



HAL
open science

Subdivision Invariant Polynomial Interpolation

Stefanie Hahmann, Georges-Pierre Bonneau, Alex Yvart

► **To cite this version:**

Stefanie Hahmann, Georges-Pierre Bonneau, Alex Yvart. Subdivision Invariant Polynomial Interpolation. International Workshop on Visualization and Mathematics (Vismath 2002), May 2002, Berlin-Dahlem, Germany. pp.191–202. inria-00537477

HAL Id: inria-00537477

<https://inria.hal.science/inria-00537477>

Submitted on 18 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Subdivision Invariant Polynomial Interpolation

Stefanie Hahmann, Georges-Pierre Bonneau*, Alex Yvart

Laboratoire LMC-IMAG, BP.53
F-38041 Grenoble cedex 9, France
email: Stefanie.Hahmann@imag.fr, Alex.Yvart@imag.fr

*Laboratoire iMAGIS-GRAVIR
INRIA Rhône-Alpes
655, avenue de l'Europe
F-38330 Montbonnot, France
email: Georges-Pierre.Bonneau@imag.fr

Abstract. *In previous works a polynomial interpolation method for triangular meshes has been introduced. This interpolant can be used to design smooth surfaces of arbitrary topological type. In a design process, it is very useful to be able to locate the deformation made on a geometric model. The previously introduced interpolant has the so-called strict locality property: when a mesh vertex is changed, only the surface patches containing this vertex are changed. This enables to locate the deformation at the size of the input triangles. Unfortunately this is not sufficient if the designer wants to add some detail at a smaller size than that of the input triangles. In this paper, we propose a modification of our interpolant, that enables to arbitrary refine the input triangulation, without changing the resulting surface. We call this property the subdivision invariance. After refinement of the input triangulation, the modification of one of the vertices will change the shape of the interpolant at the scale of the refined triangulation. In this way, it is possible to add details at an arbitrary fine scale.*

1. INTRODUCTION

Designing smooth surfaces of arbitrary topological type has attracted considerable attention in the last 20 years. Standard NURBS surfaces [1, 5, 8] are restricted to model surfaces homeomorphous to a disc. Basically two research directions have been followed in order to overcome this restriction. On the one hand, previous works have tried to interpolate or

approximate arbitrary topological type polyhedron, using parametric polynomial surfaces. More recently, subdivision surfaces, already discovered in the mid 70's, have become very popular in the computer graphic community. These surfaces are defined as the limit of a refinement process starting on an initial control polyhedron. We will come in more detail on these related works in the next section. The present paper follows the first research direction: we define a smooth (tangent plane continuous) polynomial surface that interpolates a 2D-manifold triangulation of arbitrary topological type.

In comparison to our previous works on smooth polynomial interpolation [3, 4], this paper adds the following novelties and improvements:

- the concept of subdivision invariance is introduced,
- this concept is illustrated in the curve case,
- the interpolant of [4] is modified in order to fulfill the subdivision invariance property,
- examples of smooth surfaces interpolating locally refined triangulations of non tensor product topology are given.

The subdivision invariance property states in essence that, for some refinement (subdivision) of the input mesh, the interpolant doesn't change. More precisely, if the same interpolation method is applied on the original mesh, or on the refined mesh, the resulting smooth polynomial surfaces are identical. This property enables to refine an input mesh before editing it. In this way the designer can add details at arbitrary positions, and at arbitrary scale. If she/he wants to modify very locally the model, she/he only has to refine the input control mesh sufficiently until the size of the triangles is smaller than the size of the part of the surface that she/he wants to modify. However, the subdivision invariance property would not be useful in practice, if it would be necessary to entirely refine the input mesh before applying the interpolation method. Fortunately, since the interpolation method we are talking about in this paper is strictly local, meaning that a modification of one vertex of the input mesh modifies only the patches having this vertex in common, it is possible to refine locally the triangulation instead of refining it globally. Altogether, the subdivision invariance and the strict locality enable to compactly model smooth parametric polynomial surfaces with arbitrary fine details.

The remaining of the paper is structured as follows. In section 2, we discuss related works. In section 3 the concept of subdivision invariance is introduced and illustrated in the simple case of curve interpolation. Section 4 introduces the subdivision invariant interpolant of arbitrary triangulations. Results are given in section 5. We conclude in section 6, and give some possible future works.

2. RELATED WORKS

The literature on geometric design of arbitrary topological type smooth surfaces can be

divided into two groups: those that deal with polynomial surfaces and those dedicated to subdivision surfaces.

Although subdivision surfaces were discovered already in the 70's, they became popular only recently. In comparison active research on polynomial surfaces of arbitrary topological type has been conducted earlier. Much of those works are concerned with polynomial interpolation of 2D-manifold triangulations. Piper [9], Shirman-Sequin [11] and Jensen [6] have used Clough-Tocher like splitting of the input triangles in order to develop a smooth G^1 continuous interpolant of low degree. The splitting process consists in inserting a new vertex in the interior of each input triangle, and inserting new edges between this vertex and the triangle vertices. Thus each input triangle is divided into three sub-triangles. In comparison, the present interpolant is based on a regular 4-split of the input triangles. This is clearly fully appropriate for successive refinements. Recursively splitting a triangle Clough-Tocher like very quickly leads to degenerate triangles, while the regular 4-split can be repeated any given number of times without flattening the triangle. Loop [7] has proposed a G^1 continuous surface that approximates a 2D-manifold triangulation. In comparison, the present method is dedicated to interpolation, and not approximation, and it is of lower degree.

Subdivision surfaces nowadays tend to become a de facto standard for the modeling of smooth surfaces of arbitrary topological type. Works on subdivision surfaces are too numerous to be referred here. A complete survey on subdivision surfaces can be found in [10]. Clearly subdivision surfaces - by construction - are perfectly adapted to the design of surfaces with details at any scale. They are naturally subdivision invariant. But subdivision surfaces do not have an explicit parameterization, while our interpolant has an explicit, low degree polynomial parameterization. In some application areas, for example in the CAD/CAM industry, it is very important to know an explicit parameterization of a surface. This is probably the reason why subdivision surfaces have not yet been integrated in CAD/CAM software.

3. SUBDIVISION INVARIANCE

An interpolating scheme can be viewed as a map that takes as input a discrete set of data, and parameters associated to these data. It outputs a continuously defined function that takes the prescribed data values at the corresponding parameters. This can be stated more mathematically using the following notations:

$$I : (d_i \in E, t_i \in \Omega)_{i \in \mathcal{D}} \quad \longrightarrow \quad f : \Omega \rightarrow E$$

$$f(t_i) = d_i \quad \forall i \in \mathcal{D}$$

Ω is the parameter domain containing all parameter values t_i . E is a set containing the data to be interpolated d_i . \mathcal{D} is a discrete index set.

A subdivision scheme can be viewed as a map that takes as input a discrete set of data, and that outputs another discrete set of data that is typically more dense than the input set. This can be stated using the following notations:

$$S : (d_i \in E, t_i \in \Omega)_{i \in \mathcal{D}} \quad \longrightarrow \quad (e_j \in E, s_j \in \Omega)_{j \in \mathcal{E}},$$

where \mathcal{D} and \mathcal{E} are two discrete index sets, and \mathcal{D} is included in \mathcal{E} . The subdivision scheme is said to be interpolating if $s_i = t_i$, $e_i = d_i$ for all $i \in \mathcal{D}$.

Definition : I is invariant with respect to S iff

$$I \circ S = I.$$

In other words, I is invariant relative to S iff it yields the same result before and after applying subdivision S .

If the interpolation mapping is continuous, and the subdivision scheme is converging towards a continuous limit, then the following identity holds

$$I = S^\infty.$$

Thus given any subdivision scheme, there is a unique interpolation mapping that is invariant relative to it. The converse is not true: given an interpolation mapping, there may be different subdivision schemes relative to which the interpolation mapping is invariant.

Hermite curve interpolation

We illustrate the subdivision invariance property on a simple curve interpolation example. The Hermite interpolation scheme H takes as input a set of points (position and derivatives) and parameters associated to these data, and outputs the unique piecewise cubic C^1 continuous polynomial function that interpolates these data.

Let $(d_i, d'_i)_{i=0, \dots, n}$ denote the position and derivatives to interpolate, and $(t_i)_{i=0, \dots, n}$ the corresponding parameter values. Let f denote the Hermite interpolant. f is the unique piecewise cubic C^1 continuous polynomial such that $f(t_i) = d_i$ and $f'(t_i) = d'_i$. Using f , we can easily define a subdivision scheme S - in that case a Hermite subdivision scheme - relative to which the Hermite interpolant is invariant: Simply define

$$S_H(d_i, d'_i, t_i) = (e_j, e'_j, s_j),$$

where

$$(e_{2i}, e'_{2i}, s_{2i}) = (d_i, d'_i, t_i), \text{ and } (e_{2i+1}, e'_{2i+1}, s_{2i+1}) = \left(f\left(\frac{t_i+t_{i+1}}{2}\right), f'\left(\frac{t_i+t_{i+1}}{2}\right), \frac{t_i+t_{i+1}}{2} \right).$$

In other words S is an interpolatory subdivision scheme, and the newly inserted data values are computed using the Hermite interpolant. It is almost trivial to prove that H is invariant relative to S_H . Figure 1 shows two pieces of the interpolant to the left, and at the middle the resulting three pieces after subdivision of the first segment. Since each of the two first pieces after subdivision has the same position and derivative at their end point than the original polynomial curve, and since all these polynomials are cubic, they must be identical.

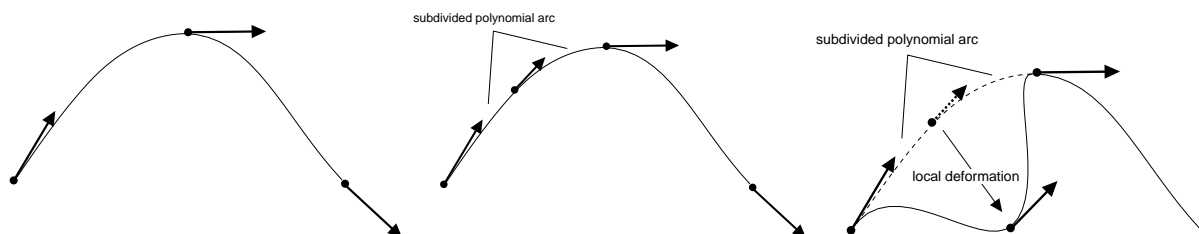


figure 1: *Hermite interpolation - subdivision - local deformation.*

Because H is invariant relative to S_H , and since H is a strictly local interpolation scheme, it is possible to add local details in the following way: use S_H in order to locally refine the input data around the region that must be modified, edit the subdivided data in this region, interpolate this new data using H . The subdivision invariance ensures that the resulting smooth curve differs from the original one only in the region of interest. This process is illustrated in figure 1.

4. SUBDIVISION INVARIANT G^1 POLYNOMIAL TRIANGULAR INTERPOLANT

This section begins by a brief recall on a previous work on G^1 polynomial triangular interpolation (section 4.1). Then section 4.2 will give the improvements made on this interpolant in order to satisfy the subdivision invariance property.

4.1 G^1 polynomial triangular interpolation

In [4], a G^1 polynomial interpolation method for 2D-manifold triangulations has been introduced. In this section, we briefly recall the main points of this interpolant, useful for the remaining of the paper.

The interpolant maps each input triangle onto four Bézier patches of degree 5, each of them is parameterized over one of the 4 triangles obtained by the regular 4-split of the input triangle. Figure 2 illustrates the parameterization of the interpolant. Each group of 4 Bézier patches corresponding to one input triangle is called a macro-patch.

The interpolant is build in three steps

- the boundary curves of the macro-patch are determined,

- the first rows of control points on both sides of the boundary curve's control points are constructed. These are the control points from which the tangent planes along the boundary curves are computed. Thus they entirely control the G^1 continuity of the interpolant across the boundary curves, and therefore between the macro-patches.
- the remaining interior control points for each macro-patch are computed so that the 4 Bézier patches connect with C^1 continuity.

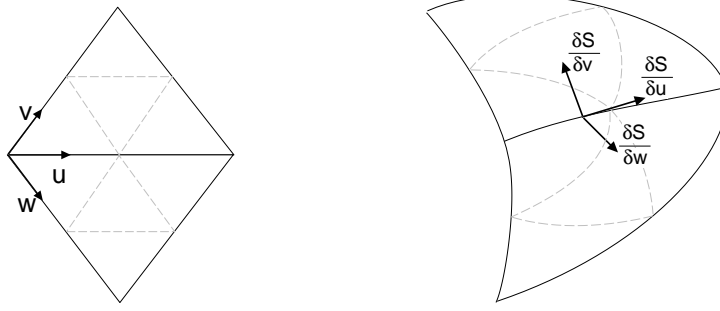


figure 2: *Parameterization of the interpolant. left, parameter domain; right, macro-patches with tangent directions.*

There are many degrees of freedom in this interpolant. The main degrees of freedom are

- the derivatives at the boundary curve's end points, and
- the twists at the mesh vertices.

Once the derivatives and the twists are chosen, the second derivatives of the boundary curves are directly fixed by the twist compatibility condition.

The main equation that governs the surface construction is the G^1 continuity condition between two patches:

$$\Phi(u) \frac{\partial S}{\partial u} = \mu(u) \frac{\partial S}{\partial v} + \nu(u) \frac{\partial S}{\partial w}. \quad (1)$$

Φ , μ and ν are piecewise linear functions defined along the common boundary curve. S is the interpolating surface. $\frac{\partial \cdot}{\partial u}$, $\frac{\partial \cdot}{\partial v}$, $\frac{\partial \cdot}{\partial w}$ are the three derivative operators in the parametric directions shown in figure 2. G^1 condition (1) states that the three tangent vectors $\frac{\partial S}{\partial u}$, $\frac{\partial S}{\partial v}$, and $\frac{\partial S}{\partial w}$ are coplanar, thus they define a common tangent plane along the boundary curve. In order to find a polynomial solution to the G^1 condition (1), the boundary curve $S(u)$ is chosen that

$$\frac{\partial S}{\partial u} = \mu(u) \nu(u) H(u) \quad \text{along the boundary curve,} \quad (2)$$

where H is a piecewise continuous degree 2 polynomial. H (and S along the boundary) is uniquely defined by the choice of the first and second derivative of the boundary curves at the end points.

Once S is fixed along the boundary curves, the cross-derivatives $\frac{\partial S}{\partial v}$ and $\frac{\partial S}{\partial w}$ are computed by:

$$\begin{cases} \frac{\partial S}{\partial v} = \frac{1}{2}(\Phi(u)\nu(u)H(u)) + \nu(u)W(u) \\ \frac{\partial S}{\partial w} = \frac{1}{2}(\Phi(u)\mu(u)H(u)) - \mu(u)W(u) \end{cases} \quad (3)$$

It is trivial to see that (3) implies (1): simply multiply the first equation in (3) by $\mu(u)$, the second equation in (3) by $\nu(u)$, and add them up.

In [4] W has been chosen as a piecewise C^0 degree 2 polynomial. But it could have been chosen as well chosen as a cubic, without increasing the degree of the interpolant.

After having computed the boundary curves and the cross boundary derivatives using (2) and (3), 15 control points remain to choose inside each macro-patch (see fig. 3). 6 out of these 15 can be freely chosen, the other 9 control points are fixed by the C^1 continuity conditions between the 4 Bézier patches.

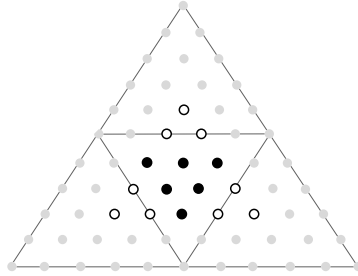


figure 3: control points of a macro-patch which is composed of four quintic Bézier triangles. The 15 inner control points are highlighted. The 6 free control points are shown as black dots.

4.2 Subdivision invariance

This section first gives the subdivision operator relative to which the previously introduced interpolation scheme will be made invariant. Then it is shown how the subdivision invariance may be fulfilled.

Subdivision operator

The subdivision operator takes as input the position, the n first derivatives along the boundary curves, and the n twists at each mesh vertex of valence n . Recall (section 4.1) that these quantities are among the degrees of freedom of the interpolation operator. Figure 4 shows to the left the input of the subdivision operator. The subdivision operator outputs the same data (position, first derivatives and twists) at the input mesh vertices and at the surface points corresponding to the midpoints of each edge (fig. 4-right). In order to compute this subdivision operator, the interpolating surface has first to be computed, and then it must be evaluated at the midpoints of each edge.

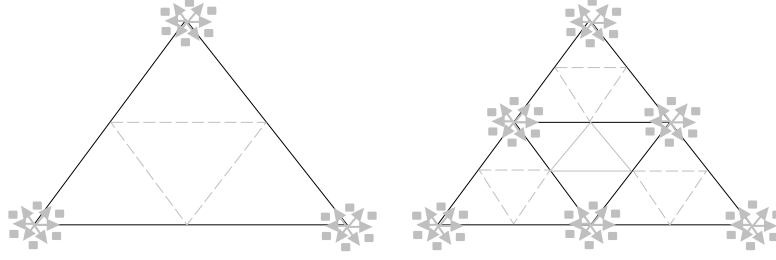


figure 4: subdivision operator.

Subdivision invariant interpolation

Beside the first derivatives and twists at the mesh vertices, the other degrees of freedom of the interpolation operator are:

- the values of the (piecewise linear C^0) functions Φ , μ , and ν at the edge midpoints
- the piecewise cubic polynomial function W along the edges. The constraints on W are its position and derivative at the end points, and the C^0 continuity at the edge midpoint.
- the 6 free inner control points.

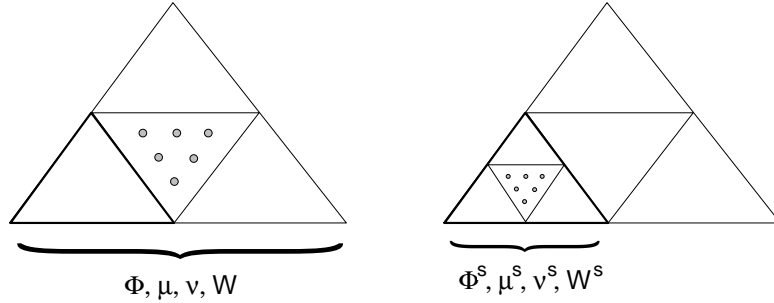


figure 5: degrees of freedom of the interpolant before (left) and after(right) subdivision.

All these degrees of freedom were previously chosen using heuristics. For example the 6 free inner control points were computed by a least squares energy minimization. The key idea to define a subdivision invariant interpolant is to choose these degrees of freedom such that they are not modified by the subdivision operator. Figure 5 shows these degrees of freedom for one macro-patch before (left) and after (right) the subdivision operator. We will concentrate on the bottom left Bézier patch of the original macro-patch (highlighted in fig. 5-left). Let Φ , ν , μ , W denote the functions controlling the G^1 continuity along the bottom edge of the macro-patch before subdivision. Let Φ^s , μ^s , ν^s , W^s denote the functions controlling the G^1 continuity along the bottom edge of the bottom left Bézier patch after subdivision. We have to choose the degrees of freedom so that $\Phi = \Phi^s$, $\mu = \mu^s$, $\nu = \nu^s$, $W = W^s$ along their common interval of definition. This property is fulfilled if and only if all the functions are chosen as *one* polynomial piece, e.g. the piecewise linear functions must be chosen linear: $[\Phi, \mu, \nu](\frac{1}{2}) = \frac{1}{2}([\Phi, \mu, \nu](0) + [\Phi, \mu, \nu](1))$ and W must be the

cubic Hermite interpolant of $W(0)$, $W'(0)$, $W(1)$, $W'(1)$ (instead of piecewise quadratic as in the previous method). It remains to fix the 6 free inner control points. In order to do this, we use the Bézier subdivision algorithm of Goldman [2] to 4-split the original Bézier patches. Then we simply copy the 6 inner control points of this Bézier patch in the corresponding Bézier patch of the subdivided macro-patch.

5. RESULTS

Figure 6 shows three local refinements of an input icosahedron mesh (left column), the smooth polynomial surface interpolating this triangulation (middle column), and the Bézier control mesh (right column). Instead of globally refining the input mesh, only one edge is splitted at each subdivision step, and the two neighbouring triangles are 4-splitted. The reader may notice discontinuities in the pictures in the left and right columns. This doesn't mean that the interpolating surface is not tangent plane continuous: indeed the middle column clearly illustrates the G^1 continuity of this surface. The gaps in the left and right columns are due to the fact that the same boundary curve is represented by control polygons at different subdivision levels. Thus although both control polygons define the same curve, they are not identical.

Figure 7 illustrates the edition of a vertex in the original model (top) and at the finest level of subdivision (bottom). The locality of the deformations can be clearly seen in this example.

6. CONCLUSION

In this paper the concept of subdivision invariant interpolation was introduced. This concept is very useful in practice, since it tells exactly when an interpolation scheme can be used to add details at any scale on a given geometric model. The subdivision invariance has been illustrated on a simple curve interpolation scheme. Then the previously introduced triangular polynomial interpolant [4] has been modified in order to fulfill the subdivision invariance property. Examples were given that illustrate the ability to add detail on a smooth surface of arbitrary topological type. In future works, the subdivision invariant interpolation scheme will serve as a basis for the definition of a multiresolution smooth surface representation based on polynomial patches of low degree.

Acknowledgements

This work was partially supported by the European Community 5-th framework program, with the Research Training Network MINGLE (Multiresolution IN Geometric modELing, HPRN-1999-00117).

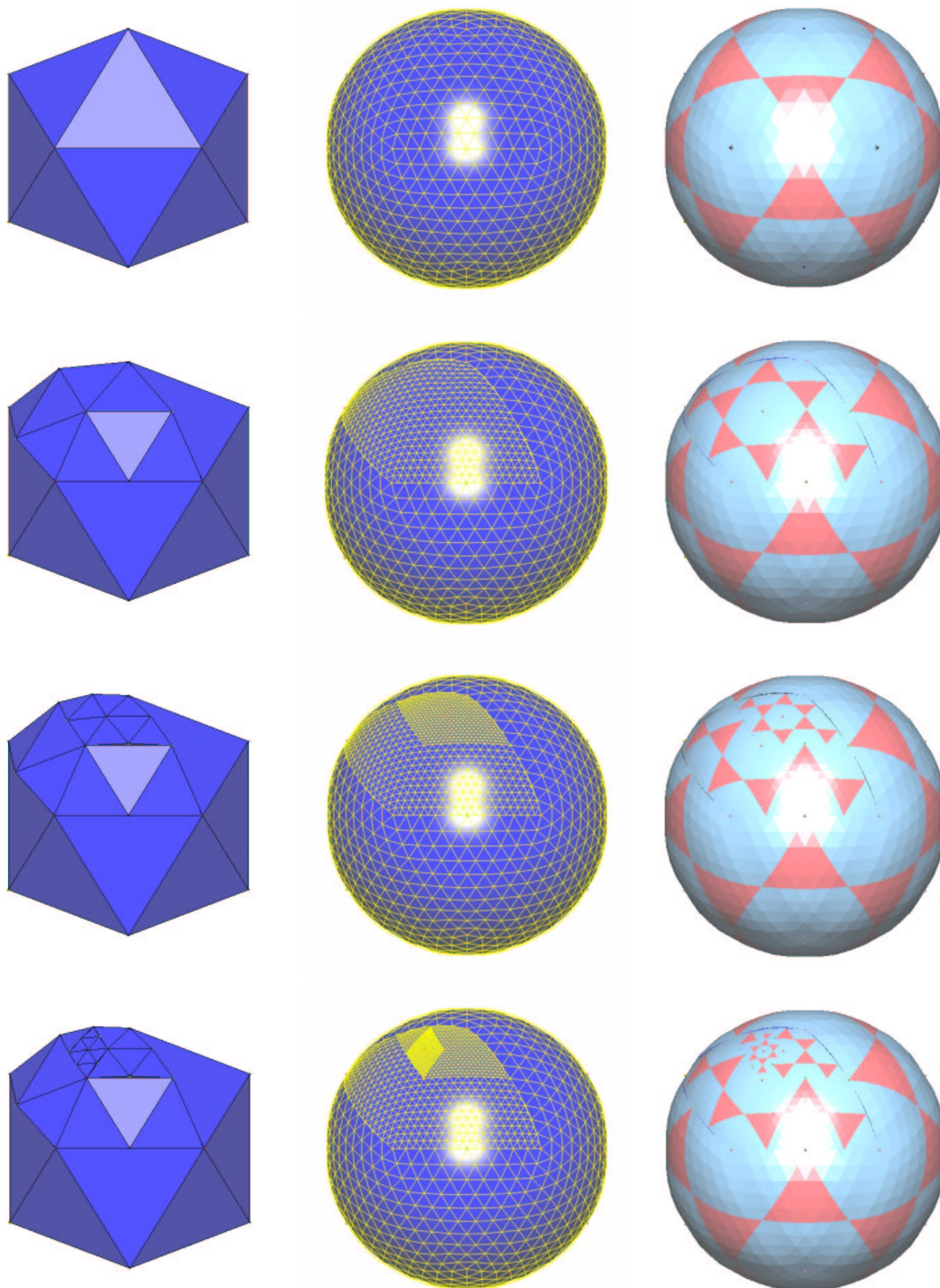


Figure 6: *Subdivision invariant local refinement of a smooth surface. From left to right are shown the input triangulation, the smooth surface together with its Bézier control mesh and the color shaded Bézier control mesh: to each input triangle correspond 4 Bézier patches, the central one is shaded in red. From top to bottom the successive refinement of the icosahedron input mesh is shown.*

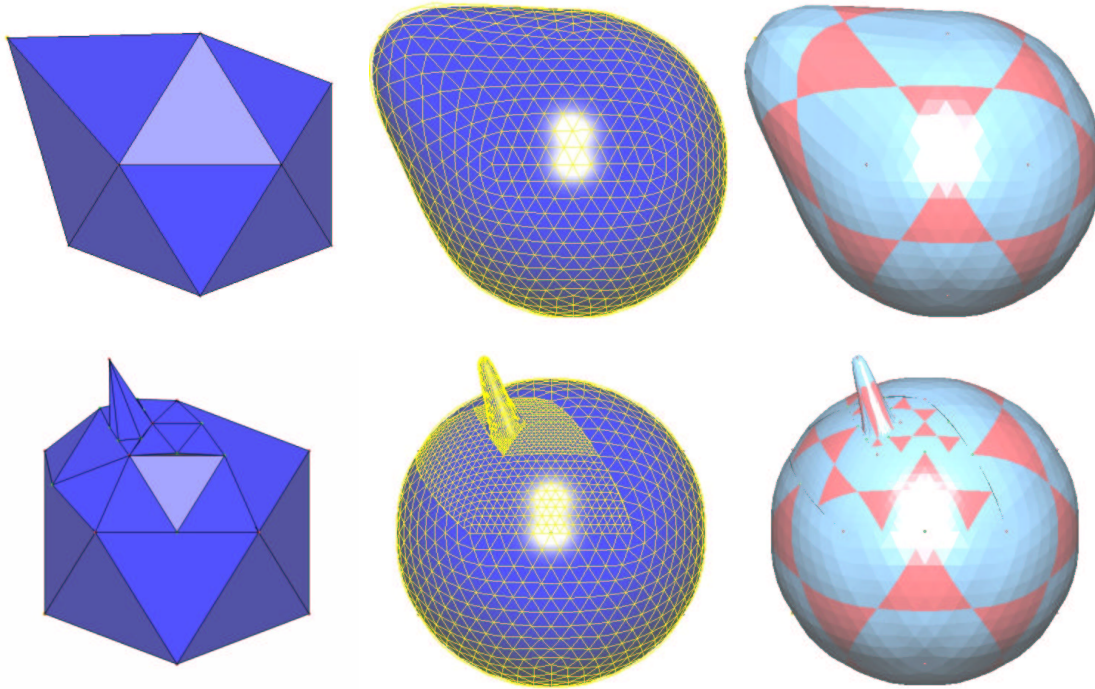


Figure 7: Deformation of input mesh at coarsest and finest level of subdivision.

REFERENCES

- [1.] Farin G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 4th edition, (1996).
- [2.] Goldman R.N., Subdivision algorithms for Bézier triangles, *CAD* **15**, (1983), 159–166.
- [3.] Hahmann S., Bonneau G-P., Triangular G^1 interpolation by 4-splitting domain triangles, *Computer Aided Geometric Design* **17** (2000), 731–757.
- [4.] Hahmann S., Bonneau G-P., Parametric surfaces over arbitrary triangulations, *IEEE Transactions on Visualization and Computer Graphics*, to appear.
- [5.] Hoschek J. and Lasser D., *Fundamentals of Computer Aided Geometric Design*, A.K. Peters, (1993).
- [6.] Jensen, Assembling triangular and rectangular patches and multivariate splines, in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM, (1987), 203–220.
- [7.] Loop C., A G^1 triangular spline surface of arbitrary topological type, *Computer Aided Geometric Design* **11** (1994), 303–330.
- [8.] Piegl L., On NURBS: a Survey, *Computer Graphics and Applications* **11** (1), (1991), 55–71.
- [9.] Piper B.R., Visually smooth interpolation with triangular Bézier patches, in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM (1987), 221–233.

- [10.] Schröder P., Zorin D., Subdivision for Modeling and Animation, SIGGRAPH course, (1998, 1999).
- [11.] Shirman L.A., Séquin C.H., Local surface interpolation with Bézier patches, Computer Aided Geometric Design **4** (1987), 279–295.