

Application of morphing technique with mesh-merging in rapid hull form generation

Ju Young Kang¹ and Byung Suk Lee²

¹*Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

²*Safety at Sea Ltd, Glasgow, UK*

ABSTRACT: *Morphing is a geometric interpolation technique that is often used by the animation industry to transform one form into another seemingly seamlessly. It does this by producing a large number of ‘intermediate’ forms between the two ‘extreme’ or ‘parent’ forms. It has already been shown that morphing technique can be a powerful tool for form design and as such can be a useful addition to the armoury of product designers. Morphing procedure itself is simple and consists of straightforward linear interpolation. However, establishing the correspondence between vertices of the parent models is one of the most difficult and important tasks during a morphing process. This paper discusses the mesh-merging method employed for this process as against the already established mesh-regularising method. It has been found that the merging method minimises the need for manual manipulation, allowing automation to a large extent.*

KEY WORDS: Morphing; Merging method; Hull form generation; Parametric variation; Mesh; Remeshing.

INTRODUCTION

Unlike mass produced goods, such as cars and televisions, ships have to be tailor-made according to the requirements of the customers. Therefore, designing ships is an exacting task satisfying the given requirements within a set of constraints. Furthermore, the highly competitive world market demands merchant ships to be designed quickly but to high standards. This requires a high degree of optimisation with the ultimate objective of the best commercial performance for a given set of circumstances.

As a consequence ship design techniques have been continuously improved in recent times and, with the ever increasing computing power, various ship performance evaluation tools are now used routinely in design work (see, for example, the reviews of key research work in this area by Nowacki (2010) and Sharma, et al. (2012)). Computer simulation tools in particular have become widely available, replacing much of physical model tests. These numerical tools also allow a large number of design alternatives to be evaluated, provided, of course, the required alternatives are available in sufficient detail. In order for automatic or semi-automatic optimisation to be successful the designs have to be made available either generated *a priori* in a database or generated ‘on demand’. The databases of systematic hull forms and parametric generation techniques of hull forms are very good examples of these.

Taylor (1915) was one of the first researchers to carry out a systematic study of hull forms. He used a mathematical formulation to generate hull forms, based on parabolic and hyperbolic functions. This method yielded a series of hull forms that could be used for exploring the effect of certain hull form parameters on ship resistance. The hull forms that could be defined using these mathematical functions, however, were too simple to be anything but very rough approximations of real ship hull

forms. Kuiper (1970) introduced a new approach to hull form design, *viz*, parametric hull form generation. He attempted to use initial parameters describing hull characteristics rather than using hull form offset data, to generate waterline, profile, and section area curves. Although this method was too simple to represent complex hull forms in sufficient details, the parametric approach to hull form generation has influenced research work that followed (Reed and Nowacki, 1974; Harries and Abt, 1998; Kim, 2004; Zhang, 2008 to name but a few).

With more widespread use of digital computer and the increasing availability of computing technology, many mathematical methods were developed to define the surface of the hull forms more accurately. In 1968 Bézier (1968) developed a free form curve generation technique for designing automobile bodies while working as an engineer at the car manufacturer Renault. The ‘Bézier curve’ is a special case of B-spline curve which is far more flexible to work with. A numerical algorithm for B-spline was developed by de Boor (1971) and Cox (1972), and its application to ship form definition was explored by Riesenfeld (1972) and Gordon and Riesenfeld (1974). Many researchers have found such curve generation techniques very useful in representing and generating hull form shapes precisely. Creutz and Schubert (1978) developed a B-spline method using hull form parameters, and Munchmeyer, Schubert and Nowachi (1979) extended the form parameter method to allow knuckle lines and bulbous bow to be created. In addition to these studies, many similar parametric methods based on B-spline have been studied and developed by many researchers. However, most of the work has focused on accurate representation rather than rapidity or efficiency of generation. Keane (1988) offered a method allowing rapid generation of hull forms using conformal mapping technique; however, the method has been found to have a weakness in generating certain types of hull forms such as frigate type forms.

One of the earliest approaches to hull form generation is to take an existing hull form and modify it to meet the requirements, and this method is still widely used in both practical design and research work. The classic example of this is the direct manual manipulation of the surface definition data. Whilst being a flexible and effective tool in the hands of experienced designers, it is nevertheless difficult to obtain a good hull form. Lackenby (1950) was one of the first researchers who introduced the idea of systematic transformation. Alef and Collatz (1976), Versluis (1977), and Rabien (1979) developed Lackenby’s method further, and investigated hull form variation from a parent hull without manual manipulations. These methods are still used in modern hull design applications, but there is a limit in achieving variation of steep curvature parts such as bow and stern.

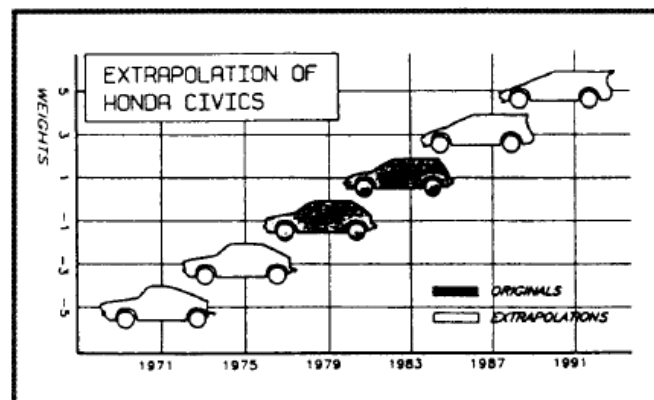


Fig. 1 Example of morphing for product design (Chen and Parent, 1989).

Morphing, meanwhile, is a geometric interpolation technique that produces intermediate forms between typically two extreme forms. Originally the morphing technology was invented as a method of working out the transformation process (interpolation) or to expect the future results (extrapolation) from satellite images of the earth’s surface in the early 1960s. More recently, morphing technique has become a significant tool of computer-graphics animators by providing an effective means of producing a seamless sequence of forms. It is used to display a smooth transition between two models by inserting a sufficient number of interpolated models between them.

There are many strategies of optimisation, but a common feature of design optimisation of any product or system involves generating a feasible design followed by the performance evaluation of the design. The morphing technique would be undoubtedly useful in this process by offering a relatively painless method of generating a vast number of design alternatives rapidly.

In form design this capability can be used to generate a large number of predictable models from a set of given 'parent' forms. Although the morphing technique holds significant potential for rapid generation of shapes, little attention has been given to the application of morphing for product design so far.

Nevertheless, similar ideas have been used in deriving interpolated or extrapolated forms in 2 D. For example, Chen and Parent (1989) developed a transformation algorithm for 3 D lofted models using the integration of 2 D planar contours. Thus, only two axes of the 2 D planar contours need to be transformed since each contour of the first parent model corresponds to a contour of the second parent model. Fig. 1 shows an example of generating future/retrospective models (1971, 1975, 1987, and 1991) from existing two Honda Civic models (1979 and 1983) through extrapolation.

One of the first attempts in the application of morphing to substantial 3 D models of a product was presented by Hsiao and Liu (2002). Kang and Lee (2009, 2010) applied mesh-based morphing to rapid generation of ship hull forms. The key step in the morphing process is the harmonisation of the mesh structures representing the parent 3 D surfaces, and they investigated the use of 'regularising method' for this purpose. The essence of this method is to build a mesh structure entirely different from the original mesh structures, whilst maintaining the form characteristics of the models. The method was found to be fast and reliable as a re-meshing algorithm, but it required extensive manual manipulations.

'Merging method' for establishing a correspondence between two mesh structures, on the other hand, can save much time by reducing the need for manual intervention, and it allows the original mesh shape to be maintained. One of the first researchers to discuss this method was Kent, and Parent and Carlson (1991) and Kent, Carlson and Parent (1992), and this method has been mainly used for producing animation (Gregory, et al., 1998; Alexa, 2000).

MORPHING OF 3 D SURFACES USING MERGING METHOD

Fig. 2 shows an overview of the algorithm used for hull form morphing based on merging method. The first task is mapping the original meshes of the 3 D hull forms onto a 2 D parameter plane in order to make it easier to handle the meshes. Then, the projected meshes are merged into a single mesh. When this 2 D merged mesh is remapped back onto each original 3 D surface, the original meshes which were originally of different structures, have now been made to correspond to each other so that they have the same number of faces, vertices, and edges. With the meshes thus harmonised, interpolation between the vertices of the parent models can be carried out. These three steps (parameterisation, merging and remapping) are examined in more detail in this section.

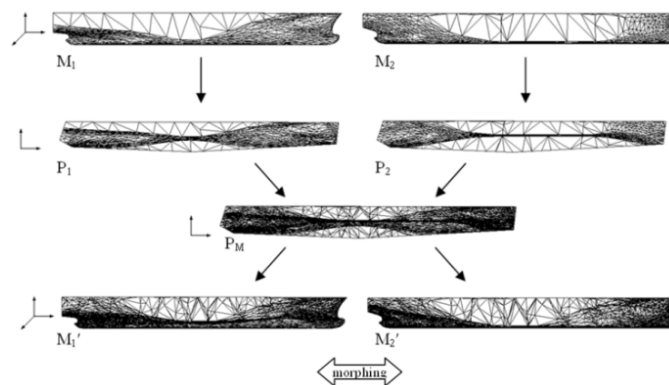


Fig. 2 Outline procedure for the hull form morphing based on merging method.

Parameterisation

As indicated earlier, one of the most important and difficult tasks in morphing is finding a proper correspondence between the vertices of parent models. This can be achieved by redefining the parent forms in an isomorphic mesh structure, either by recreating the mesh or by merging the existing meshes. This paper is concerned with the latter method only.

The difficulty, however, lies in the fact that it will be far too complicated, if not impossible, to carry out the merging operation in 3 D. Therefore, it is necessary to map the 3 D surfaces onto a 2 D plane first.

In this study, Floater’s shape-preserve parameterisation method (1997) has been used for this process. A local surface S_i (consisting of internal point s_i and its neighbours s_{jk}) is mapped to an identical 1-ring structured parameter space P_i (consisting of internal points p_i and its neighbours p_{jk}) so that the edge length in each radial direction is exactly preserved, and the angles between two consecutive edges are also preserved as shown in Eqs. (1) and (2), and Fig. 3. Here s_i and p_i denote the i^{th} internal point of a local surface and a parameter respectively, and s_{jk} and p_{jk} represents their k^{th} neighbours respectively. $k = 1, \dots, c_i$, where c_i signifies the number of the neighbours.

$$\|p_{jk} - p_i\| = \|s_{jk} - s_i\| \tag{1}$$

$$ang(p_{jk}, p_i, p_{jk+1}) = 2\pi \cdot \frac{ang(s_{jk}, s_i, s_{jk+1})}{\theta_i} \tag{2}$$

where $ang(a,b,c)$ indicates the angle between the edges ab and bc . θ_i designates the total sum of the angles between two consecutive edges of a local surface S_i .

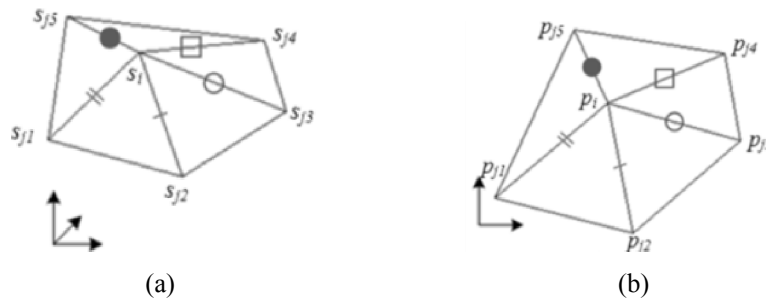


Fig. 3 An example of local parameterisation.

Having mapped a local surface S_i to a parameter space P_i , convex combination coefficients $\lambda_{i,k}$ of P_i can be calculated from an internal point P_i and its neighbours p_{jk} (See Floater, 1997). When the convex combination coefficients are uniform, the procedure is also known as Tutte’s barycentric method (Tutte, 1959). Here, $\lambda_{i,k}$ represents a convex combination coefficient of the i^{th} internal point to its k^{th} neighbour. The convex combination coefficients are non-negative and sum up to unity as follows:

$$\sum_{k=1}^{c_i} \lambda_{i,k} = 1 \quad \text{and} \quad \lambda_{i,k} > 0 \tag{3}$$

If there are three neighbours of an internal point, the convex combination coefficients $\lambda_{i,k}$ can be simply found as:

$$\begin{aligned} \lambda_{i,1} &= \frac{area(p_i, p_{j2}, p_{j3})}{area(p_{j1}, p_{j2}, p_{j3})} \\ \lambda_{i,2} &= \frac{area(p_{j1}, p_i, p_{j3})}{area(p_{j1}, p_{j2}, p_{j3})} \\ \lambda_{i,3} &= \frac{area(p_{j1}, p_{j2}, p_i)}{area(p_{j1}, p_{j2}, p_{j3})} \end{aligned} \tag{4}$$

where $area(a, b, c)$ means the area of the triangle abc . The idea is illustrated in Fig. 4.

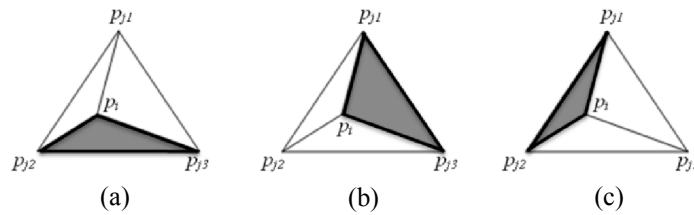


Fig. 4 An example of calculating convex combination coefficients $\lambda_{i,k}$ ($c_i = 3$).

With the convex combination coefficients $\lambda_{i,k}$ of a parameter space P_i thus obtained, an internal point p_i can be represented by its neighbours p_{jk} and their convex combination coefficients $\lambda_{i,k}$ as follows:

$$p_i = \lambda_{i,1}P_{j1} + \lambda_{i,2}P_{j2} + \lambda_{i,3}P_{j3} \tag{5}$$

which can be rewritten as:

$$P_i = \sum_{k=1}^{c_i} \lambda_{i,k} P_{jk} \tag{6}$$

If there are more than three neighbours ($c_i > 3$) of an internal point, the procedure to calculate the convex combination coefficients becomes more complex, and requires iterative operations of Eq. (4) as described below.

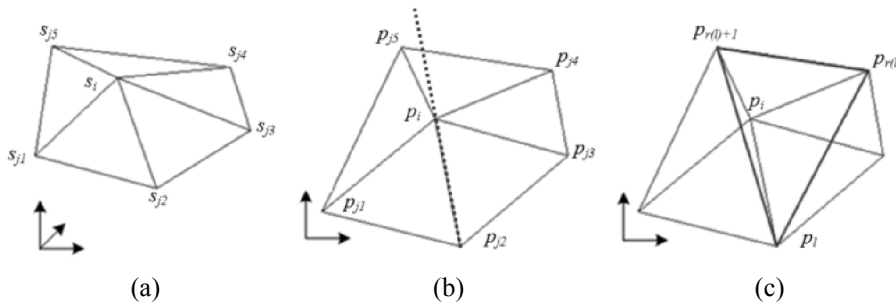


Fig. 5 An example of calculating convex combination coefficients $\lambda_{i,k}$ ($c_i > 3$).

For each neighbouring node $p_l, l \in \{1, \dots, c_i\}$, the straight line through p_l (p_{j2} in Fig. 5(b)) and p_i intersects a line segment with endpoints $p_{r(l)}$ and $p_{r(l)+1}$ (p_{j4} and p_{j5}) as shown in Fig. 5. The triangle formed by $p_l, p_{r(l)}$ and $p_{r(l)+1}$ ($p_{j2}p_{j4}p_{j5}$ in the current example) is then taken as the selected triangle for the neighbour p_l (p_{j2}). The weighting factors applied to these points δ_1, δ_2 and δ_3 are such that $\delta_1 > 0, \delta_2 > 0, \delta_3 > 0$ and $\delta_1 + \delta_2 + \delta_3 = 1$. The internal point p_i can be represented in a similar manner as Eq. (5) by the following expression:

$$p_i = \delta_1 p_l + \delta_2 p_{r(l)} + \delta_3 p_{r(l)+1}, \quad l \in \{1, \dots, c_i\} \tag{7}$$

or

$$p_i = \delta_1 p_l + \delta_2 p_{r(l)} + (1 - (\delta_1 + \delta_2)) p_{r(l)+1}, \quad l \in \{1, \dots, c_i\} \tag{8}$$

The weighing factors δ_1, δ_2 and δ_3 can be found by applying Eq. (4); they can also be calculated from Eq. (8), since there are two unknown variables (δ_1, δ_2) and two equations due to the two components $U = \{u, v\}$ in 2 D. U denotes parameterised nodes, and u and v indicate x-axis and y-axis components of U respectively. In order to apply the equations to all the neighbours, a more general notation is adopted by substituting $\mu_{l,l} = \delta_1, \mu_{r(l),l} = \delta_2, \mu_{r(l)+1,l} = \delta_3$, and $\mu_{k,l} = 0$ for all other k . In case that the

neighbour p_{j2} is selected for p_i as shown in Fig. 5(b) and (c), $\mu_{k,l}$ and p_i would be:

$$\begin{aligned} \mu_{1,2} &= 0, \mu_{2,2} = \delta_1, \mu_{3,2} = 0, \mu_{4,2} = \delta_2, \mu_{5,2} = \delta_3 \\ p_i &= \mu_{1,2}p_{j1} + \mu_{2,2}p_{j2} + \mu_{3,2}p_{j3} + \mu_{4,2}p_{j4} + \mu_{5,2}p_{j5} = \mu_{2,2}p_{j2} + \mu_{4,2}p_{j4} + \mu_{5,2}p_{j5} \end{aligned} \tag{9}$$

For each l ,

$$\sum_{k=1}^{c_i} \mu_{k,l} = 1, \mu_{k,j} \geq 0 \tag{10}$$

$$p_i = \sum_{k=1}^{c_i} \mu_{k,l} p_{jk} \tag{11}$$

Finally, the convex combination coefficients of the i^{th} internal point p_i and its neighbours p_{jk} are:

$$\lambda_{i,k} = \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l}, k \in \{1, \dots, c_i\} \tag{12}$$

From Eqs. (10) and (12) the following expression can be derived:

$$\sum_{k=1}^{c_i} \lambda_{i,k} = \sum_{k=1}^{c_i} \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l} = \frac{1}{c_i} \sum_{l=1}^{c_i} \sum_{k=1}^{c_i} \mu_{k,l} = \frac{1}{c_i} \sum_{l=1}^{c_i} 1 = 1 \tag{13}$$

In a similar manner to the way the convex combination coefficients are obtained as shown in Eqs. (12), the following equation can be derived from Eqs. (11) and (12) for the i^{th} internal point p_i :

$$p_i = \frac{1}{c_i} \sum_{l=1}^{c_i} p_i = \frac{1}{c_i} \sum_{l=1}^{c_i} \sum_{k=1}^{c_i} \mu_{k,l} p_{jk} = \sum_{k=1}^{c_i} \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l} p_{jk} = \sum_{k=1}^{c_i} \lambda_{i,k} p_{jk} \tag{14}$$

Note that Eqs. (13) and (14) are identical to Eqs. (3) and (6) respectively.

With the convex combination coefficients thus obtained, it is finally possible to parameterise the whole 3 D surface to a 2 D plane. The parameterisation starts with mapping the 3 D boundary nodes to the 2 D boundary polygon, and then the internal nodes can be mapped to the inside of the boundary polygon.

For the 2 D boundary polygon, any shape, size, or number of edges can be chosen; however, it is helpful to have the polygon roughly resembling the outline of the original 3 D model so that the shape of the parametric mesh can be checked easily. In addition, it is important for the corresponding features of the parent models to be assigned to same vertices in the parametric domain so that contextually proper correspondence can be established. For example, the nose tips of bulbous bow of the parent models should be assigned to an identical vertex of the 2 D domain. Fig. 6 shows an example of setting a 2 D domain for mapping two different 3 D hull forms.

After setting feature points onto the 2 D polygon, each 3 D boundary segment between two neighbouring feature points is mapped to the corresponding edge of the polygon. The boundary nodes on the curve A1B1 of the first parent model and on the curve A2B2 of the second are mapped onto the identical straight line between A' and B' of the parameter polygon. The boundary nodes of each original model are placed on A'B' maintaining the length ratios. In the similar manner the nodes lying on the other boundaries are mapped onto the corresponding edges of the 2 D polygon.

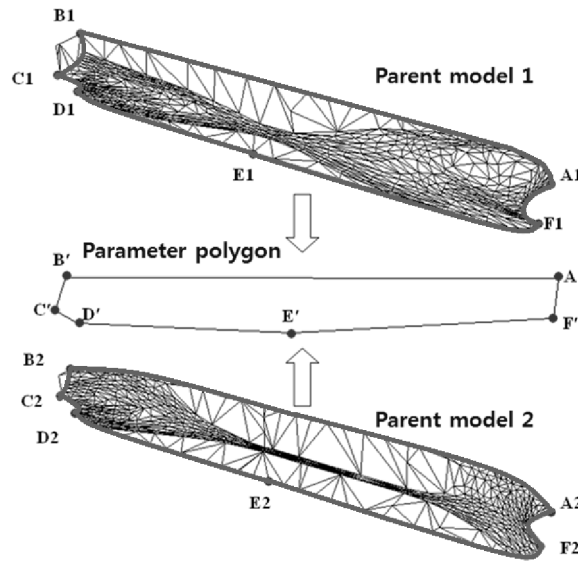


Fig. 6 An example of setting a 2 D parameter polygon and selecting feature points on 3 D hull form meshes.

With the boundary nodes mapped on the boundary of 2 D polygon and the convex combination coefficients calculated from Eq. (12), internal points can now be parameterised. Let the total number of nodes and internal nodes be N and n respectively, then the internal nodes of the mapped 2 D plane would be U_1, \dots, U_n and the boundary nodes would be U_{n+1}, \dots, U_N . For the i^{th} internal node, $i \in \{1, \dots, n\}$, the convex combination coefficients $\lambda_{i,k}$ satisfy the following:

$$\begin{cases} \lambda_{i,k} > 0 & \text{if } (i,k) \in E \\ \lambda_{i,k} = 0 & \text{if } (i,k) \notin E \end{cases}, \sum_{k=1}^N \lambda_{i,k} = 1 \tag{15}$$

where a node U_k is a neighbour of node U_i if $(i, k) \in E$. Here, E is the edge set of the parent model mesh. In a similar way to Eqs. (6) and (14), each internal node U_i can be represented by its neighbouring nodes U_k and convex combination coefficients $\lambda_{i,k}$ as :

$$U_i = \sum_{k=1}^N \lambda_{i,k} U_k \tag{16}$$

$$U_i = \sum_{k=1}^n \lambda_{i,k} U_k + \sum_{k=n+1}^N \lambda_{i,k} U_k, \quad i = 1, \dots, n \tag{17}$$

The first term of the right hand side of Eq. (17) represents the parameterised internal nodes, which are unknown, and the second term signifies parameterised boundary nodes, which are known. By grouping the unknown terms on the left hand side and the known terms on the right, Eq. (17) can be rewritten in the following form:

$$U_i - \sum_{k=1}^n \lambda_{i,k} U_k = \sum_{k=n+1}^N \lambda_{i,k} U_k, \quad i = 1, \dots, n \tag{18}$$

In 2 D, U_i has two components u_i and v_i , and therefore it can be expressed in the form of two matrix equations:

$$\mathbf{A}\mathbf{u} = \mathbf{b}_1 \tag{19-a}$$

$$\mathbf{A}\mathbf{v} = \mathbf{b}_2 \tag{19-b}$$

Here \mathbf{u} and \mathbf{v} are the column vectors of (u_1, \dots, u_n) and (v_1, \dots, v_n) respectively, and \mathbf{A} is an $n \times n$ matrix whose elements are:

$$\begin{pmatrix} a_{i,k} = 1 & \text{if } i = k \\ a_{i,k} = -\lambda_{i,k} & \text{if } i \neq k \end{pmatrix}, \quad i = 1, \dots, n, \quad k = 1, \dots, n \quad (20)$$

b_1 and b_2 are the multiplication of an $n \times (N - n)$ matrix by the column vectors of (u_{n+1}, \dots, u_N) and (v_{n+1}, \dots, v_N) respectively. The elements of the $n \times (N - n)$ matrix are equivalent to $\lambda_{i,k}$ for $i = 1, \dots, n$ and $k = n + 1, \dots, N$. Let us take an example case with three internal nodes ($n = 3$) and four boundary nodes ($N = 7$), the two matrix equations would be:

$$\begin{matrix} & \text{(known)} & \text{(unknown)} & & \text{(known)} & \text{(known)} \\ \begin{bmatrix} 1 & -\lambda_{1,2} & -\lambda_{1,3} \\ -\lambda_{2,1} & 1 & -\lambda_{2,3} \\ -\lambda_{3,1} & -\lambda_{3,2} & 1 \end{bmatrix} & \times & \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} & = & \begin{bmatrix} \lambda_{1,4} & \lambda_{1,5} & \lambda_{1,6} & \lambda_{1,7} \\ \lambda_{2,4} & \lambda_{2,5} & \lambda_{2,6} & \lambda_{2,7} \\ \lambda_{3,4} & \lambda_{3,5} & \lambda_{3,6} & \lambda_{3,7} \end{bmatrix} & \begin{bmatrix} u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} \\ \mathbf{A} & \times & \mathbf{u} & = & & b_1 \end{matrix} \quad (21-a)$$

$$\begin{matrix} \begin{bmatrix} 1 & -\lambda_{1,2} & -\lambda_{1,3} \\ -\lambda_{2,1} & 1 & -\lambda_{2,3} \\ -\lambda_{3,1} & -\lambda_{3,2} & 1 \end{bmatrix} & \times & \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} & = & \begin{bmatrix} \lambda_{1,4} & \lambda_{1,5} & \lambda_{1,6} & \lambda_{1,7} \\ \lambda_{2,4} & \lambda_{2,5} & \lambda_{2,6} & \lambda_{2,7} \\ \lambda_{3,4} & \lambda_{3,5} & \lambda_{3,6} & \lambda_{3,7} \end{bmatrix} & \begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} \\ \mathbf{A} & \times & \mathbf{v} & = & & b_2 \end{matrix} \quad (21-b)$$

By substituting the values of $\lambda_{i,k}$ and boundary nodes to both matrix equations, the internal nodes mapped onto the 2 D plane can be finally obtained by solving the simultaneous equations. Fig. 7 shows the result of mapping an original model of 5,000 TEU container carrier and another of 300,000 DWT VLCC onto a common 2 D parametric domain, ready for the merging operation.

In order to improve the correspondence between two parent shapes, more feature points can be used for 2 D boundary polygon. On the other hand such a practice would increase the subdivisions of surface which can complicate the situation. The optimum number and position of the feature points, therefore, still remain to be determined. However, this would call for a more systematic investigation and consequently has been left for future research.

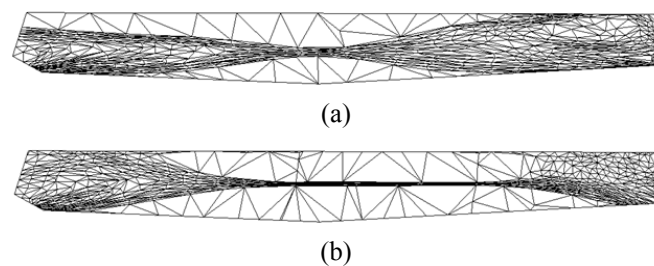


Fig. 7 Parametric meshes obtained by mapping onto a common 2 D polygon: (a) a container ship; (b) a VLCC.

Merging

Any number of parent models can be used for morphing, but, for the sake of clarity and ease of discussion the case of two parent models is discussed here as an example. The task of merging starts with establishing a merged boundary edge set from the two boundary sets of the original parametric meshes. Fig. 8 illustrates an example of the procedure of the boundary merging. Since the nodes of both boundary sets are parameterised from different mesh structures, the number and positions of the nodes are usually not identical as shown in Fig. 8 (a). The boundary AD (red line) with four vertices and EI (blue line) with five

vertices come from the two original model.

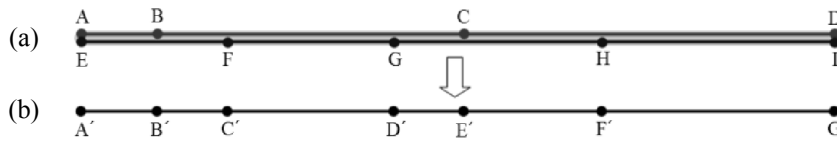


Fig. 8 An example of merging at boundaries.

As shown in Fig. 8 (b), a new point set is created by taking all the vertices on the two edges and removing one from duplicate vertex pair – in this case E and I (or A and D). Some edges are split by the introduction of a new vertex. For example, edge EF (or A'C') is split into EB and BF (A'B' and B'C') by the insertion of vertex B (B'). The nodes of the new vertex set are then arranged in the order of occurrence from one end (from the left end in this case). Thus, the new vertex set is {A', B', C', D', E', F', G'}, and the new edge set is {A'B', B'C', C'D', D'E', E'F', F'G'} in this example.

The concept of merging internal mesh elements is similar to the boundary merging with some additional considerations. Let us take an example case where two parameterised meshes share a same boundary ABCE (black lines), one has edges BE and BD (red), and the other has edge AC (blue) for internal mesh elements as shown in Fig. 9. The obvious constituents of the new vertex set are the existing vertices of the two parametric meshes. To these some new intersection vertices, created by edges crossing, are added (G and F in Fig. 9).

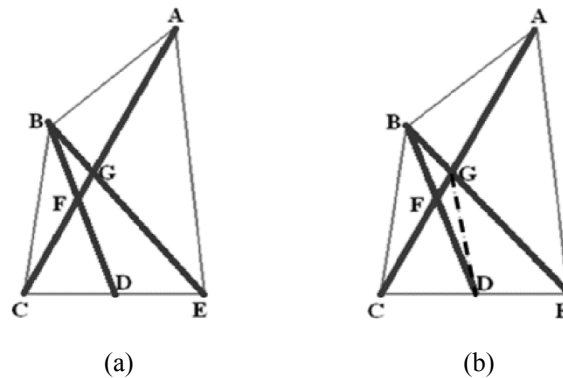


Fig. 9 An example of additional intersection vertices and consequent new triangular faces.

Some edges can be split into two or more segments by the new vertices thus created. If an edge is not crossed by any edge, it simply becomes an element of the new edge set. The new edge set of Fig. 9, for example, will be {AB, BC, CD, DE, EA, AG, GF, FC, BG, GE, BF, FD}. However, this edge set is not yet complete, since the newly created vertices have introduced non-triangular face GFDE, and this face should be split into two triangular faces by generating the edge GD or FE.

These new edges can be found by first checking if each of the neighbouring nodes of a vertex is joined to the next neighbour by an edge. For example, vertex A has three neighbours {B, G, E}, and they are connected by edges {BG, GE}. The neighbours of vertex F, on the other hand, are {B, C, D, G}, but there is no edge between vertices G and D. Consequently the edge GD is created and added to the edge set. To begin with nodes F and E, both neighbours of vertex G, also needed a new edge FE, but with the edge GD newly created, the neighbours of vertex G are now {A, B, F, D, E}. Each neighbour is connected to its next neighbour with edges {AB, BF, FD, DE, EA}, and thus a new edge FE is not required for the vertex G. Consequently, the new edge set will be {AB, BC, CD, DE, EA, AG, GF, FC, BG, GE, BF, FD, GD} in this case.

As briefly mentioned above, non-triangular faces arising from merging two meshes are found and split into triangular faces. By operating this process repeatedly for all the vertices, the complete new edge set is established, ensuring that the resultant mesh is triangular with no non-triangular faces. Fig. 10 (b) shows the parametric hull form mesh which is the result of merging two parameterised parent forms (a) and (c). The merged mesh has all the vertices of the two parent meshes, and there is no duplicate vertex or edge.

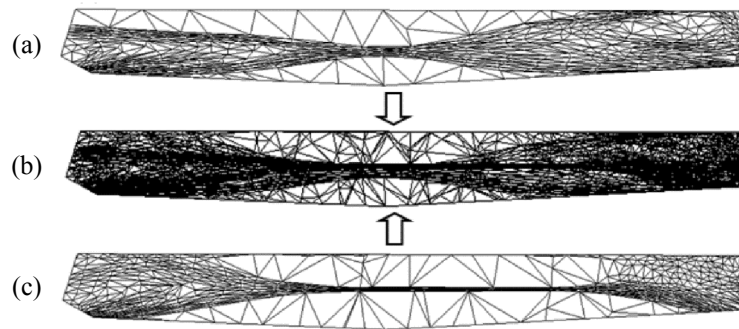


Fig. 10 Parameterised meshes of parent model 1 (a) and parent model 2 (c), and their merged mesh (b).

Remapping

In order to remap the merged parametric meshes back to the two 3 D surfaces, let us assume that the vertex sets of the two 3 D original model meshes are $S_1 = (x_i, y_i, z_i)$ and $S_2 = (x_j, y_j, z_j)$ respectively, and the vertex sets of their parameterised meshes are $P_1 = (\zeta_i, \xi_i)$ and $P_2 = (\zeta_j, \xi_j)$ respectively. Let the vertex set of the merged parameterised mesh be represented as $M = (\zeta_k, \xi_k)$. The vertex set M contains the vertices of both P_1 and P_2 . In addition, P_1 and P_2 have accurate correspondence with S_1 and S_2 respectively, since the vertex sets P_1 and P_2 are parameterised from the vertex sets of S_1 and S_2 . If the numbers of vertices in S_1 , S_2 and M are n_1, n_2 and n_3 respectively, then, $i = 0, 1, \dots, n_1, j = 0, 1, \dots, n_2$ and $k = 0, 1, \dots, n_3$.

For remapping the parametric mesh onto the first 3 D parent model, the vertices of the merged mesh $M = (\zeta_k, \xi_k)$ need to be divided into two groups: 'P₁ group' and 'non-P₁ group'. If a vertex of $M = (\zeta_k, \xi_k)$ is an element of $P_1 = (\zeta_i, \xi_i)$, the vertex belongs to the 'P₁ group', and its corresponding vertex of $S_1 = (x_i, y_i, z_i)$ can easily be found. If not, however, the vertex is put into the 'non-P₁ group', and it should be remapped to the 3 D surface with barycentric coordinates of $P_1 = (\zeta_i, \xi_i)$.

A similar operation can be carried out with the second parent model. Unlike remapping of the vertex set, the edge and face sets of the parametric merged mesh can be directly used for both of the remapped meshes, since the structure of remapped 3 D meshes are identical to the one of the merged parameterised mesh.

Fig. 11 shows the original parent model meshes ((a) and (b)), with their merged meshes ((c) and (d)). The mesh structures of the original parent models are quite different, but it is apparent that the two re-meshed models have an identical mesh structure.

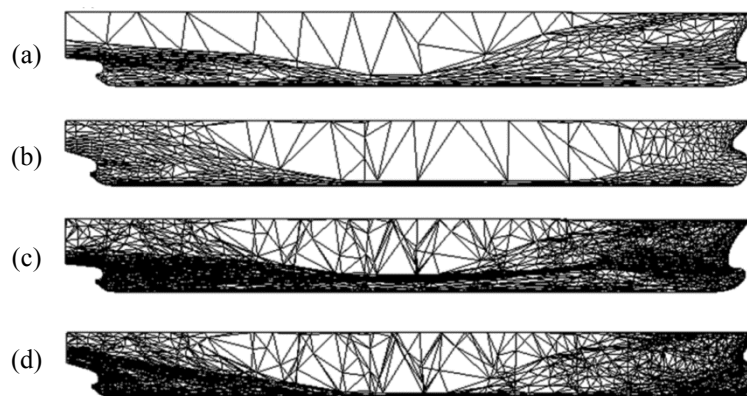


Fig. 11 Original parent model 1 (a) and parent model 2 (b), and their re-meshed models (c) and (d).

MORPHING

With the two re-meshed models holding an identical mesh structure, a perfect correspondence is established and there is no problem now in proceeding with morphing proper. In this paper, only linear morphing has been investigated and it is defined by:

$$M(t) = (1-t) \cdot P_0 + t \cdot P_1, \quad t \in [0,1] \tag{22}$$

where t is the morphing parameter, $M(t)$ is the generated model and P_0 and P_1 denote the source and target models respectively. The model at $t = 0$ is equivalent to the first parent model while the model at $t = 1$ corresponds to the second parent model. Since t can take on an unlimited number of values between 0 and 1, an infinite number of intermediate forms can be generated. Fig. 12 shows examples of intermediate hull forms in various views (in this example $t = 0.0, 0.2, 0.4, 0.6, 0.8$ and 1.0) using the merging method. The two parent models are a 5,000 TEU container carrier and a 300,000 DWT VLCC.

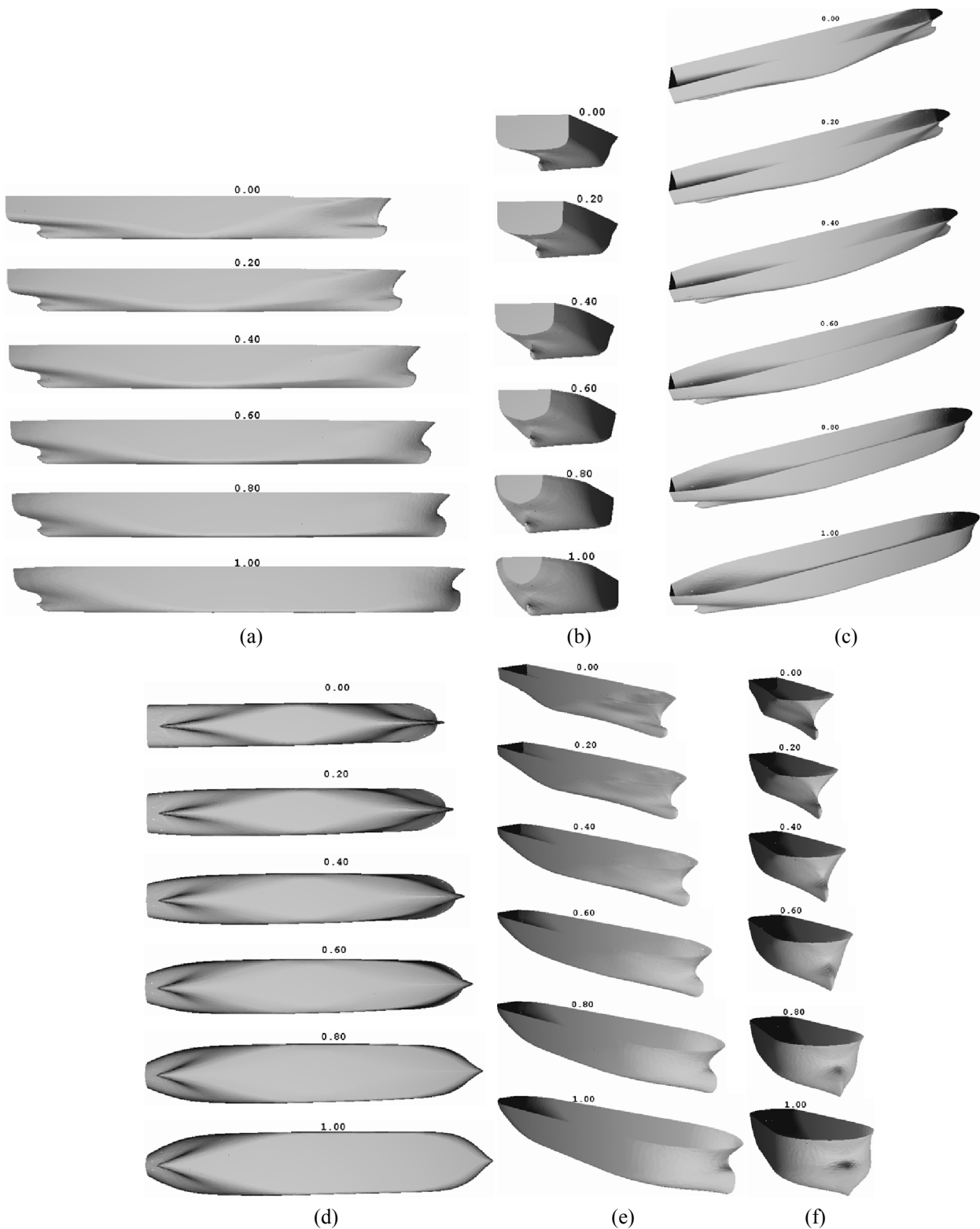


Fig. 12 Hull form morphing using the merged method.

CONCLUSIONS

It has been found that the morphing technique allows an infinite number of predictable hull forms to be generated rapidly on demand. One of the key steps in 3 D form morphing is establishing mesh correspondence. This is achieved by re-meshing and in this paper the mesh-merging technique was successfully applied to this crucial process. Its alternative, mesh-regularising method is possibly much faster in the remeshing algorithm itself. However, mesh-merging requires much less manual intervention and manipulations, making it ultimately more efficient and less error-prone. In any case the time penalty is negligible, measured in seconds rather than hours. It took approximately 15 *seconds* using the mesh-merging approach to re-mesh a container ship model of 550 vertices and 993 faces and a VLCC model of 559 vertices and 993 faces, so that both of them had identical mesh structure with 4900 vertices and 9788 faces. The computer used was a PC with Intel Core i5-480M Processor and 4 GB of RAM. Fig. 13 illustrates comparison of the re-meshed container ship model by the mesh-regularising (b) and merging (c) methods from original model (a).

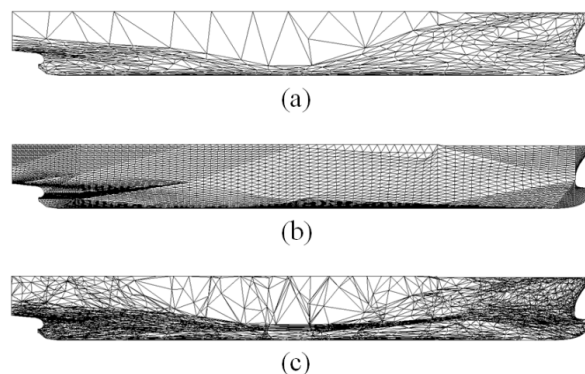


Fig. 13 Original mesh of a 5,000 TEU container carrier ship and its re-meshed models by mesh-regularising and merging methods: (a) original mesh; (b) regularised mesh; (c) merged mesh with a VLCC model mesh.

It was also found that two models of widely differing shapes, such as a mono hull and catamaran hull, can be morphed, although this paper does not mention it in detail. Moreover, it is also possible to apply morphing to local features, such as bulbous bow and flare. These exercises will serve as an example of morphing as a powerful tool of idea generation.

Much work still needs to be done before morphing can be used as a practical design tool. One of the uncertainties is the question of the continuity of the surfaces created through morphing. On the face of it, a form generated by linear interpolation of two forms both of which have continuity of degree k would have similar continuity. However, this requires a more rigorous proof. Application of surface smoothing algorithms such as Laplacian smoothing may also be required.

ACKNOWLEDGEMENTS

Part of this research has been carried out with the support of the research programme 'Improved ship design and operation, by operational data aggregation, key performance indices and numerical optimization (ImproVEDO)' headed by SINTEF Fisheries and Aquaculture, and funded by Rolls-Royce Marine and the Norwegian Research Council.

REFERENCES

- Alef, W.E. and Collatz, G., 1976. Computer aided design of ship's lines by nonlinear distortion of parent forms. *Proceedings of ICCAS 76*. pp.157-163.
- Alexa, M., 2000. Merging polyhedral shapes with scattered features. *The Visual Computer*, 16(1), pp.25-37.
- Bézier, P.E., 1968. *How renault uses numerical control for car body design and tooling*, SAE Paper 680010. Detroit, MI: Society of Automotive Engineers Congress.
- Chen, E. and Parent, R., 1989. Shape averaging and its applications to industrial design. *IEEE Computer Graphics and Applications*, 9(1), pp.47-54.

- Cox, M.G., 1972. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10(2), pp.134-149.
- Creutz, G. and Schubert, C., 1978. Interactive curve creation from form parameters by means of B-splines. *Schiffstechnik*, 25(122), pp.121-145.
- De Boor, C., 1971. CADRE: An algorithm for numerical quadrature. In: J. R. Rice, ed. *Mathematical Software*. New York and London: Academic press. pp.201-209.
- Floater, M., 1997. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14, pp.231-250.
- Gordon, W. and Riesenfeld, R.F., 1974. B-spline curves and surfaces. In: Barnhill, R.E, Riesenfeld, R.F., eds. *Computer Aided Geometric Design*. New York: Academic Press. pp.95-126.
- Gregory, A., State, A., Lin, M.C., Manocha, D. and Livingston, M.A., 1998. Feature-based surface decomposition for correspondence and morphing between polyhedra. *Proceedings of computer animation*.
- Harries, S. and Abt, C., 1998. Parametric curve design applying fairness criteria. *international workshop on creating fair and shape-preserving curves and surfaces*. Network Fairshape, Berlin/Potsdam; Teubner.
- Hsiao, S.W. and Liu, M.C., 2002. A morphing method for shape generation and image prediction in product design. *Design Studies* 2002, 23(6), pp.533-556.
- Kang, J.Y. and Lee, B.S., 2009. An investigation of parametric rolling and its reduction through geometric design. *Proceedings of greenTech-International conference on marine science & technology for green shipping*. Glasgow, pp.131-141.
- Kang, J.Y. and Lee, B.S., 2010. Mesh-based morphing method for raid hull form generation. *Computer-Aided Design*. 42(11). pp.970-976.
- Keane, A.J., 1988. A Computer based method for hull form concept design: application to stability analysis. *Trans RINA*, 130.
- Kent, J., Parent, R. and Carlson, W., 1991. Establishing correspondences by topological merging: A new approach to 3-D shape transformation. *Proceedings of graphics interface '91*. pp.271-278.
- Kent, J., Carlson, W. and Parent, R., 1992. Shape transformation for polyhedral objects. *SIGGRAPH '92*. pp.47-54.
- Kim, H.C., 2004. *Parametric design of ship hull forms with a complex multiple domain surface topology*. Ph.D. Technische Universität Berlin.
- Kuiper, G., 1970. Preliminary design of ship lines by mathematical methods. *Journal of ship Research*, 14(1), pp.52-66.
- Lackenby, H., 1950. On the systematic geometrical variation of ship forms. *Transaction INA*, 92. pp.289-316.
- Munchmeyer, F.C., Schubert, C. and Nowacki, H., 1979. Interactive design of fair hull surfaces using B-splines. *Computer applications in the automation of shipyard operation and ship design III. Proceedings of IFIP*. North-Holland Publishing Company.
- Nowachi, H., 2010. Five decades of computer-aided ship design. *Computer-Aided Design*, 42(11), pp.956-969.
- Rabien, V., 1979. Ship surface design by transforming given mesh representations. *Proceedings of ICCAS 79*. pp.85-93.
- Reed, A.M. and Nowacki, H., 1974. Interactive creation of fair ship lines. *Journal of Ship Research*, 18(2).
- Riesenfeld, R.F., 1972. *Application of B-spline approximation to geometric problems of computer aided design*. Ph.D. Syracuse University.
- Sharma, R., Kim, T.W., Storch, R.L., Hopman, H.(J.J.) and Erikstad, S.O., 2012. Challenges in computer applications for ship and floating structure design and analysis. *Computer-Aided Design*, 44(3), pp.166-185.
- Taylor, D.W., 1915. Calculation of ships' forms and light thrown by model experiments upon resistance, propulsion and rolling of ships. *Trans. International Engineering Congress*. San Francisco.
- Tutte, W.T., 1959. Convex representations of graphs. *Proceedings of London mathematical society*. 10, pp.304-320.
- Versluis, A., 1977. Computer aided design of ship form by affine transformation. *International Shipbuilding Progress*, 24 (274), pp.147-160.
- Zhang, P., 2008. Parametric approach to design of hull forms. *Journal of Hydrodynamics*, 20(6), pp.804-810.