determined by minimizing the following energy function E:

$$E = \sum_{i=1}^{m} \Big(k_l (l_i^{opt} - l_i)^2 + k_a (\theta_i^{opt} - \theta_i)^2 \Big), \tag{13}$$

where m is the number of polygon legs which is equal to the number of joints, and k_l and k_a are constants called *spring coefficients*. l_i is the length of the i-th leg and θ_i is the joint angle between the $(i-1)$-st and i-th legs. l_i^{opt} and θ_i^{opt} are the optimal lengths and joint angles of the corresponding legs. In the optimal case the start and the end points of the fine-tuned curve coincide. Therefore we assume l_i^{opt} and θ_i^{opt} are the lengths and joint angles of the fine-tuned curve without the end-point constraint. Because of the invariance property of the endpoint tangent directions, the joint angle between the first and the last legs can be obtained simply by calculating the angle between them. The position of at least one point and rotation about at least one point of the curve should be fixed to avoid rigid motion during the optimization.

The conjugate gradient method (see, for instance, [13]) is used for the minimization process. The initial values of the minimization process are the control points of the invariant curve fine tuned by an identity scalar function whose degree is 1 and whose segmentation equals that of the given scalar function. To achieve geometrically similar effects under uniform scaling, the original curve is pre-scaled to fit in a unit square. After the fine tuning process, the curve will be rescaled to the original size. The final shape also depends on the ratio of the spring coefficients k_l and k_a. Large k_a relative to k_l implies a stronger resistance to change the joint angles and, therefore, a bigger tendency for the fine-tuned curve to preserve the local maxima of the curvature. The ratio used in the examples in Figures 14(c) and 14(e) is $k_l : k_a = 1 : 2$. The white marks on the curves in these figures indicate the fixed points. The processing time for the deformation is less than 0.1 second. The deformation performed by elongating the bottom leg of the original curve is shown in Figure 14(f) as a comparison to our method. Because of the limited degree of freedom of the curve, it has a small curvature around the top.

4.3 Subdivision surface

The three issues (domain, product and closedness) that have to be addressed when dealing with closed subdivision curves and surfaces are more complicated in the surface case. Fortunately, they can still be resolved using the same strategy as the curve case.

The key idea in applying the fine tuning technique to subdivision surfaces is that segmentation of the parameter space of the scalar function α must have the same knot intervals as the original subdivision surface, so that one only need to assign control values to the edges or vertices of the control mesh.[4] This requirement puts some restriction on the choice of the scalar function, but would still leave one with enough room to yield various deformations, as will be seen in this section. It should be pointed out that, by assigning control values to the edges of a subdivided control mesh instead of the original control mesh, one can overcome this restriction to have a finer segmentation for the scalar function.

The Doo-Sabin and Catmull-Clark surfaces are two popular subdivision surfaces used in graphics community. They are constructed by generalizing the idea of obtaining uniform biquadratic and uniform bicubic B-spline patches from a rectangular mesh, respectively. When using a Doo-Sabin surface as the original surface and applying a linear scalar function to it for the fine tuning process, the ideal situation would be for the fine-tuned surface to become a Catmull-Clark surface. This is a desirable combination of the degrees of the original surface and the fine-tuned surfaces as far as continuity is concerned, as mentioned in Section 2. As the degree elevation process requires the use of multiple knots, the uniform Catmull-Clark formulation can not represent a Doo-Sabin surface even if it is defined by a regular control mesh. Fortunately, non-uniform versions of these surfaces have been presented by Sederberg et al. [16] and a Doo-Sabin surface can be converted to a non-uniform Catmull-Clark surface precisely for a regular control mesh and approximately for a control mesh with arbitrary topology, as explained in Appendix (b).

Figure 16(b) shows a typical Doo-Sabin surface defined by the control mesh shown in 16(a). The control mesh is converted to the control mesh of a non-uniform Catmull-Clark surface in 16(c) where 1 is assigned as the knot interval to each edge in yellow color and 0 to each edge in red.

[4] Whether the control values are assigned to the edges or vertices depends on the degree of the scalar function. If the scalar function is linear or cubic, the control values are assigned to the edges. If the scalar function is quadratic, the control values are assigned to the vertices. This is because each segment of the scalar function corresponds to an edge or vertex of the control mesh.

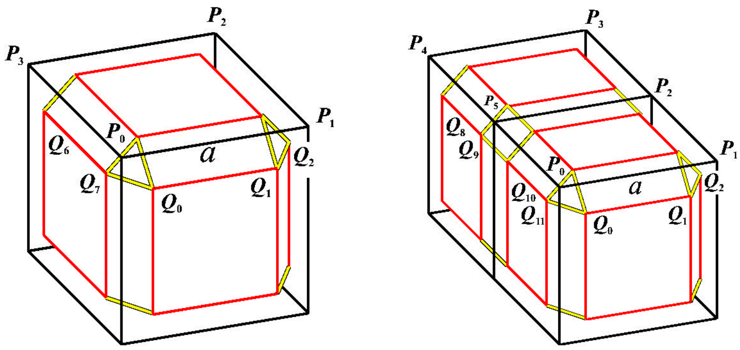
4.4 Closed surface

A physical model similar to the curve case is used to keep the fine-tuned surface closed. As a preliminary process, each edge of the control mesh is assigned a control value a_i as a specified value of the scalar function α. Each edge is again assumed to be made of a flexible spring and two adjacent edges connected to the same vertex are jointed with a rotational spring. An energy function E similar to Eq.(13) is defined and minimized to determine the positions of the control points of the fine-tuned surface:

$$E = k_l \sum_{i=1}^{m_0}(l_i^{opt} - l_i)^2 + k_a \sum_{i=1}^{m_1}(\theta_i^{a,opt} - \theta_i^{a})^2 + k_b \sum_{i=1}^{m_2}(\theta_i^{b,opt} - \theta_i^{b})^2, \tag{14}$$

where m_0 is the number of edges of the control mesh, m_1 is the number of constrained pairs of edges connected to the same vertices, as will be explained later in this subsection, and m_2 is the number of total pairs of adjacent edges connected to the same vertices, i.e. the total number of the valences (degrees) of all the vertices of the control mesh. l_i^{opt} and $\theta_i^{a,opt}$ are defined as in Section 4.2. The first and second summations are responsible for the fine tuning process and the last term is for preserving original shape. The calculation of the first term is straightforward if the optimal lengths l_i^{opt} of the edges are available. The second term is calculated as follows. Consider, for example, the boundary edges of the face $\boldsymbol{P}_0\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3$ of the Doo-Sabin control mesh in Figure 15(a). These edges are transformed to an edge loop $\boldsymbol{Q}_0\boldsymbol{Q}_1\cdots\boldsymbol{Q}_7$ in the converted Catmull-Clark control mesh. By using the assigned control values of the edges (value a is assigned to edge $\boldsymbol{P}_0\boldsymbol{P}_1$ in both (a) and (b)) we can determined optimal lengths l_i^{opt} of the edges and optimal angles $\theta_i^{a,opt}$ of the edge joints in the loop. Hence we can obtain the optimal length of, for instance, edge $\boldsymbol{Q}_0\boldsymbol{Q}_1$ and the optimal joint angle between edges $\boldsymbol{Q}_7\boldsymbol{Q}_0$ and $\boldsymbol{Q}_0\boldsymbol{Q}_1$. The situation could be more complicated and it is possible that the boundary edges of several faces correspond to the same edge loop. For example, in Figure 15(b), the boundaries of the two faces $\boldsymbol{P}_0\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_5$ and $\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4\boldsymbol{P}_5$ correspond to the same edge loop $\boldsymbol{Q}_0\boldsymbol{Q}_1\cdots\boldsymbol{Q}_{11}$. Identifying the loops which are the targets of the fine tuning is not difficult because the degree of each vertex of the converted control mesh is equal to 4 and a loop is identified by tracing the opposite edge each time we enter a new vertex. It is easy to see that each edge of the converted polygon belongs to one and only one loop. For the shape preservation term, $\theta_i^{b,opt}$ is calculated from the converted control mesh.

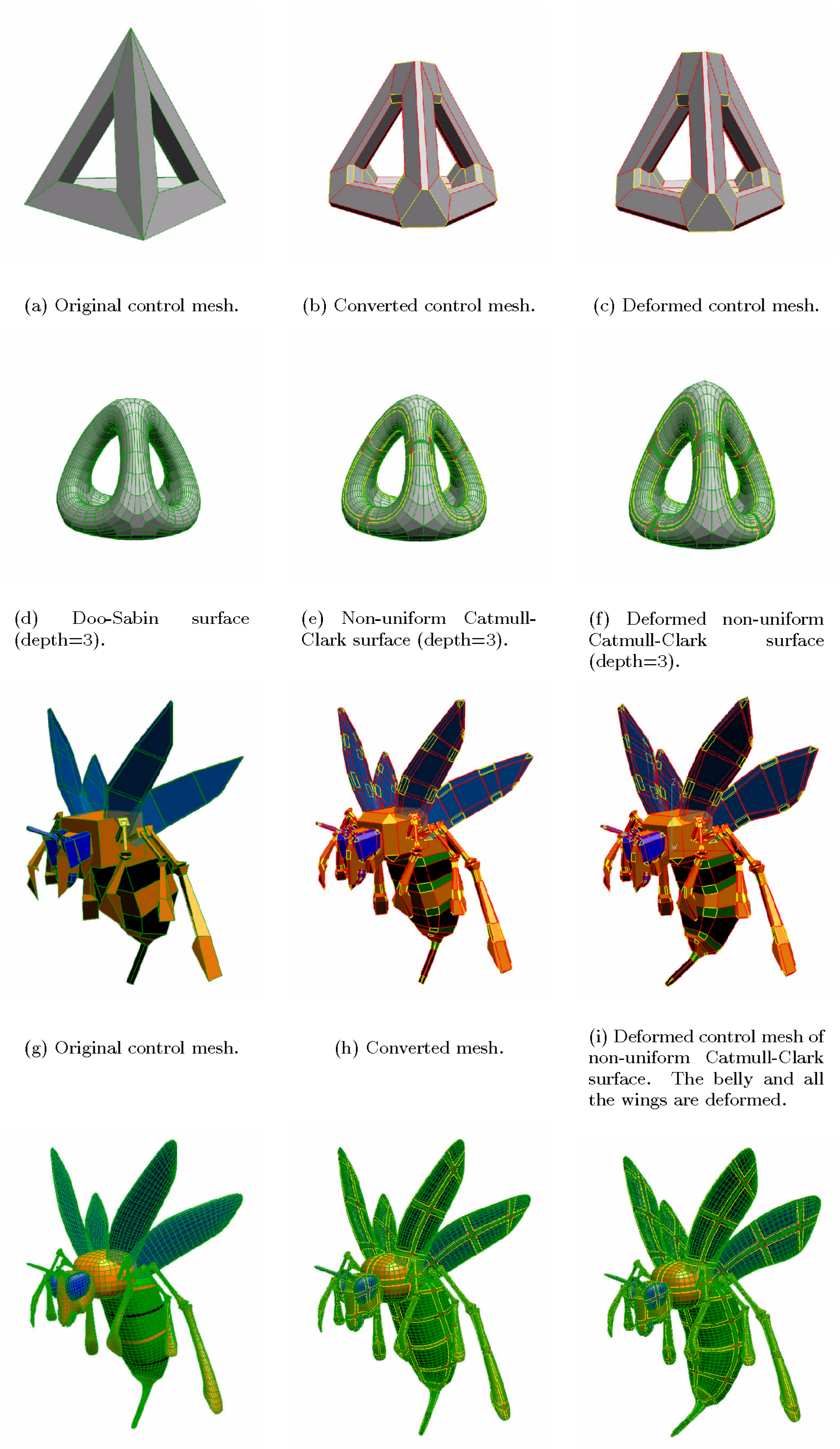
Figure 16(f) shows an example of a fine-tuned Doo-Sabin subdivision surface. Initially, each edge of the control mesh is assigned a value 1 as the default control value. Some of the edges are then given different values to deform the surface. To prevent rigid motion and to reduce the number of parameters in the optimization process, one may have some of the vertices fixed. For this example, 1.3 is assigned to each of the three edges connected to the top vertex as the control value and three vertices at the bottom are fixed. The conjugate gradient method is used for the minimization process as in the curve case. The original surface is pre-scaled to fit in a unit cube and the ratio $k_l : k_a : l_b = 1 : 1 : 1$ is used. The processing time for the deformation is less than a second. The yellow edges in 16(b) and 16(c) are given the value of 1 as their knot intervals. The red edges are given 0.05 instead of the theoretical value of 0 to clarify the effect of small knot interval value which generates dense meshes around. The example shows that one can deform a subdivision surface without even manipulating its control mesh.



(a) Edge loop around one face.

(b) Edge Loop around two faces.

Figure 15. Extraction of edge loop.



(a) Original control mesh.

(b) Converted control mesh.

(c) Deformed control mesh.

(d) Doo-Sabin surface (depth=3).

(e) Non-uniform Catmull-Clark surface (depth=3).

(f) Deformed non-uniform Catmull-Clark surface (depth=3).

(g) Original control mesh.

(h) Converted mesh.

(i) Deformed control mesh of non-uniform Catmull-Clark surface. The belly and all the wings are deformed.

(j) Doo-Sabin surface (depth=3).

(k) Non-uniform Catmull-Clark surface (depth=3).

(l) Deformed non-uniform Catmull-Clark surface (depth=3).

Figure 16. Examples of subdivision surface fine tuning.