Computer-Aided Design and Applications

http://www.cadanda.com

Surface Deformation under Area Constraints

Ying Xu[1], Ajay Joneja[2] and Kai Tang[3]

[1]Hong Kong University of Science & Technology, xyxu@ust.hk
[2]Hong Kong University of Science & Technology, joneja@ust.hk
[3]Hong Kong University of Science & Technology, mektang@ust.hk

ABSTRACT

Shape deformation is an important problem in CAD. In this paper, we look at the problem of modification of a given surface such that its surface area is changed by a user-specified amount. A differential geometric approach is adopted, and a non-linear constrained optimization formulation is proposed. A numerical technique for solving this problem is developed. The approach has been implemented as a prototype program, and some simple examples are presented.

Keywords: constrained shape deformation.
DOI: 10.3722/cadaps.2009.711-719

1. INTRODUCTION

Shape deformation has numerous applications in design and graphics. In this paper, we address the issue of modification of a surface with constraints on the surface area after the transformation. The problem was motivated by an issue arising in architectural design. Several modern buildings have complex, curved skins, especially forming skylights or canopies around the main building (see Figure 1). Typically, the architects design the shape guided strongly by aesthetic concerns. Subsequently, the shape is modified in order to meet specific engineering requirements. The cost of the canopy increases proportionally with its area. In some green buildings, the skins may be used to collect heat or light, or reduce convective losses – all of these issues are related to the surface area of the skin.

[Source: Foster and Partners]
Fig. 1: (a) Outdoor canopy (Beach Road, Singapore) (b) A Curved Skylight (Patent office building, Washington DC).

An informal problem description for such applications is: given a 3D surface, modify it in some user-specified region(s) such that the new shape is similar to the input, but has a given, specified area. Figure 2 shows examples of typical input surfaces to our problem (Figure 2 is based on two real architectural designs, courtesy of GT Asia Ltd.). In the next section, we discuss related past work and approaches on shape deformation, and also give a formal definition of the problem we shall solve.

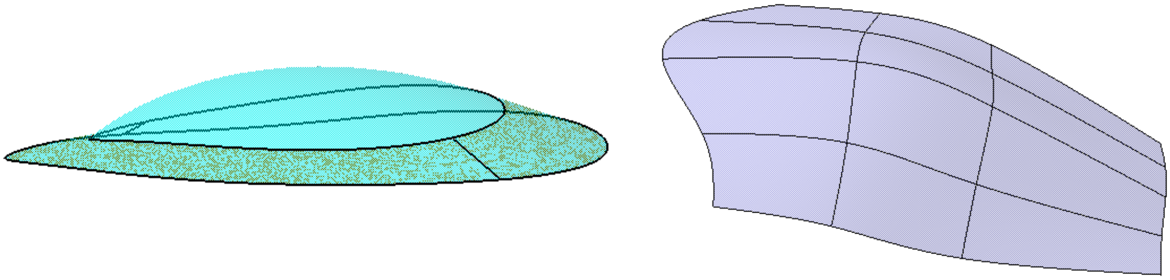

Fig. 2: CAD surface models of (a) a domed skylight for a mall (b) a portion of the canopy of a hotel building.

2. BACKGROUND

2.1 Past Work

Shape deformation and manipulation is useful in many engineering applications, including design of curved objects, manipulation of graphics objects etc. Technically, the simplest way to modify the shape of curves or surfaces is by explicitly moving one or a set of control points or curve points [17]. In such cases, the implicit constraint is for the deformed shape to maintain specified level of continuity at all knot points. However, in many applications, especially where the control mesh has a very large number of points, there is need for a higher level of abstraction. Various techniques for curve and surface manipulation including user-defined constraints on points, tangents and normal directions have been developed [10, 4, 13]. For higher level geometric control, several extensions of the Free-Form Deformation [19] technique have been used with success; Sederberg's original technique encapsulated the solid within a simple lattice of points, one or more of which could be translated in space. By associating the lattice points with the surface control points, a function that mapped each lattice motion to unique surface deformation was achieved. The extensions of this approach, including Extended FFD, Rational FFD, Axial Deformation, Space Deformation and direct FFD [13] have been well summarized in different reviews including [2, 6]. Some recent examples of Free Form Deformation of NURBS surfaces for engineering applications with point and tangent constraints include [14, 3, 16].

Several researchers have reported methods for surface deformations with volume-constraints. Sheffer [18] fixed the volume of each tensor product solid which together composes an object model in the model deformation. The method mainly works for solid primitives assembling process, but is not an independent deformation tool. In [1], a volume-preserving method based on space deformation model called DOGME is presented. Hirota *et al.* [12] preserved the total volume of a solid under free-form deformation using discrete level-of-detail representations. They start from triangle volume preserving deformation and then use multi-level-of-detail representations extending the method to general objects. The Augmented Lagrangian method is applied to solve the constrained minimization problem for the deformation. There have also been numerous surface deformation methods that use various types of multi-resolution models of the surface (including subdivision models, wavelet-based approaches as well as multi-level grids). An excellent survey on past work on shape deformation in general and on constrained shape deformation using multi-resolution techniques in particular is found in [11].

Broadly, it appears that multi-resolution methods may suffer from lack of precision, but are fast; on the other hand, differential geometry based methods tend to suffer from poor computation times and potential convergence issues. This can be averted in some cases due to special properties of the problem. For example, [8] uses the property of convexity of area enclosed by a B-Spline with respect to the control points to solve for constrained curve deformation; [20] also demonstrate that a large class of 2D curve deformation problems can be modeled in a convex optimization form, allowing the use of specialized solvers such as second order cone programming to speed up the differential geometric techniques. We are not aware of differential geometric approach in area-constrained surface deformation.

2.2 Problem Statement

Given: A B-Spline surface, ***S***, (namely, knot vectors U and V and control points, ***CP***), a subset $\boldsymbol{Q} \subseteq \boldsymbol{CP}$, and a target area, A_f, find a least cost dislocation of ***Q*** to yield a new surface, ***S**** having area = A. Here we define the cost by the least squares measure. Let the new location of the point(s) Q be denoted by the set ***P***, then we can write the problem as:

$$\begin{aligned} Min \quad & \frac{1}{2}(P-Q)^T(P-Q) \\ s.t. \quad & \text{Area}(P) = A_f \end{aligned} \tag{1}$$

3. METHODOLOGY

In section 3.1, we provide some basic definitions for completeness. Following this, we then transform our constrained optimization problem to an unconstrained one, and propose a modified Newton's approach to solve it. Section 3.3 shows some initial results of our approach, and in section 3.4 we discuss some possible extensions as well as potential techniques to improve our method.

3.1 B-Spline Definitions, Surface Area and its Derivatives

A B-Spline surface is defined as a tensor product of B-Spline curve equations over two parameters, u and v. The control points, $CP = \left\{P_{0,0}, \ \ldots \ P_{0,m}, \ \ldots \ P_{n,0}, \ \ldots \ P_{n,m}\right\}$ form a rectangular grid of (n+1) × (m+1) points in 3D Euclidean space. This allows the use of common knot vectors along each parametric direction. The iso-parametric curves along u-direction have degree p, while those along the v-direction have degree q. Given two knot vectors U (r+1 knots, where r = n+p+1) and V (s+1 knots, s=m+q+1), and the set of control points CP as above,

$$U = \left\{\underbrace{0,\ldots 0}_{p+1}, u_{p+1}, \ldots u_{r-p-1}, \underbrace{1,\ldots 1}_{p+1}\right\} \text{ and } V = \left\{\underbrace{0,\ldots 0}_{q+1}, v_{q+1}, \ldots v_{s-q-1}, \underbrace{1,\ldots 1}_{q+1}\right\}$$

the B-Spline surface is defined as:

$W(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\ P_{i,j}^{W}$, *where* $u \in [u_{p+1}, u_{r-p}], v \in [v_{q+1}, v_{s-q}]$ *and* $W = X, Y, Z$, where $N_{i,p}(u)$ and $N_{j,q}(v)$ are corresponding B-Spline basis functions and $P_{i,j}^{W}$ is the W-coordinate of control point $P_{i,j}$ [17].

Using [9] and [15], the area, A, of a parametric surface can be computed from: $A = \iint_D \left|\mathbf{r}_u \times \mathbf{r}_v\right| dudv$, and $\mathbf{r} = \mathbf{r}(u,v) = X(u,v)\mathbf{i} + Y(u,v)\mathbf{j} + Z(u,v)\mathbf{k}$; $\mathbf{r}(u,v)$ is the position vector of surface points, X,Y,Z are Cartesian coordinates, and the u, v subscripts denote the corresponding partial derivatives. So:

$$Area(P) = \iint_D \sqrt{\left(X_uY_v - X_vY_u\right)^2 + \left(Y_uZ_v - Y_vZ_u\right)^2 + \left(Z_uX_v - Z_vX_u\right)^2}\, dudv$$

From the above, the problem formulation may be re-written as:

$$\begin{aligned} Min \quad & \frac{1}{2}(P-Q)^T(P-Q) \\ s.t. \quad & \iint_D \sqrt{\left(X_uY_v - X_vY_u\right)^2 + \left(Y_uZ_v - Y_vZ_u\right)^2 + \left(Z_uX_v - Z_vX_u\right)^2}\, dudv = A_f \end{aligned} \tag{2}$$

$$W_u = \frac{\partial W}{\partial u} = \sum_{i=0}^{n} \sum_{j=0}^{m} N'_{i,p}(u) N_{j,q}(v) P_{i,j}^{w}$$
$$W_v = \frac{\partial W}{\partial v} = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N'_{j,q}(v) P_{i,j}^{w}$$
$$W = X, Y, Z$$

In order to solve the above non-linear constrained optimization problem, we shall need a numerical approach, and therefore it will be necessary to compute the gradient, i.e. the rate of change of the area with respect to the (coordinates of the) control points. According to the definition of B-Spline surface area, we denote function *f* as:

$f = \sqrt{\left(X_u Y_v - X_v Y_u\right)^2 + \left(Y_u Z_v - Y_v Z_u\right)^2 + \left(Z_u X_v - Z_v X_u\right)^2}$. Thus, the first and second derivatives are:

$$\frac{\partial Area(P)}{\partial P} = \iint_D \frac{\partial f}{\partial P} dudv \tag{3}$$
$$\frac{\partial^2 Area(P)}{\partial P^2} = \iint_D \frac{\partial^2 f}{\partial P^2} dudv \tag{4}$$

For a specific point's coordinate $P_{i,j}^{x}$, the first derivative function is

$$\frac{\partial f}{\partial P_{i,j}^{x}} = \frac{1}{f}\begin{bmatrix} X_u & X_v \end{bmatrix}\left(\begin{bmatrix} Y_v \\ -Y_u \end{bmatrix}\begin{bmatrix} Y_v & -Y_u \end{bmatrix} + \begin{bmatrix} Z_v \\ -Z_u \end{bmatrix}\begin{bmatrix} Z_v & -Z_u \end{bmatrix}\right)\begin{bmatrix} N'_{i,p} N_{j,q} \\ N_{i,p} N'_{j,q} \end{bmatrix}$$

Here for simplifications, $N_{i,p} = N_{i,p}(u)$ $N_{i,q} = N_{i,q}(v)$ $N'_{i,p} = \dfrac{dN_{i,p}(u)}{du}$ $N'_{j,q} = \dfrac{dN_{j,q}(v)}{dv}$

For the second derivative computations, there are two situations. The first is when the two coordinates share the same dimension (e.g., they are both X):

$$\frac{\partial^2 f}{\partial P_{i,j}^{x} \partial P_{s,t}^{x}} = \frac{1}{f}\begin{bmatrix} N'_{s,p} N_{t,q} & N_{s,p} N'_{t,q} \end{bmatrix}\left(\begin{bmatrix} Y_v \\ -Y_u \end{bmatrix}\begin{bmatrix} Y_v & -Y_u \end{bmatrix} + \begin{bmatrix} Z_v \\ -Z_u \end{bmatrix}\begin{bmatrix} Z_v & -Z_u \end{bmatrix}\right)\begin{bmatrix} N'_{i,p} N_{j,q} \\ N_{i,p} N'_{j,q} \end{bmatrix} - \frac{1}{f}\frac{\partial f}{\partial P_{i,j}^{x}}\frac{\partial f}{\partial P_{s,t}^{x}} \tag{5}$$

The second case is when the differentiation is with respect to two different coordinates (e.g. x and y):

$$\frac{\partial^2 f}{\partial P_{i,j}^{x} \partial P_{s,t}^{y}} = \frac{1}{f}\begin{bmatrix} X_u & X_v \end{bmatrix}\left(\begin{aligned} &\begin{bmatrix} N_{s,p} N'_{t,q} \\ -N'_{s,p} N_{t,q} \end{bmatrix}\begin{bmatrix} N'_{i,p} N_{j,q} & N_{i,p} N'_{j,q} \end{bmatrix} \\ &+\begin{bmatrix} N_{s,p} N'_{t,q} & -N'_{s,p} N_{t,q} \end{bmatrix}\begin{bmatrix} N'_{i,p} N_{j,q} \\ N_{i,p} N'_{j,q} \end{bmatrix} \end{aligned}\right)\begin{bmatrix} Y_v \\ -Y_u \end{bmatrix} - \frac{1}{f}\frac{\partial f}{\partial P_{i,j}^{x}}\frac{\partial f}{\partial P_{s,t}^{y}} \tag{6}$$

Substituting (5) and (6) in (3), (4) respectively, and subsequently performing a numerical integration, we can compute the required derivatives.

Since the iterations of the optimization function will require repeated computation of the area, we shall also use a simplified numerical computation for the area by adopting Fubini's theorem [7] and a Gaussian quadrature method [5]. The details are omitted here.

3.2 The Penalty Method

We first transform the constrained optimization problem into an unconstrained problem, (MI), via a standard penalty formulation, as:

$$Min\ \ Dist(P,Q)+\frac{1}{2}N(Area(P)-A_f)^2,$$

where N is a positive penalty parameter, and $\frac{1}{2}(Area(P)-A_f)^2$ is the penalty function from the equality constraint. As N increases, the solution of problem (MI) approaches the exact solution of problem (2). Thus generally it is required that $N\rightarrow\infty$ to assure that $\left|Area(P)-A_f\right|\rightarrow 0$. In our examples, we experimentally determined a reasonable large value for N from some examples, and this value was used as an unchanging constant subsequently. Let

$$F(P)=Dist(P)+\frac{1}{2}N(Area(P)-A_f)^2$$

To solve (MI), the first order necessary condition should be satisfied, that is to find the stationary point of function *F(P)*: $\nabla F(P)=\dfrac{\partial F(P)}{\partial P}=0$

We adopt the Newton's method to solve the equations. For simplicity, the searching direction is given without line search:

$$\Delta P=P_{k+1}-P_k=-[\nabla^2F(P)]^{-1}\nabla F(P)$$

In the above function,

$$\nabla F(P)=\frac{\partial F(P)}{\partial P}=\frac{\partial Dist(P)}{\partial P}+N[Area(P)-A_f]\frac{\partial Area(P)}{\partial P}$$
$$=P-Q+N[Area(P)-A_f]\frac{\partial Area(P)}{\partial P}$$

$$\nabla^2F(P)=\frac{\partial^2F(P)}{\partial P^2}$$
$$=\frac{\partial^2Dist(P)}{\partial P^2}+N\left\{(\frac{\partial Area(P)}{\partial P})(\frac{\partial Area(P)}{\partial P})^T+[Area(P)-A_f]\frac{\partial^2Area(P)}{\partial P^2}\right\}$$
$$=I+N\left\{(\frac{\partial Area(P)}{\partial P})(\frac{\partial Area(P)}{\partial P})^T+[Area(P)-A_f]\frac{\partial^2Area(P)}{\partial P^2}\right\}$$

The above formulation can be directly programmed as an iterative numerical method. Unfortunately, the function F(P) is not convex with respect to any of the control point coordinates in general (the proof is omitted from this version of the paper). Therefore, in practice we find that the algorithm fails to converge for many cases, especially when the dimensionality is high (i.e. many control points are simultaneously allowed to move in arbitrary direction). This is especially true if the starting point of the iterations uses the initial control point mesh as its input. There are several ways to potentially alleviate this issue. In this paper, we have implemented only one simple method: the initialization of the iterations is changed to points that satisfy the area constraint first, which would better "push" the iterations into the area close to the solutions. The following figure shows the modified Newton's algorithm.

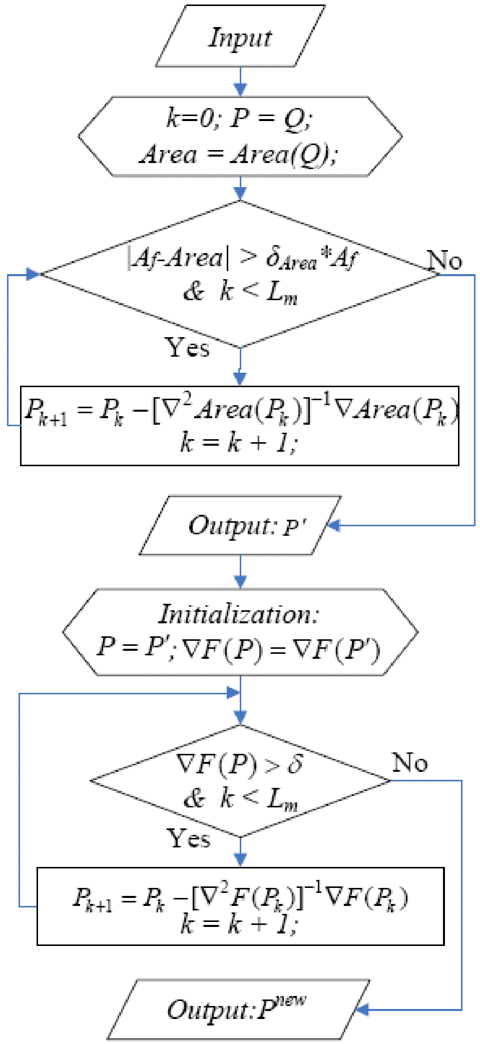

Fig. 3: The modified Newton's approach for the constrained surface modification problem.

3.3 Examples

The above algorithm is programmed using MATLAB, and several cases were tested for the hotel canopy example shown in figure 2(b). Figure 4 shows the original surface, and its control mesh. In this case, a single variable, $Z(P_{1,2})$, was allowed to move, and the area constraint required the total surface area to be increased from 5014.36m² to 5139.72m² (2.5% increase in area). Figure 5(a) and (b) show two views of the corresponding new surface superposed on the original one (the isoparametric lines show how the surface shape is changed). Figure 6(a) shows the effect of reducing the original area by 2.5% via the same variable, and Figure 6(b) shows this surface with the original one superposed over it.

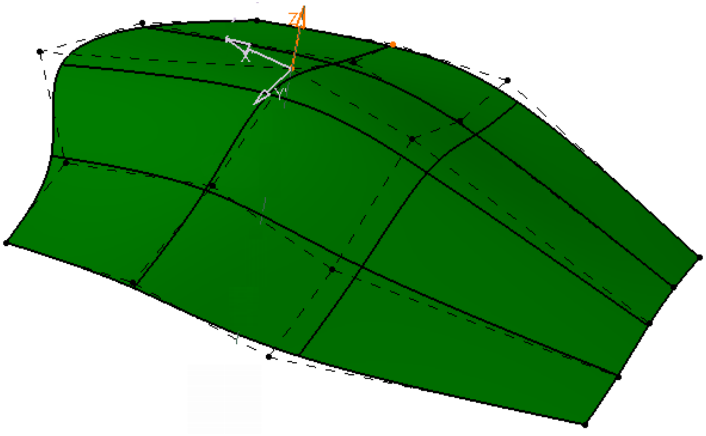

Fig. 4: Canopy surface; original area: 5014.36m²; dashed lines show control mesh.

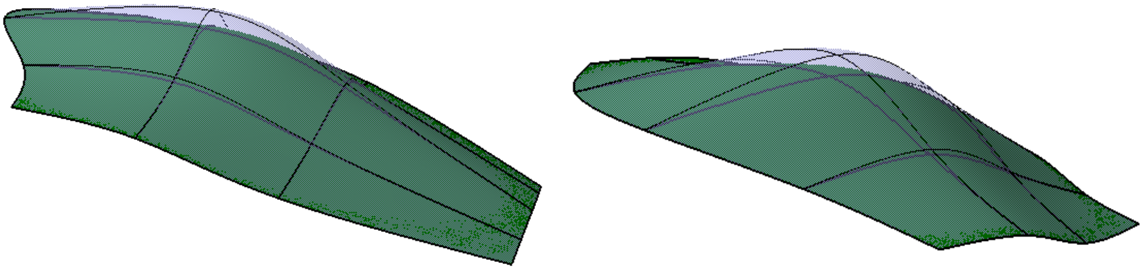

Fig. 5: (a) Modified surface (transparent blue) and the original one (green) (b) another view.

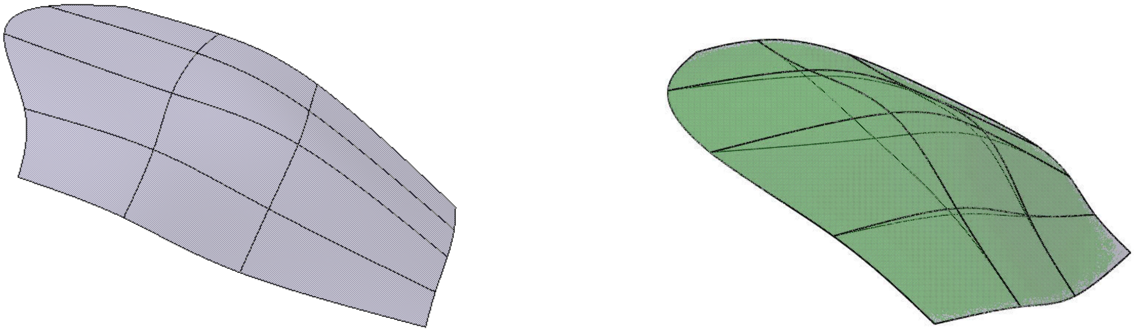

Fig. 6: (a) Same surface with area reduced by 2.5% (125m²) (b) modified and original surfaces.

In the second example, six of the control points were allowed to be moved to give a 10% increase in area (from 5014.36m² to 5515.81m²). First, we limit the program to change only the Z-coordinate of each of these six control points; Figure 7 shows two views of the resulting surface displayed superposed over the original one. Next, all coordinates of all of these six control points were allowed to be free (i.e. 18 variables), with an area increase of 5% (from 5014.37m^2 to 5265.08m^2); the results are shown in figure 8(a) and (b).

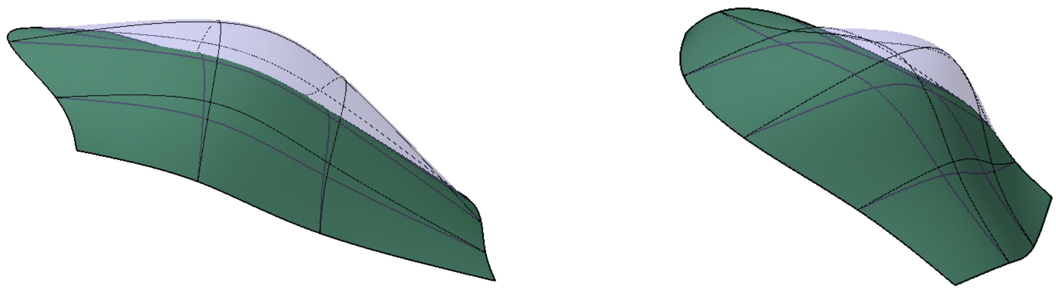

Fig. 7: (a) 10% increase in area moving Z-coords of 6CPs; (b) modified surface (blue) and original (green).

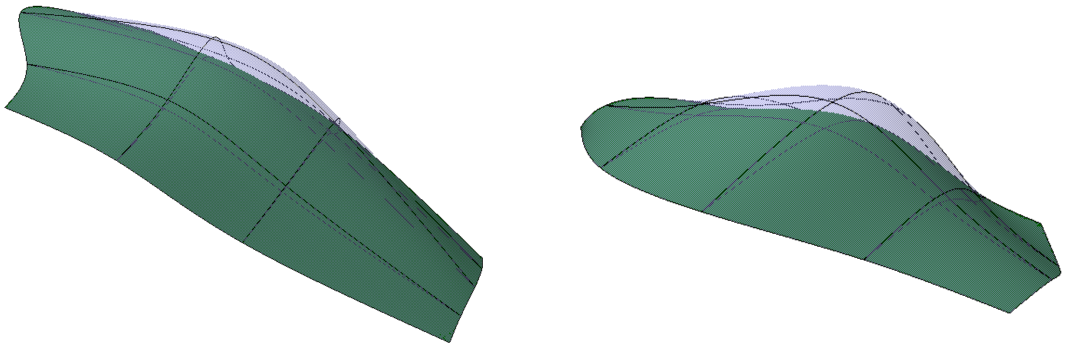

Fig. 8: (a) 5% increase in area freely moving 6CPs; (b) modified surface (blue) and original (green).

3.4 Discussion

These initial results are encouraging and indeed yielded some useful surfaces for a real world application. We are now looking at several important issues that emerge. From a theoretic view, we are yet using a relatively straightforward Newton's method for solving the non-linear optimization formulation with a moderate number of variables. Unfortunately the problem is not convex, therefore we are still exploring which particular non-linear solver may best solve a problem with this structure. Some past experience with curve shape manipulation indicates that Uzawa method may not work well for this problem. We are exploring other variations of the Newton's method, in particular, the damped Newton method to provide a more robust implementation. Another issue is that of convergence. Our current implementation uses a rather costly numerical technique to solve the double integrals for equations (3) and (4), resulting in fairly slow convergence time (ranging from two minutes for single variable to over 60 minutes for the 18 variable case). The third and very practical issue is to incorporate aesthetic considerations as constraints. In several examples where the control point was free to translate along any axis resulted in the shape of the surface changing far from the original. One potential solution is to restrict each control point to a single degree of freedom, e.g. along the normal to the control mesh at that point (this can be computed as the mean of the normal vectors of all triangular facets surrounding the control point, for example).

4. CONCLUSIONS

In this paper, we present a simple Newton's method variation to solve a constraint shape modification problem with non-linear integral constraints. The problem has practical applications in many fields of engineering apart from our motivating field, architecture design. The initial results we obtained show that the operator can be successfully developed and used as an application program linked to a typical

CAD system. The approach we have used appears promising enough for us to devote continued effort in improving the method in stability and speed.

5. ACKNOWLEDGEMENTS
This research is supported in part by the RGC CERG grant number 614407. The authors also thank Mr. Billy Wong Chi Pan and David Kwok of GT Asia for their valuable inputs and sample data.

6. REFERENCES

[1] Aubert, F.; Bechmann, D.: Volume-Preserving Space Deformation, Computer & Graphics, 21, 1997, 625–639.
[2] Bechmann, D.: Space Deformation Models Survey, Computer & Graphics, 18 (4), 1994, 571-586.
[3] Bechmann, D.; Gain, J.: Point, Curve, Surface, Volume-Based Spatial Deformation, http://dpt-info.u-strasbg.fr/~bechmann/Deformation1.pdf, 2005.
[4] Borel, P.; Rappoport, A.: Simple Constrained Deformations for Geometric Modeling and Interactive Design, ACM Transactions on Graphics, 13(2), 1994, 137-155.
[5] Chapra, C.; Canale, R. P.: Numerical Methods for Engineers, 3rd ed., McGraw-Hill, Inc., New York, 1998.
[6] Cheutet, V.; Daniel, M.; Hahmann, S.; La Greca, R.; Léon, J.-C.; Maculet, R.; Ménegaux, D.; Sauvage, B.: Constraint Modeling for Curves and Surfaces in CAGD: A Survey, HAL – CCSD, 2007.
[7] Dineen, S.: Multivariate Calculus and Geometry, Springer-Verlag Ltd., London, 2001.
[8] Elber, G.: Multi-resolution curve editing with linear constraints, 6th ACM/IEEE Symposium on Solid Modeling and Applications, Ann Arbor, Michigan, June 2001, 109-119.
[9] Folland, G. B.: Advanced Calculus, Prentice-Hall. Inc., 2002.
[10] Fowler, B.; Bartels, R.: Constraint-Based Curve Manipulation, IEEE Computer Graphics and Applications, 13(5), 1993, 43-49.
[11] Hahmann, S.; Elber, G.: Constrained Multiresolution Geometric Modeling, Advances in Multiresolution for Geometric Modeling, http://www-lmc.imag.fr/lmc-mga/Stefanie.Hahmann/PUBLICATIONS/HE04.pdf
[12] Hirota, G.; Maheshwari, R.; Lin, M. C.: Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization, Computer-Aided Design, 32, 2000, 499–512.
[13] Hsu, W. S.; Hughes, J. F.; Kautinan, H.: Direct Manipulation of Free-Form Deformations, Computer & Graphics, 26(2), 1992, 177-184.
[14] Hu, S. M.; Zhang, H.; Tai, C. L.; Sun, J-G.: Direct Manipulation of FFD: Efficient Explicit Solutions and Decomposable Multiple Point Constraints, The Visual Computer, 17, 2001, 370-379.
[15] Kreyszig, E.: Advanced Engineering Mathematics, John Wily & Sons, Inc., 1999.
[16] Orazi, L.: Constrained Free Form Deformation as a Tool for Rapid Manufacturing, Computers in Industry, 58, 2007, 12-20.
[17] Piegl, L.; Tiller, W.: The NURBS Book, Springer-Verlag, 1997.
[18] Rappoport, A.; Sheffer, A.; Bercovier, M.: Volume-Preserving Free-form Solids, ACM Symposium on Solid Modeling 95, 1995, 361-370.
[19] Sederberg, T.: Parry, S.: Free-Form Deformation of Solid Geometric Models, ACM Computer Graphics (SIGGRAPH 86 Proceedings), 20, 1986, 151-160.
[20] Xu Y.; Joneja, A.: A New Constrained Shape Deformation Operator with Applications in Footwear Design, Procs. of the IDETC, CIE, Aug 3-6 2008, New York.