
|---|---|---|---|---|
| **–0.17712** | –0.17712 | 0.0238 | 0.0238 | –0.0016 |
| w7 | w8 | w9 | w10 | w11 |
| **–0.0016** | 0.0249 | 0.0249 | 0.0249 | 0.0249 |

For comparison with other wavelet transform, we made the same tests on approximate loop subdivision wavelet, developed by Bertram [10], and lazy interpolatory subdivision wavelet. Since the lifting template of Bertram's scheme only includes 4 points around the edge point. We use our template, including 10 points around the edge point, instead of original one. The more points in template make the wavelet function orthogonal to more scaling functions and the fitting quality of result should be better. All the results of noise-filter experimental results are showed in **Figure 7** and **Figure 8**. These experimental results show that: compared with loop subdivision wavelet and lazy interpolatory subdivision wavelet, the interpolatory subdivision wavelet has better performance in noise reduction. While, we should pay attention to the fact that, though the $L_2$-norm error of loop wavelet analysis is much more than interpolatory subdivision wavelet, the surfaces generated by loop subdivision wavelet transform seem smoother than interpolatory subdivision wavelet transform after subdividing. It caused mainly by the awful ability of approximate subdivision to generate smooth surfaces.

We also compared the fitting quality of meshes, which come from loop subdivision wavelet analysis and our interpolatory subdivision wavelet analysis individually.

**Figure 9** shows the results generated by computing wavelet analysis 4 times. The blue circles are used to mark the obvious difference between the results. Since the approximate subdivision averages both the vertices of triangle and edge points, the surfaces are smoother in general. But when performing surface decomposition, the change from eliminated points also affect the residual points and make the result surface bulge and may generate surface with ripples. The interpolation subdivision keeps the points of original surface, which fixes the basic shape of subdivision surface. Because the subdivision wavelets inherit the properties of subdivision, though the lifting operation changed the values of points for both wavelets, the affection to the interpolatory wavelet is less



(a) (b)

(c) (d)

**Figure 7. The comparison of L$_2$-norm errors by noise-filtering experiments on: (a) venus; (b) horse; (c) bunny; (d) feline.**

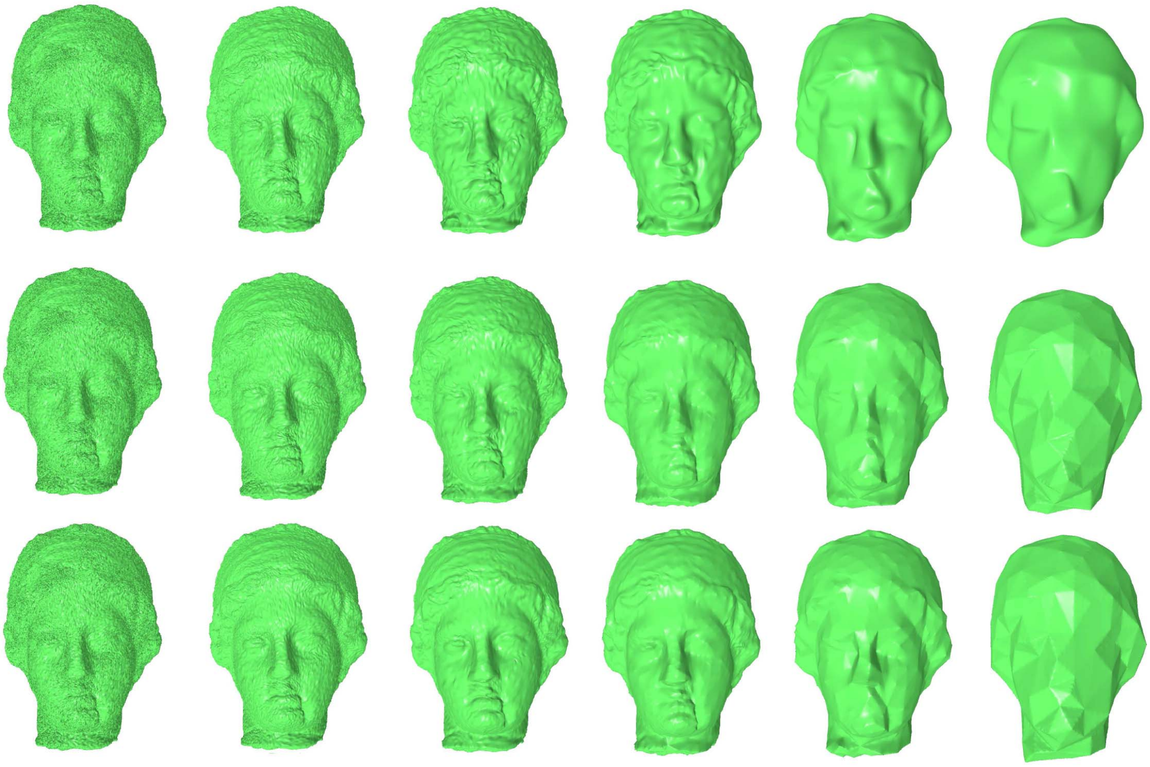


**Figure 8. The low-pass filter sequence of noisy meshes from level 5 to level 0: the first row shows the results by approximate loop subdivision wavelet; the second row shows the results by lazy interpolatory loop subdivision wavelet; the third row shows the results by lifted interpolator loop subdivision wavelet.**

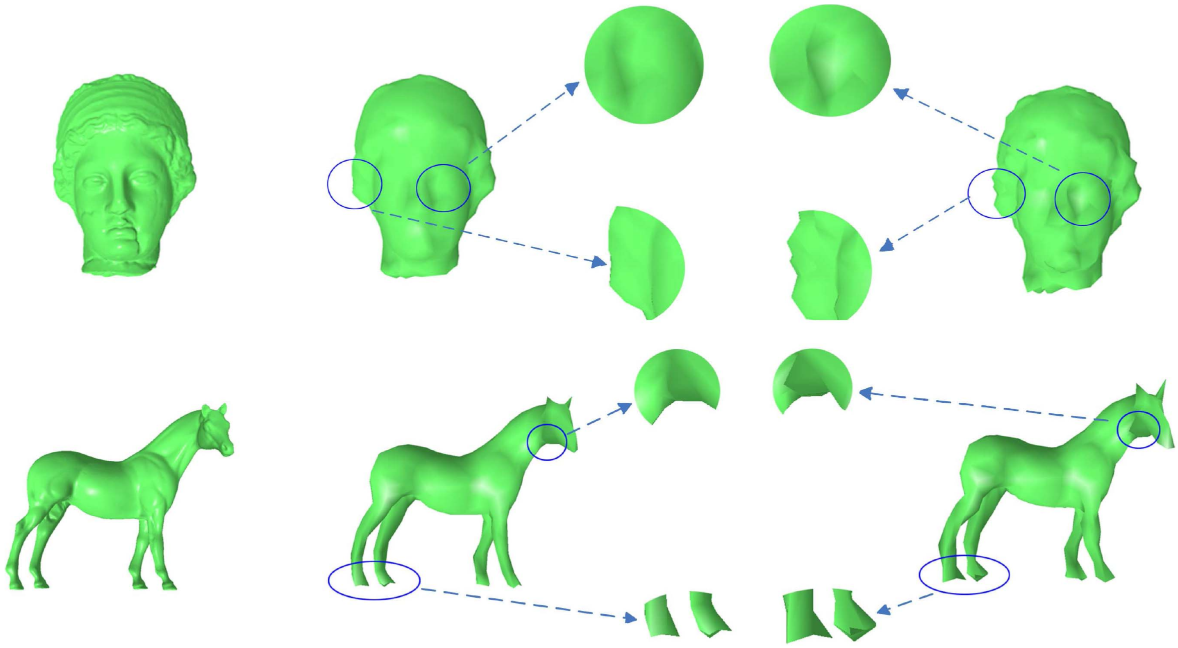


**Figure 9. Multiresolution surfaces by loop subdivision wavelet analysis and interpolatory wavelet analysis. The 1st column shows the original models; the 2nd column shows the results decomposed by interpolatory subdivision wavelet analysis 4 times; the 3rd column shows the results decomposed by loop subdivision wavelet analysis 4 times. Here, blue circles used to enlarge the viewing differences between them. As for loop subdivision, there are ripples in the surfaces generated by loop subdivision wavelets analysis after several times decomposing. The interpolatory subdivision wavelet analysis we propose plays better on avoiding these defects.**

than it to the approximate subdivision wavelets, according to the results of experiments. So the interpolatory subdivision wavelet should have better shape preserve ability than the approximate subdivision wavelet in visual appearance. This conforms to the fact that $L_2$-norm error of the interpolatory subdivision wavelet is lower than the approximate subdivision wavelet.

We tested the efficiency of wavelet transform by using a PC equipped with Intel Core(TM)2 Quad CPU Q8200 at 2.33 HZ and 4 G memory. The results are listed in **Table 2**. Because the lifting operation is executed on each vertex and the time complexity of each operation is O(1), the time complexity of lifting operations of wavelet transform only depends on the number of vertices. Since it avoids solving a complex system, the proposed transform performs efficiently for multiresolution surfaces, in both wavelet analysis and synthesis of mesh models.

**Table 2. The time cost of performing wavelet transform 5 times. Curent rows show the time cost of algorithm in this paper and previous row show the time cost of algorithm in [17].**

| **Analysis** | Horse<br>(112642 pt) | Venus<br>(198658 pt) | Feline<br>(258046 pt) |
|---|---|---|---|
| **Current** | 0.08550 sec. | 0.13247 sec. | 0.19051sec. |
| **Previous** | 0.11395 sec. | 0.200944 sec. | 0.258138 sec. |
| **Synthesis** | Horse | Venus | Feline |
| **Current** | 0.07002 sec. | 0.11976 sec. | 0.18068 sec. |
| **Previous** | 0.107769 sec. | 0.189748 sec | 0.245778 sec. |

## 7. Summary

In this paper, we propose the novel wavelet transform based on matrix-valued interpolatory loop subdivision for multiresolution triangular meshes. Since the matrixvalued subdivision is directly generated from the basis function vector, it is easy to be used to construct lazy wavelet. For better fitting quality, the additional lifting operations are applied to increase the orthogonality of wavelet. Further from our initial work [17], we have worked on the follo- wing major features: To speed up the wavelets transform, we work out and optimize the algorithm of interpolatory loop subdivision wavelets in detail. Our approach can deal with the extraordinary points and boundary points faithfully. We have designed the full version of basis function of interpolatory subdivision, instead of the simpler version. So the lifting weights can be used for the optimized wavelets transform we propose.

By applying these methods, the final transform is efficient, and has low memory usage because no additional memory used in the processing of points. The computation is fully in-place and efficient. The testing experiments showed that the wavelets transform we develop is stable, and has good performance on noise reduction and fitting quality. Our proposed wavelets transform can be applied in a wide range of applications, including 3D-model progressive transmission, data compression, multiresolution rendering, and interactive geometric editing. The construction method we develop is easy to be extended to other matrix-valued interpolatory subdivision schemes for triangular and quadrilateral meshes. In the future work, we will focus on how to eliminate the defect of interpolatory wavelet when generating simplified smooth surface. We will try to construct the transform that has the advantages of interpolatory and approximate subdivision wavelets for such applications.

## 8. Acknowledgements

This work was supported by RGC research grant (Ref. 415806) and UGC direct grant for research (No. 2050423, 2050454).

## 9. References

[1] M. Lounsbery, T. DeRose and J. Warren, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type," *ACM Transaction of Graphics*, Vol. 16, No. 1, 1997, pp. 34-73.

[2] S. Valette and R. ProstS, "Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 10, No. 2, 2004, pp. 113-122. doi:10.1109/TVCG.2004.1260763

[3] S. Valette and R. ProstS, "Wavelet-Based Progressive Compression Scheme for Triangle Meshes: Wavemesh," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 10, No. 2, 2004, pp. 123-129. doi:10.1109/TVCG.2004.1260764

[4] F. F. Samavati and R. H. Bartels, "Multiresolution Curve and Surface Editing: Reversing Subdivision Rules by Least-Squares Data Fitting," *Computer Graphics Forum*, Vol. 18, No. 2, 1999, pp. 97-119. doi:10.1111/1467-8659.00361

[5] F. F. Samavati, N. Mahdavi-Amiri and R. H. Bartels, "Multiresolution Surfaces Having Arbitrary Topologies by a Reverse Doo Subdivision Method," *Computer Graphics Forum*, Vol. 21, No. 2, 2002, pp. 121-136. doi:10.1111/1467-8659.00572

[6] W. Sweldens, "The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets," *Applied and Computational Harmonic Analysis*, Vol. 3, No. 2, 1996, pp. 186-200. doi:10.1006/acha.1996.0015

[7] P. Schroder and W. Sweldens, "Spherical Wavelets: Efficiently Representing Functions on the Sphere," *In SIGGRAPH*'95*: Proceedings of the* 22*nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, 6-11 August 1995, pp. 161-172.

[8] M. Bertram, M. Duchaineau, B. Hamann and K. Joy, "Bicubic Subdivision-Surface Wavelets for Large-Scale Isosurface Representation and Visualization," *In VIS*'00: *Proceedings of the Conference on Visualization*'00, Salt Lake City, 8-13 October 2000, pp. 389-396.

[9] M. Bertram, M. Duchaineau, B. Hamann and K. Joy, "Generalized b-Spline Subdivision-Surface Wavelets for Geometry Compression," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 10, No. 3, 2004, pp. 326-338. doi:10.1109/TVCG.2004.1272731

[10] M. Bertram, "Biorthogonal Loop-Subdivision Wavelets," *Computing*, Vol. 72, No. 1-2, 2004, pp. 29-39. doi:10.1007/s00607-003-0044-0

[11] D. Li, K. Qin and H. Sun, "Unlifted Loop Subdivision Wavelets," *In PG'*04*: Proceedings of the Computer Graphics and Applications,* 12*th Pacific Conference, IEEE Computer Society*, 2004, pp. 25-33.

[12] H. Wang, K. Qin and K. Tang, "Efficient Wavelet Construction with Catmull-Clark Subdivision," *Visual Computer*, Vol. 22, No. 9, 2006, pp. 874-884. doi:10.1007/s00371-006-0074-7

[13] H. Wang, K. Qin and H. Sun, "$\sqrt{3}$-Subdivision-Based Biorthogonal Wavelets," *IEEE Transaction on Visualization and Computer Graphics*, Vol. 13, No. 5, 2007, pp. 914-925. doi:10.1109/TVCG.2007.1031

[14] H. Wang, K. Tang and K. Qin, "Biorthogonal Wavelets Based on Gradual Subdivision of Quadrilateral Meshes," *Computer Aided Geometric Design*, Vol. 25, No. 9, 2008, pp. 816-836. doi:10.1016/j.cagd.2007.11.002

[15] H. Wang, K. Tang and K. Qin, "Biorthogonal Wavelets Based on Interpolatory p2 Subdivision," *Computer Graphics Forum*, Vol. 28, No. 6, 2009, pp. 1572-1585. doi:10.1111/j.1467-8659.2009.01349.x

[16] H. Zhang, G. Qin, K. Qin and H. Sun, "A Biorthogonal Wavelet Approach Based on Dual Subdivision," *Computer Graphics Forum*, Vol. 27, No. 7, 2009, pp. 1815-1822. doi:10.1111/j.1467-8659.2008.01327.x

[17] C. Zhao, H. Sun and K. Qin, "Computing Efficient Matrix-Valued Wavelets for Meshes," *Pacific Conference on Computer Graphics and Applications*, Hangzhou, Vol. 0, 25-27 September 2010, pp. 32–38.

[18] C. K. Chui and Q. Jiang, "Surface Subdivision Schemes Generated by Refinable Bivariate Spline Function Vectors," *Applied and Computational Harmonic Analysis*, Vol. 15, 2003, pp. 147-162. doi:10.1016/S1063-5203(03)00062-9

[19] C. K. Chui and Q. Jiang, "Matrix-Valued Symmetric Templates for Interpolatory Surface Subdivisions, i: Regular Vertices," *Applied and Computational Harmonic Analysis*, Vol. 19, 2005, pp. 303-339. doi:10.1016/j.acha.2005.03.004

[20] C. K. Chui and Q. Jiang, "From Extension of Loop's Approximation Scheme to Interpolatory Subdivisions," *Computer Aided Geometric Design*, Vol. 25, No. 2, 2008, pp. 96-115. doi:10.1016/j.cagd.2007.05.004