© Strojniški vestnik 49(2003)4,230-241
ISSN 0039-2480
UDK 004.021:629,5.023.9:531.72
Predhodna objava (1.03)

© Journal of Mechanical Engineering 49(2003)4,230-241
ISSN 0039-2480
UDC 004.021:629,5.023.9:531.72
Preliminary paper (1.03)

Hitri algoritem za rekonstrukcijo in odvoj digitaliziranih ploskev

A Fast Algorithm for Reconstructing and Flattening Digitised Surfaces

Simon Kolmanič - Nikola Guid

Odvoj ploskev je bistvenega pomena za načrtovanje ravninskih vzorcev, potrebnih za izdelavo npr.: letalskih kril, ladijskih trupov, delov avtomobilske pločevine, zgornjih delov čevljev in oblačil. Čeprav lahko za gradnjo sodobnih ladijskih trupov uporabljamo tudi kompozitne materiale, pa ostaja jeklo v ladjedelništvu še vedno najpogosteje uporabljan material. Zaradi tega ima postopek načrtovanja ravninskih vzorcev še vedno pomembno vlogo pri konstrukciji ladij. Odvoj večjega dela ladijskega trupa je dokaj preprost, izjema sta le premec in krma, ki ju odlikuje zahtevna oblika, zaradi česar potrebujemo učinkovit algoritem za odvoj ploskev. V tem prispevku je predstavljen novi hitri algoritem za rekonstrukcijo in odvoj ploskev, ki temelji na strategiji "deli in vladaj". Za rekonstrukcijo ploskve se uporabljajo odvojni trakovi. Na ta način se lahko ploskev odvije v ravnino brez deformacij.

© 2003 Strojniški vestnik. Vse pravice pridržane.

(Ključne besede: ploskve odvite, ploskve premonosne, vzorci ravninsi, ploskve digitalizirane)

The problem of surface flattening is an important one in pattern engineering. Surface flattening is needed for the design of thin-walled objects, such as airplanes' wings, outer ship hulls, parts of car bodies, shoe uppers, and textile products. Although composite materials can be used for building modern ships' hulls, steel is still the most commonly used material for the ship-building industry. Therefore, the pattern engineering process still has an important role in ship designing. For most of the ship's hull the flattening is quite simple; the problem is the flattening of the stem and the stern, because the surfaces of these parts of the ship are very complex, and therefore an efficient surface-flattening algorithm is needed. In this article a new fast algorithm for surface reconstruction and surface flattening, based on a divide-and-conquer strategy, is presented. Developable stripes are used to approximate the surface, and in this way the surface can be flattened quickly and without any distortion.

© 2003 Journal of Mechanical Engineering. All rights reserved.

(Keywords: surface flattening, developable surfaces, ruled surfaces, flat pattern, digital surfaces)

0 UVOD

Odvoj ploskev je že dolgo znan in zelo razširjen problem. Najdemo ga tako v kartografiji kakor tudi v različnih vejah industrije. Odvoj ploskev je preslikava 3D ploskve v ravnino, pri čemer moramo ohranjati razdalje med točkami ploskve. Kot rezultat te preslikave dobimo ravninski vzorec. Na žalost pa lahko brez napak, ki se kažejo kot prekrivki in vrzeli v nastalem ravninskem vzorcu, odvijemo le določene tipe ploskev. Te ploskve imenujemo odvojne ploskve in v splošnem velja, da poljubne ploskve ne moremo odviti v ravnino brez deformacij [1]. Jasno je, da želi načrtovalec ravninskih vzorcev deformacije zmanjšati na najnižjo mogočo raven. To je težko avtomatizirati, čeprav obstajajo metode, ki

0 INTRODUCTION

The problem of surface flattening is an old one, and is common in cartography and in various branches of industry. Surface flattening is the mapping of a 3D surface onto a 2D plane, where the distances between surface points have to be preserved. As a result of this operation a flat pattern is generated. Unfortunately, only special types of surfaces can be unrolled onto a plane without errors that result in tearing and overlapping in the generated, flat pattern. These surfaces are known as developable surfaces [1]. It is clear that pattern designers would like to reduce distortions to a minimum. This is very hard to do automatically, although some methods of significantly reducing the distortions do already exte deformacije občutno zmanjšajo ([2] in [3]). Posebno težak problem pri načrtovanju vzorcev so prekrivki v končnem ravninskem vzorcu [4]. Prekrivki v ravninskem vzorcu namreč pomenijo vrzel na 3D ploskvi, ki jo iz takega vzorca dobimo. V kartografiji je ta problem rešen z uporabo raznih projekcij, toda odvoj industrijskih predmetov je veliko zahtevnejši, ker so ti pogosto dvojno ukrivljeni. V tem primeru je namreč prekrivke še posebej težko odstraniti [4]. Problem odvoja ploskev je znan tudi v ladjedelništvu, saj je ladijski trup sestavljen iz ravninskih materialov. Na tem mestu je treba omeniti, da se je v zadnjih dveh desetletjih pri izdelavi ladijskih trupov močno povečala uporaba kompozitnih materialov, na primer steklena vlakna, saj jih odlikuje izjemno razmerje med trdnostjo in težo. So tudi zelo upogljivi in zato nadvse primerni za oblikovanje zahtevnih ploskev. Odvoj ploskev, ki jih želimo izdelati s kompozitnimi materiali, je poseben problem, ki ga ne moremo primerjati z odvojem ploskev, izdelanih iz jekla in podobnih materialov. Ta problem je podrobneje obdelan v [5] in ne bo predmet obravnave v tem prispevku. Kljub vsem prednostim kompozitnih materialov pa ostaja jeklo v ladjedelništvu še vedno najpogosteje uporabljan material. Odvoj takih ploskev je v bistvu običajen primer odvoja ploskev, znan tudi v čevljarski in podobnih vejah industrije.

Ker je izdelava ravninskih vzorcev iz 3D ploskev že dolgo znan problem, obstaja mnogo metod, ki ga skušajo rešiti. Te metode smo razdelili v dve skupini: metode, ki skušajo odviti ploskev v enem kosu, in metode za odvoj ploskev po delih [6]. Metode iz prve skupine niso najbolj primerne za odvoj poljubnih ploskev, saj prekrivke v dobljenem ravninskem vzorcu odpravijo le, če je ploskev odvojna. Metode iz druge skupine ploskev razdelijo v manjše krpe, ki jih lažje odvijejo v ravnino in ob tem prihaja do manj deformacij.

Delitev ploskve na manjše krpe je odvisna od samo enega numeričnega parametra in ne zahteva dodatnega uporabnikovega posredovanja. Metoda, predstavljena v tem prispevku, temelji na podobni zamisli. V sistemih računalniško podprtega načrtovanja (RPN - CAD) so ploskve večinoma predstavljene z množico točk in trikotnikov, ki jih povezujejo. Včasih pa se zgodi, da je ploskev predstavljena samo z oblakom točk v 3D prostoru. V tem primeru je treba ploskev poprej rekonstruirati. Za to nalogo smo uporabili odvojne trakove. Po končani rekonstrukciji je odvoj ploskve preprost, saj moramo v ravnino odviti le odvojne trakove. Zamisel odvoja ploskev po delih ni nova, predhodno jo je uporabljal že Elber [7], nato pa še Hoschek za odvoj vrteninskih ploskev [8]. Le-to smo priredili in jo prilagodili delu z digitaliziranimi ploskvami. Postopek približkov krmili samo en numerični parameter in je zaradi tega preprost za uporabo.

ist ([2] and [3]). The biggest problem in pattern generation is overlaps in the generated pattern [4]. These overlaps in the flat pattern represent gaps in the 3D surface obtained by such a pattern. In cartography this problem is already solved by various projections, but the flattening of industrial objects is far more complicated because they are often double curved. In this case the overlaps are especially difficult to eliminate [4]. The problem of surface flattening is also present in the ship-building industry because ships' hulls are assembled from flat material. We should mention here that in the last two decades the use of advanced composite materials, such as fibreglass, for the manufacture of ships' hulls has increased because of their remarkable strength-to-weight ratio and high pliability when deformed into complex, curved shapes. The problem of the flattening of surfaces to be made of a composite material is very specific and cannot be compared with the flattening of surfaces to be made from steel or similar materials. This problem is further discussed in [5], and it will not be covered in this paper. In spite of the advantages of composite materials steel remains the most used material when building ships' hull. The flattening of such a surface is a classical flattening problem that is also encountered in the shoe and similar industries.

Since deriving patterns from 3D surfaces is an old problem, many methods already exist. We have divided them into two groups: methods for flattening the surfaces in one piece and methods for flattening the surfaces using more than one piece [6]. The methods of the first group are not suitable for the automatic flattening of arbitrary surfaces, since the overlaps are eliminated from the generated pattern only if the surface is developable. The methods of the second group divide the surface into smaller patches, which can be flattened more easily and therefore with fewer distortions.

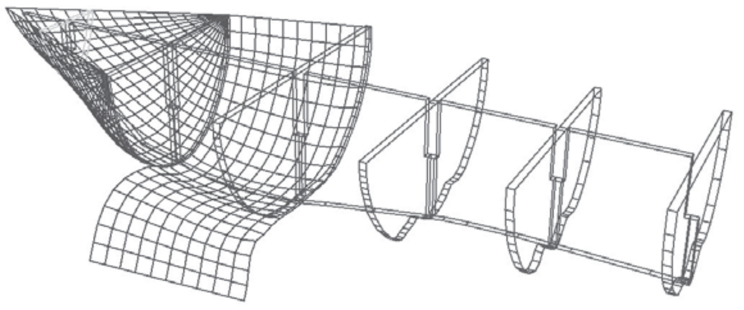
A division of the surface into smaller patches is controlled by simple numerical parameters and does not require any additional interference by the user. The method presented in this paper is based on a similar idea. In computer aided design (CAD) systems the surfaces are usually represented by a set of points and the triangles that connect them. But sometimes the surface is represented only by a cloud of 3D points. In this case the surface has to be reconstructed first. We have used developable stripes to do this. After the surface reconstruction is completed, the flattening is easy, since we only have to unroll the developable stripes onto the plane. The idea of multi-parts surface flattening is not new. Originally it was used by Elber [7] and by Hoschek to flatten surfaces of revolution [8]. We have adopted the Hoschek idea to work with surfaces obtained by digitisation. The approximation process is controlled by only one numerical parameter, and therefore it is very simple to use.

1 PROBLEM ODVOJA PLOSKEV V LADJEDELNIŠTVU

Ladjedelništvo ima že dolgo zgodovino. Stoletja so gradili ladje, podobne telesom vretenčarjev. Ladijska rebra so prekrili z lupino iz usnja, lubja, desk ali jeklenih plošč. Medtem ko ladijska rebra popolnoma določajo obliko ladijskega trupa in mu dajejo trdnost, omogoča lupina ladijskemu trupu vodotesnost in plovnost. Lupina poleg tega daje ladijskemu trupu potrebno trdnost, da vzdrži kombinacijo sil: navzgor usmerjeno silo vzgona, navzdol usmerjeno silo teže in močno silo morskih valov.

1 THE PROBLEM OF SURFACE FLATTENING IN THE SHIP-BUILDING INDUSTRY

The ship-building industry has a long history. For hundreds of years ships have been built like the bodies of vertebrate animals: the ships' ribs have been covered with hide, bark, planks or metal plates. The form of the ship's ribs completely defines the form of the ship, while the skin makes the ship watertight and buoyant. The skin also provides the ship's hull with the necessary strength to withstand a combination of forces: the upward force of buoyancy, the downward force of gravity, and the force of sea waves.



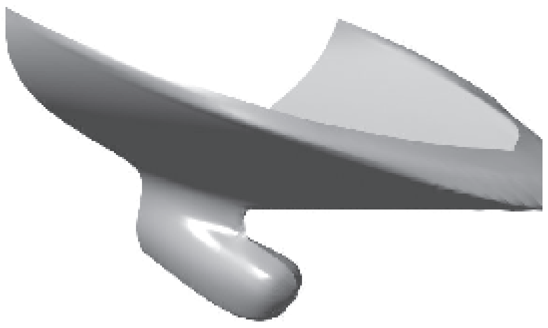
Sl. 1. *Izdelava lupine ladijskega trupa z uporabo ravninskega materiala*
Fig. 1. *Generation of a ship's outer hull using a plane material*

Glede na to, da je lupina izdelana iz ravninskega materiala, moramo pred izdelavo ladje rešiti problem odvoja ploskve. Če je oblika ladijskega trupa nezahtevna, je to dokaj preprosto (sl. 1). Toda če želimo, da je ladja hitra, je oblika ladijskega trupa veliko zahtevnejša. Da zmanjšamo zaviranje valov in trenja, uporabimo premec z zaobljenim profilom, na krmi pa dodamo koleno ([9] in [10]). Premec z zaobljenim profilom in koleno sta zahtevni 3D ploskvi (sl. 2), kar močno otežuje postopek odvoja celotnega ladijskega trupa. Nove oblike ladijskih trupov se dandanes načrtujejo v posebnih sistemih RPN/RPI (računalniško podprte izdelave) z vgrajenimi metodami računalniške dinamike tekočin (RDT - CFD), ki jih uporabljamo za optimizacijo ploskve. Dodatno lahko ploskev optimiziramo tudi v testnih bazenih, kjer lahko merimo zaviranje zaradi trenja. Pred začetkom uporabe sistemov RPN/RPI in metod RDT so bili testni bazeni glavni pripomoček pri načrtovanju novih tipov ladijskih trupov. Ladijski modeli so bili tako najprej izboljšani in nato razžagani v prereze. Vsak prerez je natančno izmerjen, na podlagi meritev pa je nato izdelan načrt prereza. Načrte nato ustrezno povečajo, da ustrezajo dejanski velikosti ladje. Iz teh načrtov nato izdelajo posamezne dele, ki jih sestavijo v ladijski trup naravne velikosti.

Geometrijsko obliko modela ladijske lupine lahko dobimo tudi z njeno digitalizacijo, ki jo izvedemo na vzporednih ravninah v prostoru. Enako ustvarjeno množico točk lahko približamo s 3D ploskvijo ([11] do [15]). Da lahko dobimo načrt za ravninski material lupine, je treba tako dobljeno ploskev odviti. Večina

Since the skin is made of a flat material, it is clear that the surface flattening problem has to be solved before the ship can be built. If the form of the ship's hull is simple, this is quite easy (see Figure 1). However, if the ship is to be fast, the hull shape will need to be more complex: to reduce friction and wave drag, a bulb is added to the bow of the ship and at the stern skegs can be added ([9] and [10]). The shapes of the bulb and the skegs are complex 3D surfaces (see Figure 2), which makes the flattening process of the ship's hull difficult. Today, new hull forms are designed in special CAD/CAM (Computer Aided Manufacturing) systems with integrated computational fluid dynamics (CFD) methods, which are used for surface optimisation. In addition the surface can be optimised in testing pools, where the drag on the ship's model can be measured. The designing of new ships' hulls using testing pools was the main design method before CAD/CAM systems and CFD methods were used. After the models were approved, they were sawed into cross sections; blueprint measurements were made from each cross section; these were then mathematically enlarged so that the ship could be built to full scale. From the full-scale measurements, parts were constructed and assembled to build the full-sized hull.

The geometry of the ship's hull model can also be obtained by surface digitising, made on parallel planes in 3D space. The set of points generated in this way can be approximated by a 3D surface ([11] to [15]). To get the blueprint for the plane material of the skin, the obtained surface has to



Sl. 2. *Ploskev testnega premca z zaobljenim profilom za zmanjšanje trenja*
Fig. 2. *The bow surface of the test hull with the bulb for friction-drag minimisation*

metod, uporabljenih v današnjih tržnih paketih RPN, ki temeljijo na geometrijskem načinu odvoja, je približnih. Izvirno 3D ploskev približajo s trikotniki, ki sestavljajo 3D mrežo. Pri odvoju nato izberemo začetni trak ter smer odvoja, nakar odvijemo prvi trak v ravnino. Preostali del ploskve nato odvijemo na podoben način ([2] in [16]). Glavna pomanjkljivost teh metod so prekrivki in razpoke med trakovi v nastalem ravninskem vzorcu. Do omenjenih nepravilnosti prihaja zaradi kotne okvare, ki jo najdemo v definiciji Gaussove ukrivljenosti v posamezni točki triangulacijske ploskve [16]:

be flattened. Most of flattening methods used in commercial CAD systems that are based on a geometrical approach are approximate. The original 3D surface is approximated by triangles forming a 3D grid. Then the starting strip and the direction are chosen, along which these triangles are unfolded into the plane. The rest of the surface is unfolded in similar fashion ([2] and [16]). The main disadvantage of these methods is the presence of gaps and overlaps between the stripes in the generated, flat pattern. These anomalies arise from the angular defect, which can be found in the definition of the Gaussian curvature at a given point on the triangulated surfaces [16]:

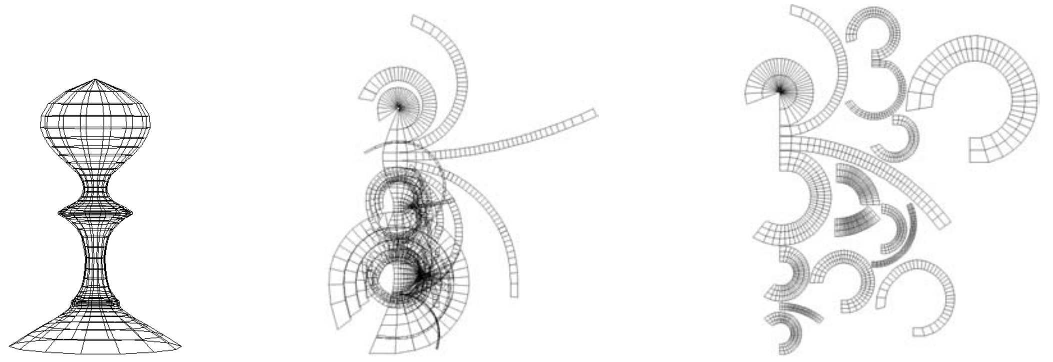
$$K = \frac{\delta}{S} \qquad (1),$$

kjer je δ kotna okvara, ki je enaka kotu 2π, zmanjšanem za vsoto notranjih kotov trikotnikov pri dani točki, *S* pa je enak 1/3 ploščine teh trikotnikov. Iz enačbe (1) je razvidno, da je kotna okvara odvisna od vrednosti Gaussove ukrivljenosti v dani točki in od dane triangulacije, saj z manjšimi trikotniki zmanjšamo tudi kotno okvaro in s tem napako v ravninskem vzorcu. Ker pa je v našem primeru triangulacija odvisna od algoritma za rekonstrukcijo predmeta, to ne pomaga veliko. Na število napak v ravninskem vzorcu pomembno vpliva tudi izbira začetnega traku in smeri odvoja, kar pa je prav tako zahteven problem.

Da v celoti odpravimo prekrivke v ravninskem vzorcu, obenem pa se izognemo težavam pri izbiri začetnega traku in smeri odvoja, smo uporabili zamisel odvoja ploskev po delih [6]. Tudi v tem primeru gre za približni postopek, vendar je kakovost ravninskega vzorca boljša. Razliko lahko prikažemo že na preprosti vrteninski ploskvi, prikazani na sliki 3a. Metoda, temelječa na Gaussovi ukrivljenosti [16], vrne ravninski vzorec, prikazan na sliki 3b, ki je zaradi številnih prekrivkov neuporaben. Ravninski vzorec na sliki 3c je dobljen z metodo, temelječo na strategiji “deli in vladaj” [6], ki ploskev razdeli na posamezne, med seboj neodvisne trakove. To metodo smo tudi izbrali za izhodišče pri razvoju metode za rekonstrukcijo in odvoj digitaliziranih ploskev.

where δ is the angular defect equal to 2π less the sum of the interior angles meeting at the given point, and *S* is equal to 1/3 of the areas of those triangles. In equation 1 it can be seen that the angular defect depends on the value of the Gaussian curvature at the given point and for a given triangulation, since the angular defect, and with it also the error in the flat pattern, can be decreased by the use of smaller triangles. Because in our case the triangulation depends on the object-reconstruction algorithm, this is not helpful. The number of errors in the flat pattern is related to the choice of the starting strip and the unfolding direction, which is also a complex problem.

To eliminate completely the overlaps in the flat pattern and, at the same time, to avoid the difficulties associated with the choice of starting strip and unfolding direction, we use the approach of per partes surface flattening [6]. This is also an approximate approach, but the quality of the pattern is better. The difference can be shown on simple surfaces of revolution, shown in Figure 3a. The method, based on Gaussian curvature [16], generates the pattern shown in Figure 3b, which is unusable because of the numerous overlaps in it. The pattern shown in Figure 3c is generated by a method based on a divide-and-conquer strategy [6], which divides the surface into a set of independent stripes. We have chosen this method as the origin for developing a method for the reconstructing and flattening of digitized surfaces.



Sl. 3. *a) Preprosta vrteninska ploskev*
b) Ravninski vzorec, dobljen z metodo, temelječo na [16], ob slabo izbranem začetnem traku.
c) Ravninski vzorec, dobljen z metodo, temelječo na strategiji "deli in vladaj" [6]
Fig. 3 *a) Simple surface of revolution*
b) Flat pattern generated by the method based on [16] with a badly chosen starting strip
c) Flat pattern generated by the method based on a divide-and-conquer strategy [6]

Množico 3D točk približamo z množico odvojnih trakov. Metoda za rekonstrukcijo ploskev, temelječa na uporabi odvojnih trakov, je predstavljena bolj podrobno v naslednjem razdelku.

2 REKONSTRUKCIJA PLOSKEV Z UPORABO ODVOJNIH TRAKOV

Digitalizacijo izvedemo tako, da točke ležijo v vzporednih ravninah. Zaradi splošnosti je v ravninah dovoljeno različno število točk. Če v vsaki ravnini med seboj povežemo točke v vrstnem redu, določenem z digitalizacijo, dobimo krivulje prečnih prerezov, ki jih bomo označevali s $\mathbf{c}^i$, $i = 1, 2, \ldots, N$. V [6], [17] lahko vidimo, da je med dvema krivuljama prečnih prerezov mogoče skonstruirati odvojni trak. Tem krivuljam pravimo tudi vodila. Odvojni trak je posebna premonosna ploskev [1]. Daljico, ki povezuje dve točki na vodilih, imenujemo generator [17]. Rekonstrukcijo ploskve začnemo z izvedbo odvojnega traku med prečnima prerezoma $\mathbf{c}^1$ in $\mathbf{c}^N$. To dosežemo s klicem postopka `divide(1, N)`, ki uporablja strategijo "deli in vladaj" (izpis 1). Nato je treba izračunati presečišča med odvojnim trakom in ravninami, v katerih ležijo vmesni prečni prerezi. Če je razlika med izračunanimi presečišči in danimi prečnimi prerezi prevelika, odvojni trak z uporabo strategije "deli in vladaj" zožimo. Postopek večkrat ponavljamo, dokler ne dosežemo želene natančnosti. Če zapišemo problem bolj splošno, velja, da odvojni trak, $\mathbf{s}(\mathbf{c}^b, \mathbf{c}^f)$, napet med krivuljama $\mathbf{c}^b$ in $\mathbf{c}^f$ $(1 \leq b < f \leq N)$, seka ravnine prečnih prerezov $\mathbf{c}^k$, $b < k < f$ (sl. 4).

Presečišče $\mathbf{s}_j^k = \left(s_{xj}^k,\ s_{yj}^k,\ s_{zj}^k\right)$ med j-tim generatorjem odvojne ploskve, to je daljico, ki povezuje j-to točko na b-ti prečni krivulji, tj. $\mathbf{c}_j^b = \left(c_{xj}^b,\ c_{yj}^b,\ c_{zj}^b\right)$, in j-to točko na f-ti prečni

The set of 3D points is approximated by the set of developable stripes. In the next section the method of surface reconstruction using the approximation with developable stripes is explained in more detail.

2 SURFACE RECONSTRUCTION USING DEVELOPABLE STRIPES

The surface digitisation is performed in such a way that all the points lie in parallel planes. Generally, there are a different number of points in the planes. If line segments connect the points in each plane in the order given by the digitising process, cross-section curves are generated that will be denoted by $\mathbf{c}^i$, $i = 1, 2, \ldots, N$. As shown in [6] and [17], a developable surface can be constructed between two cross-section curves. These kinds of curves are also known as directrix curves. The developable stripe is a special case of a ruled surface [1]. A line segment connecting two points on directrices is called a linear generator [17]. The surface reconstruction is started by the generation of a developable stripe between the cross-section curves $\mathbf{c}^1$ and $\mathbf{c}^N$. This can be done with the call of the procedure `divide(1, N)`, based on the divide-and-conquer strategy (see listing 1). Then the intersecting points between the generated developable stripe and the cross-section planes lying in between have to be calculated. As long as the difference between the calculated intersection and cross-section points is not small enough, the developable stripe is narrowed with the help of the divide-and-conquer strategy. The procedure is recursively repeated until the desired precision is achieved. If we state this problem more generally, it is necessary that the developable stripe $\mathbf{s}(\mathbf{c}^b, \mathbf{c}^f)$ between the cross-section curves $\mathbf{c}^b$ and $\mathbf{c}^f$ $(1 \leq b < f \leq N)$ intersects the planes of the cross-section curves $\mathbf{c}^k$, $b < k < f$ (see Fig. 4).

The intersection point $\mathbf{s}_j^k = \left(s_{xj}^k,\ s_{yj}^k,\ s_{zj}^k\right)$ between the j-th surface generator, this is the line segment connecting the j-th point of the b-th cross-section curve, $\mathbf{c}_j^b = \left(c_{xj}^b,\ c_{yj}^b,\ c_{zj}^b\right)$, and the j-th point

Izpis 1. *Algoritem za rekonstrukcijo ploskev, temelječ na strategiji "deli in vladaj"*
Listing 1. *Algorithm for surface reconstruction based on the divide-and-conquer strategy*

```
procedure divide(b, f)
 Vhod/Input:   indeksa vodilj/indexes of directrix curves
 Izhod/Output:  množica odvojnih trakov/set of developable stripes

begin
  Generiraj odvojni trak med prečnima prerezoma z indeksoma b in f./
  Generate a developable stripe between cross sections b and f.

  if curveDifference(b, f) > e then
   begin
    if f-b = 1 then /* končni pogoj rekurzije/ end condition of recursion
*/
      Shrani odvojni trak v množico odvojnih trakov./
      Save the developable stripe in a set of developable stripes.
     else
      begin          /* w določa enačba 6/ w is defined by Eq. 6*/
       divide(b, w); /* rešitev levega podproblema/ solution of the left
subproblem */
         divide(w, f); /* rešitev desnega podproblema/ solution of the
right subproblem */
      end
    end
  else
    Shrani odvojni trak v množico odvojnih trakov./
    Save the developable stripe into a set of developable stripes.
end
```

krivulji, tj. $\mathbf{c}_j^f = \left(c_{xj}^f,\ c_{yj}^f,\ c_{zj}^f\right)$, ter k-to ravnino prečnega prereza izračunamo z naslednjimi enačbami:

of the f-th cross-section curve, $\mathbf{c}_j^f = \left(c_{xj}^f,\ c_{yj}^f,\ c_{zj}^f\right)$, and the k-th cross section plane is calculated using the following equations:

$$s_{xj}^k = -\frac{Ba + Cb + D}{A + Bl + Ch}$$
$$s_{yj}^k = l s_{xj}^k + a \qquad (2),$$
$$s_{zj}^k = h s_{xj}^k + b$$

kjer so A, B, C, in D parametri k-te ravnine prečnega prereza ($Ax + By + Cz + D = 0$), ki jih določimo iz naslednjih enačb:

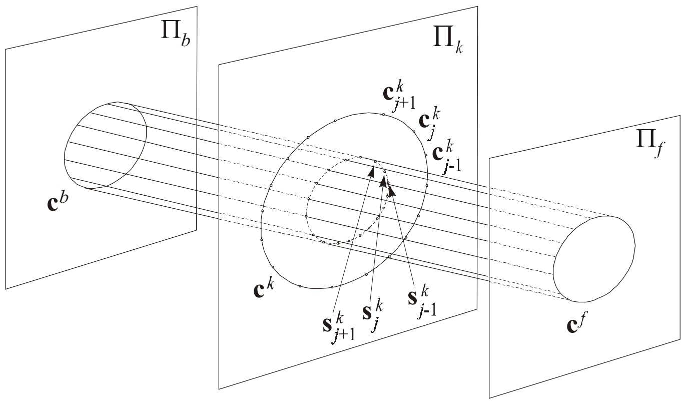
where A, B, C, and D are parameters of the k-th cross-section plane ($Ax + By + Cz + D = 0$), determined by the following equations:

$$\begin{aligned}
A &= c_{yu}^k (c_{zv}^k - c_{zt}^k) + c_{yv}^k (c_{zt}^k - c_{zu}^k) + c_{yt}^k (c_{zu}^k - c_{zv}^k) \\
B &= c_{xu}^k (c_{zt}^k - c_{zv}^k) + c_{xv}^k (c_{zu}^k - c_{zt}^k) + c_{xt}^k (c_{zv}^k - c_{zu}^k) \\
C &= c_{yu}^k (c_{xt}^k - c_{xv}^k) + c_{yv}^k (c_{xu}^k - c_{xt}^k) + c_{yt}^k (c_{xv}^k - c_{xu}^k) \\
D &= c_{xu}^k (c_{yt}^k c_{zv}^k - c_{yv}^k c_{zt}^k) + c_{xv}^k (c_{yu}^k c_{zt}^k - c_{yt}^k c_{zu}^k) + c_{xt}^k (c_{yv}^k c_{zu}^k - c_{yu}^k c_{zv}^k)
\end{aligned} \qquad (3).$$

Točke $\mathbf{c}_u^k = \left(c_{xu}^k,\ c_{yu}^k,\ c_{zu}^k\right)$, $\mathbf{c}_v^k = \left(c_{xv}^k,\ c_{yv}^k,\ c_{zv}^k\right)$ in $\mathbf{c}_t^k = \left(c_{xt}^k,\ c_{yt}^k,\ c_{zt}^k\right)$ so točke k-te krivulje prečnega prereza, ki so med seboj najbolj oddaljene. Vrednosti l, h, a in b v enačbi 2 so parametri j-tega linearnega generatorja, ki ga izračunamo z enačbami:

The points $\mathbf{c}_u^k = \left(c_{xu}^k,\ c_{yu}^k,\ c_{zu}^k\right)$, $\mathbf{c}_v^k = \left(c_{xv}^k,\ c_{yv}^k,\ c_{zv}^k\right)$ and $\mathbf{c}_t^k = \left(c_{xt}^k,\ c_{yt}^k,\ c_{zt}^k\right)$ are the three most distant points of the k-th cross-section curve. The values l, h, a, and b in Equation 2 are the parameters of the j-th linear generator, which can be calculated with the following equations:

$$l = \frac{c_{yj}^f - c_{yj}^b}{c_{xj}^f - c_{xj}^b},\quad h = \frac{c_{zj}^f - c_{zj}^b}{c_{xj}^f - c_{xj}^b},\quad a = c_{yj}^b - k c_{xj}^b \ \text{ in/and } \ b = c_{zj}^b - h c_{xj}^b \qquad (4).$$



Sl. 4. *Rekonstrukcija ploskve z uporabo odvojnih trakov*
Fig. 4. *Reconstruction of a surface with developable stripes*

Ustrezno presečišče je treba določiti za vsak linearni generator posebej.

Vektor napake $\mathbf{e} = \left(e_x, \ e_y, \ e_z\right)$, ki določa razliko med prečnim prerezom in izračunanimi presečišči, izračunamo z obrazci:

The intersection point has to be calculated for all linear generators.

The blunder vector $\mathbf{e} = \left(e_x, \ e_y, \ e_z\right)$, representing the difference between the cross-section curve and the intersection points, is calculated using the formulae:

$$e_x = \frac{1}{n}\sum_{j=1}^{n}\left|s_{xj}^k - c_{xj}^k\right|, \quad e_y = \frac{1}{n}\sum_{j=1}^{n}\left|s_{yj}^k - c_{yj}^k\right|, \quad e_z = \frac{1}{n}\sum_{j=1}^{n}\left|s_{zj}^k - c_{zj}^k\right| \qquad (5),$$

kjer je n število točk prečnega prereza. Dane enačbe so preproste in hitre. Odvojni trak je treba zožiti, kakor je prikazano v izpisu 1, če je izpolnjen pogoj 6:

where n is the number of points on the cross-section curve. The formulae are simple and fast. The developable stripe has to be narrowed, as shown in listing 1, if the condition 6 is true:

$$e = \frac{e_x + e_y + e_z}{3} > \varepsilon \qquad (6),$$

kjer je ε tolerančni prag, ki pomeni vrednost v milimetrih, saj so predmeti izraženi v tej merski enoti. Enačbi (5) in (6) sta v izpisu 1 zajeti v klicu funkcije `curveDifference(e, f)`, kjer najprej določimo enačbo sredinske ravnine prečnega prereza med krivuljama $\mathbf{c}^b$ in $\mathbf{c}^f$. Nato izračunamo presečišča med to ravnino in odvojnim trakom (2). Če je razlika znotraj tolerance, izračunamo še presečišča s preostalimi ravninami. Izračun ustavimo takoj, ko je pogoj 6 izpolnjen. Ko moramo odvojni trak zožiti, je treba poiskati novo krivuljo prečnega prereza $\mathbf{c}^w$. Njen indeks w je celo število, ki ga dobimo z izločitvijo decimalnega dela aritmetičnega povprečja indeksov b in f:

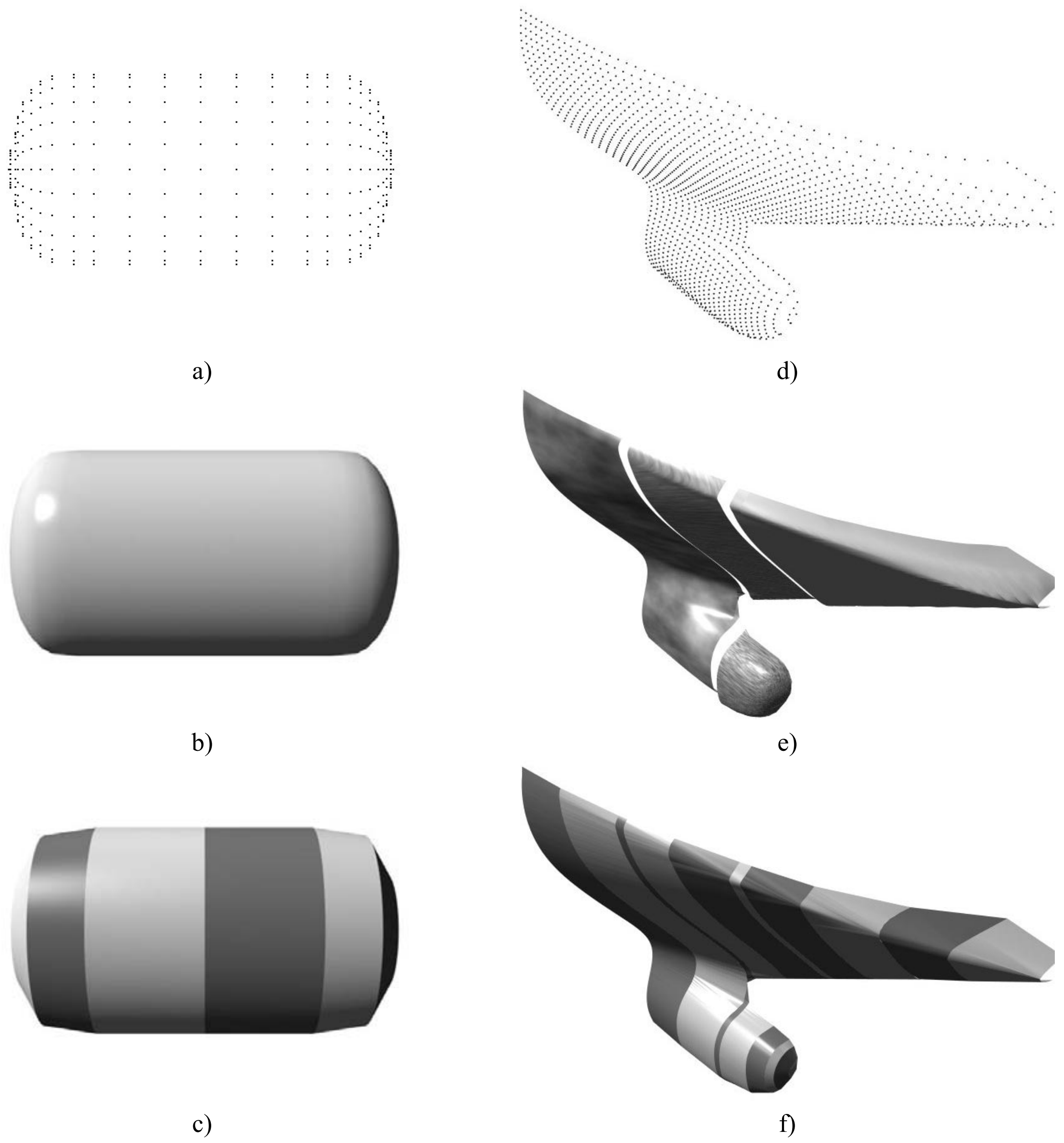
Kot rezultat dobimo dva trakova $\mathbf{s}(\mathbf{c}^b, \mathbf{c}^w)$ in $\mathbf{s}(\mathbf{c}^w, \mathbf{c}^f)$, ki ju ponovno zožimo, če nista znotraj tolerančnega praga. Kakovost približka je odvisna tudi od natančnosti digitalizacije. Če je bila pri digitalizaciji izpuščena kakšna podrobnost, tega ne moremo popraviti.

Rezultate rekonstrukcije ploskve, ki temelji na strategiji "deli in vladaj" ob uporabi odvojnih trakov, lahko vidimo na sliki 5, kjer sta predstavljeni rekonstrukciji hrama za gorivo in polovice ladijskega premca, katerega podatke smo dobili v ladjedelnici Fincantieri v Trstu. Medtem ko je bila rekonstrukcija hrama goriva preprosta, smo morali premec poprej

where ε is a tolerance limit, which is a numerical value measured in millimetres (because our objects are measured in this unit). Equations 5 and 6 are included in the call of a function `curveDifference(e, f)` (see listing 1), where the equation of the middle cross-section plane between the cross sections $\mathbf{c}^b$ and $\mathbf{c}^f$ is determined first. Then the intersection points between that plane and developable stripe are calculated (Eq. 2). If the difference is within the tolerance, the intersection points with the other planes are calculated too. The calculation is stopped when the condition 5 is true. If the developable stripe needs to be narrowed, the new cross-section curve $\mathbf{c}^w$ has to be found. The index w is an integer value, determined by the extraction of the decimal part of the mean value of the indexes b and f:

As a result, two stripes, $\mathbf{s}(\mathbf{c}^b, \mathbf{c}^w)$ and $\mathbf{s}(\mathbf{c}^w, \mathbf{c}^f)$, are generated, which are recursively narrowed if they are not within the tolerance limit. The approximation quality also depends on the digitisation precession. If some details have been missed in the digitisation process, they cannot be corrected here.

The results of the surface reconstruction, based on the divide-and-conquer strategy and the use of the developable stripes, can be seen in Figure 5, where the reconstruction of a fuel tank and a half of a ship's bow, obtained in the shipyard Fincantieri in Trieste, is presented. While the reconstruction of the fuel tank was easy, the reconstruction of the ship's bow was more



Sl. 5. *a) Množica 3D točk, ki določa ploskev hrama za gorivo.*
b) Pričakovana ploskev hrama.
c) Rekonstrukcija ploskve hrama za gorivo (z dolžino 3,6 m), pri tolerančnem pragu ε = 1,50 mm.
d) Množica 3D točk, ki določa obliko polovice ladijskega premca z zaobljenim profilom.
e) Delitev premca, potrebna za odstranitev nezveznosti v prečnih prerezih zaobljenega profila.
f) Rekonstrukcija premca, pri tolerančnem pragu ε = 0,50 mm.
Fig. 5. *a) Set of 3D points defining the surface of the fuel tank*
b) Expected surface of the fuel tank
c) Reconstructed surface of the fuel tank (its length is 3.6m) with the tolerance limit ε = 1.50 mm
d) Set of 3D points defining the shape of the half of the bow
e) The division of the bow necessary to eliminate the discontinuity in the cross-sections of the bulb
f) Reconstruction of the bow with the tolerance limit ε = 0.50 mm

razdeliti na štiri dele (sl. 5e), da smo odstranili nezveznost v prečnih prerezih zaobljenega profila. Vse štiri dele smo ločeno rekonstruirali in potem združili v sliki 5f. Trakove, uporabljene v rekonstrukciji predmeta, je treba še odviti v ravnino. Opis tega postopka lahko najdemo v naslednjem razdelku.

complicated. First, the ship's bow had to be divided into four parts (see Figure 5e) in order to remove the discontinuation of the cross-section curves of the bulb. All four parts were reconstructed separately, and then joined in Figure 5f. After the surface reconstruction the stripes had to be flattened. The description of that process can be found in the next section.

3 ODVOJ PLOSKVE V RAVNINO

Odvojni trak sestavljajo 3D štirikotniki, ki povezujejo po dve ustrezni točki med sosednjimi vodili (na primer $\mathbf{p}_1^*$ s $\mathbf{p}_4^*$ in $\mathbf{p}_2^*$ s $\mathbf{p}_3^*$, sl. 6). Njegov odvoj v ravnino je v bistvu preslikava 3D štirikotnikov v ravnino, pri čemer je treba ohraniti dolžine robov. Ta preslikava mora biti pri tem natančna in hitra. Zaradi tega ustrezen štirikotnik raje skonstruiramo v ravnini, kakor da bi izvirni 3D štirikotnik postavili v ravnino s pomočjo zasukov. Naj bodo dana oglišča 3D štirikotnika $\mathbf{p}_1^*$, $\mathbf{p}_2^*$, $\mathbf{p}_3^*$ in $\mathbf{p}_4^*$, ki je v splošnem neravninski. Iz razdalj med oglišči določimo dolžine njegovih stranic: d_1, d_2, d_3 in d_4. Neravninski štirikotnik moramo skonstruirati v ravnini z uporabo dveh trikotnikov, ki ju dobimo z eno od diagonal štirikotnika. Pri tem ni pomembno, katero od diagonal izberemo. Naj prvi trikotnik določajo točke $\mathbf{p}_1^*$, $\mathbf{p}_2^*$ in $\mathbf{p}_4^*$, drugi trikotnik pa točke $\mathbf{p}_2^*$, $\mathbf{p}_3^*$ in $\mathbf{p}_4^*$. Če želimo drugi trikotnik postaviti v ravnino prvega, moramo izračunati dolžino skupne stranice, ki jo označimo z d_5 (ta pomeni razdaljo med točkama $\mathbf{p}_2^*$ in $\mathbf{p}_4^*$). Točke, ki določajo ravninski štirikotnik, označimo s $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ in $\mathbf{p}_4$.

Najprej izračunamo vrednosti notranjih kotov ravninskega štirikotnika α, β in γ z naslednjimi enačbami (sl. 6a):

3 UNROLLING THE SURFACE ONTO THE PLANE

The developable stripe is composed of 3D quadrangles, which connect at two points of neighbouring directrices (for example $\mathbf{p}_1^*$ with $\mathbf{p}_4^*$ and $\mathbf{p}_2^*$ with $\mathbf{p}_3^*$, see Figure 6). The unrolling of the developable surface is, in fact, a mapping of the 3D quadrangles onto the plane, where the edge distances are preserved. The mapping process must be fast and accurate. Therefore, we construct the quadrangles in the plane instead of rotating the original 3D quadrangles onto the plane. Let $\mathbf{p}_1^*$, $\mathbf{p}_2^*$, $\mathbf{p}_3^*$ and $\mathbf{p}_4^*$ be vertices of a general nonplanar 3D quadrangle. The lengths of the edges (d_1, d_2, d_3, and d_4) are determined by the distances between the vertices. The nonplanar quadrangle has to be constructed in the plane with the help of two triangles that we get with the help of one of its diagonals. It is not important which diagonal is chosen. Let us take the first triangle be determined by the points $\mathbf{p}_1^*$, $\mathbf{p}_2^*$ and $\mathbf{p}_4^*$, and the second one by the points $\mathbf{p}_2^*$, $\mathbf{p}_3^*$ and $\mathbf{p}_4^*$. If we want to set the second triangle in the plane of the first one, we have to calculate the distance of their common edge, denoted by d_5 (which is the distance between the points $\mathbf{p}_2^*$ and $\mathbf{p}_4^*$). The points, determined on the planar quadrangle, are denoted by $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$.

First we calculate the values of the inner angles α, β, and γ of a planar triangle using the following equations (see Figure 6a):

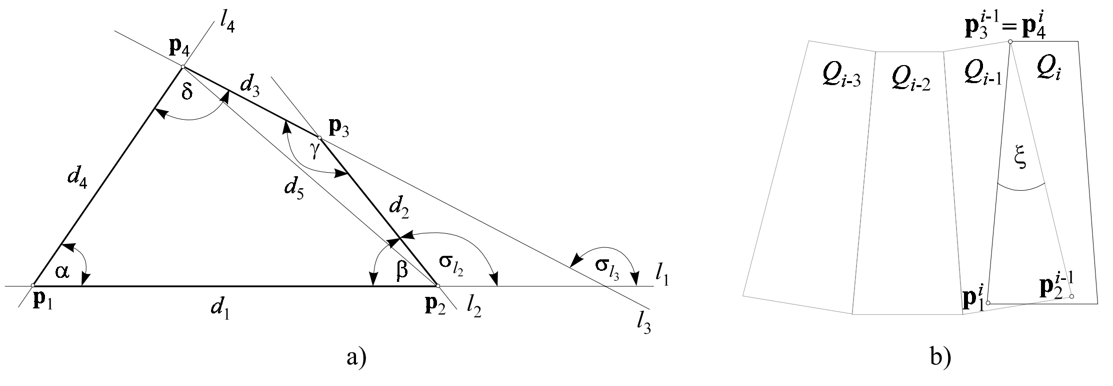
$$\alpha = 2\arctan\left(\frac{r_1}{s_1 - d_5}\right),\ \beta = 2\left(\arctan\left(\frac{r_1}{s_1 - d_4}\right) + \arctan\left(\frac{r_2}{s_2 - d_3}\right)\right),\ \gamma = 2\arctan\left(\frac{r_2}{s_2 - d_5}\right) \quad (7),$$

kjer so / where

$$s_1 = \frac{d_4 + d_1 + d_5}{2},\ r_1 = \sqrt{\frac{(s_1 - d_4)(s_1 - d_1)(s_1 - d_5)}{s_1}},\ s_2 = \frac{d_5 + d_2 + d_3}{2} \ \text{in / and}\ r_2 = \sqrt{\frac{(s_2 - d_5)(s_2 - d_2)(s_2 - d_3)}{s_2}} \quad (8).$$

Brez izgube splošnosti lahko skonstruiramo ravninski štirikotnik $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4$ v ravnini *xy*. Njegovo konstrukcijo začnemo z določitvijo lege enega od njegovih oglišč, na primer oglišča $\mathbf{p}_1$. Nato izberemo eno izmed dveh stranic, s katerima je povezano

Without any loss of generality we can construct the quadrangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4$ in the *xy* plane. We start its construction by defining the position for one of the vertices, for example, the vertex $\mathbf{p}_1$. Then we select one of two edges that are connected to the



Sl. 6. *a) Konstrukcija splošnega ravninskega štirikotnika*
b) Ohranjanje odvisnosti sosednosti pri preslikavi štirikotnikov
Fig. 6. *a) Construction of an arbitrary planar quadrangle*
b) Preserving the neighbouring relation by quadrangle mapping

izbrano oglišče $\mathbf{p}_1$. Naj bo to stranica $\mathbf{p}_1\mathbf{p}_2$, ki leži na premici l_1. Zaradi enostavnosti računanja postavimo, da je premica l_1 vodoravna. Lego oglišča $\mathbf{p}_2$ tako izračunamo kot (x_1+d_1, y_1). Izračun lege oglišča $\mathbf{p}_3$ je nekoliko zahtevnejši, saj je treba ohraniti vrednost notranjega kota β. Na sliki 5a lahko vidimo, da ležita oglišči $\mathbf{p}_2$ in $\mathbf{p}_3$ na skupni premici l_2. Premica l_2 je določena z lego oglišča $\mathbf{p}_2$ in vrednostjo kota σ_{l_2}, ki je zunanji kot h kotu $\beta(\sigma_{l_2} = \pi - \beta)$. Lego oglišča $\mathbf{p}_3$, ki je od oglišča $\mathbf{p}_2$ oddaljena za razdaljo d_2, lahko izračunamo z naslednjima enačbama:

vertex $\mathbf{p}_1$. Let it be the edge $\mathbf{p}_1\mathbf{p}_2$ that lies on the line l_1. To reduce the computational complexity, we define that the line l_1 is horizontal. The position of the vertex $\mathbf{p}_2$ is (x_1+d_1, y_1). The calculation of the position of the vertex $\mathbf{p}_3$ is not so easy, since the angle β has to be preserved. In Figure 5a it can be seen that the vertices $\mathbf{p}_2$ and $\mathbf{p}_3$ lie on the same line l_2. The line l_2 is determined by the position of the vertex $\mathbf{p}_2$ and the angle σ_{l_2}, which is the supplementary angle to the angle $\beta(\sigma_{l_2} = \pi - \beta)$. The position of the vertex $\mathbf{p}_3$, whose distance to vertex $\mathbf{p}_2$ is d_2, can be calculated using the following equations:

$$x_3 = \frac{-(2kN - 2x_2 - 2ky_2) \pm \sqrt{(2kN - 2x_2 - 2ky_2)^2 - 4(1+k^2)(x_2^2 + y_2^2 - d_2^2 + N^2 - 2Ny_2)}}{2+2k^2} \qquad (9),$$

$$y_3 = kx_3 + N$$

kjer sta $k = \tan(\sigma_{l_2})$ in $N = y_2 - kx_2$. Enačbi (9) določata dve možni rešitvi. Prava je tista, ki določa pozitivno usmeritev štirikotnika.

Nazadnje je treba določiti še lego oglišča $\mathbf{p}_4$. Ker leži oglišče $\mathbf{p}_4$ na premici l_3, moramo poprej določiti vrednost kota σ_{l_3}. To vrednost lahko izračunamo z enačbo:

where $k = \tan(\sigma_{l_2})$ and $N = y_2 - kx_2$. Equations 9 define two possible solutions. We take the vertex that defines the positive orientation of the quadrangle.

Finally, the position of the vertex $\mathbf{p}_4$ has to be calculated. Since the vertex $\mathbf{p}_4$ lies on the line l_3, we have to determine the angle value σ_{l_3} first. This can be calculated using the following equation:

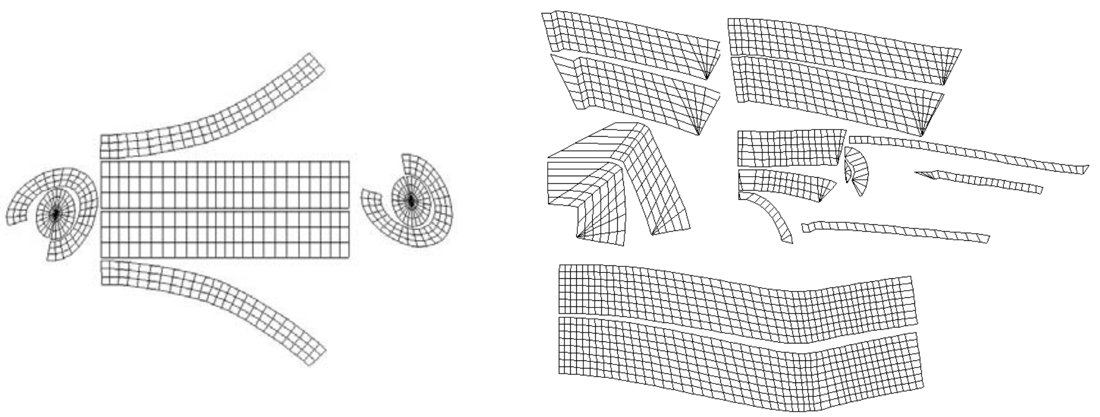
$$\sigma_{l_3} = 2\pi - \beta - \gamma \qquad (10).$$

Lego oglišča $\mathbf{p}_4$ lahko določimo z enačbo (9), kjer sta $k = \tan(\sigma_{l_3})$ in $N = y_3 - kx_3$, namesto x_2 in y_2 pa vzamemo koordinati x_3 in y_3 oglišča $\mathbf{p}_3$, namesto d_2 pa d_3.

Odvoj štirikotnika je končan, ko vzpostavimo odvisnost sosednosti do predhodno odvitih štirikotnikov. Označimo oglišča tekočega štirikotnika Q_i s $\mathbf{p}_j^i$ (j = 1, 2, 3, 4). Štirikotnik Q_i ima stranico $\mathbf{p}_1^i\mathbf{p}_4^i$ enako stranici $\mathbf{p}_2^{i-1}\mathbf{p}_3^{i-1}$ sosednjega štirikotnika Q_{i-1}. Zato moramo najprej premakniti skonstruiran štirikotnik Q_i v takšno lego, da se točki $\mathbf{p}_3^{i-1}$ in $\mathbf{p}_4^i$ ujemata. Nato ga

The position of the vertex $\mathbf{p}_4$ can be determined by Eq. 9, where $k = \tan(\sigma_{l_3})$, $N = y_3 - kx_3$, and instead of the parameters x_2 and y_2 the coordinate values x_3 in y_3 of the vertex $\mathbf{p}_3$ are used. Instead of d_2 the distance d_3 is used.

The flattening of a quadrangle is finished after the neighbouring relation to a previously flattened quadrangle is established. Let us denote the vertices of the running quadrangle Q_i by $\mathbf{p}_j^i$ (j = 1, 2, 3, 4). The edge $\mathbf{p}_1^i\mathbf{p}_4^i$ of the quadrangle Q_i and $\mathbf{p}_2^{i-1}\mathbf{p}_3^{i-1}$ of its neighbour Q_{i-1} are the same. Therefore, we have to translate the quadrangle Q_i into the position where the vertices $\mathbf{p}_3^{i-1}$ and $\mathbf{p}_4^i$ coincide. Second, the generated quadrangle Q_i has to be rotated, by



a) b)

Sl. 7. *a) Rezultat odvoja hrama za gorivo*
b) Rezultat odvoja polovice ladijskega premca
Fig. 7. *a) Result of flattening the fuel tank*
b) Result of flattening one half of the ship's bow

zavrtimo za kot ξ v lego, da se zgoraj omenjeni stranici ujemata (sl. 6b). Vrednost kota ξ računamo podobno kot vrednost kota α (7). Rezultata odvoja ploskev pri uporabi opisanega algoritma sta prikazana na sliki 7.

Testni ploskvi smo rekonstruirali z različnima vrednostima tolerančnih pragov. Pri rekonstrukciji hrama je bila vrednost tolerančnega praga enaka ε = 1,50 mm, pri rekonstrukciji ladijskega premca je bila ta vrednost enaka ε = 0,50 mm. Prvo ploskev smo približali z osmimi odvojnimi trakovi, drugo pa s šestnajstimi. Hram goriva sestavlja 577 točk in polovico ladijskega premca 2550 točk. Za rekonstrukcijo in odvoj hrama z gorivom smo potrebovali 0,120s, za rekonstrukcijo in odvoj polovice ladijskega premca pa 0,260s. Čeprav so druge metode za odvoj ploskev po delih opisanemu algoritmu idejno blizu, jih neposredno z našim algoritmom ne moremo primerjati, saj so vezane na točno določene tipe predstavitve ploskev. Algoritem smo testirali na osebnem računalniku Athlon 900 MHz.

the angle with value ξ, into the position where the above-mentioned edges coincide (Figure 6b). The angle value ξ is calculated in the same way as the angle value α (Eq. 7). The results of surface flattening, using the described algorithm, can be seen in Figure 7.

The test surfaces were reconstructed with a different value of the tolerance limit, which was ε = 1.50 mm for the fuel tank and ε = 0.50 mm for the ship's bow. The first surface was approximated with 8 developable stripes and the surface of the ship's bow was approximated with 16 stripes. The fuel tank consists of 577 points and one half of the ship's bow consists of 2550 points. We needed 0.120s for the reconstruction and the flattening of the fuel tank and 0.260s for the reconstruction and the flattening of one half of the ship's bow. Although there are other methods for per partes surface flattening based on the same idea, we cannot compare them with our algorithm directly, since they are bound to particular surface representation types. The algorithm was tested on an Athlon 900 MHz personal computer.

4 SKLEP

V prispevku smo predstavili novo metodo za rekonstrukcijo in odvoj digitaliziranih krivulj, ki temelji na strategiji "deli in vladaj" ob uporabi odvojnih trakov za približek ploskve. Metoda je hitra, ni pa primerna za rekonstrukcijo predmetov z odprtinami. Metodo smo testirali na dejanskih podatkih ladje, pri kateri smo nezveznost v delu prečnih prerezov premca odpravili tako, da smo le-tega razdelili na štiri dele, ki smo jih nato ločeno rekonstruirali in nato odvili. Za uporabo te metode v ladjedelništvu bi morali še dodatno razviti metodo za določanje oblike jeklenih plošč glede na odvoj in obliko lupine. S tem bi lahko v veliki meri avtomatizirali postopek konstrukcije ladijskih trupov, s čimer bi lahko prihranili precej časa. Metodo bi bilo treba še dodatno prilagoditi za odvoj lupin, izdelanih iz kompozitnih materialov. Čeprav je ostalo še veliko dela, pomeni opisana metoda dober temelj za prihodnje delo.

4 CONCLUSION

In this paper we have presented a new method for reconstructing and flattening digitised surfaces that is based on the divide-and-conquer strategy using the approximation of the surface with developable stripes. The method is fast, but it cannot be used for the reconstruction of objects with holes. We have tested the method on the real data of a ship, where the discontinuation of part of the cross-section curves was eliminated by dividing the bow into four parts that were reconstructed and then flattened separately. To use this method in the ship-building industry, a method for a determining the shape of steel plates, according to the skin flattening and the hull form, has to be developed. This could automate a large part the hull-construction process, which could save a lot of time. Additionally, the method could be adopted to flatten the hulls of ships made of composite materials. Although there is much work to be done, the described method represents a good basis for future work.

Zahvala

Za sodelovanje se zahvaljujemo Upravi Fincantieri v Trstu (Italija), posebej pa še ing. Marcu Amabilinu.

Acknowledgements

The authors would like to thank Direction of Fincantieri in Trieste (Italy) and especially ing. Marco Amabilino.

5 LITERATURA
5 REFERENCES

[1] Faux, I. D., M. J. Pratt (1981) Computational geometry for design and manufacture. *Chichester: Ellis Harwood*.
[2] Azariadis, P., N. Asparagathos (1997) Design of plane developments of doubly curved surfaces. *Computer - Aided Design*, Vol. 29, No. 10, 675 – 685.

[3] Parida, L., S. P. Mudur (1993) Constraint - satisfying planar development of complex surfaces. *Computer - Aided Design*, Vol. 25, No. 4, 225 – 232.

[4] McCarthney, J., B. K. Hinds, and B. L. Seow (1999) The flattening of triangulated surfaces incorporating darts and gussets. *Computer - Aided Design*, Vol. 31, No. 4, 249 – 260.

[5] Aono M., D. E. Breen and M. J. Wozny (1994) Fitting a woven-cloth model to a curved surface: mapping algorithm. *Computer – Aided Design*, Vol. 26, No. 4, 278 – 292.

[6] Kolmanič, S., N. Guid, (2000) The flattening of arbitrary surfaces by approximation with developable stripes. *Proceedings of Seventh IFIP WG 5.2 workshop on geometric modeling: fundamentals and applications*, 43-52.

[7] Elber, G. (1995) Model fabrication using surface layout projection. *Computer - Aided Design*, Vol. 27, No. 4, 283 – 291.

[8] Hoschek, J. (1998) Approximation of surface of revolution by developable surfaces. *Computer - Aided Design*. Vol. 30, No. 10, 757-763.

[9] Forgach, K. M. (1997) Resistance and design propeller powering experiments for the mid-term sealift ships represented by models 5501PR and 5502FP, *David Taylor Model Basin, Report CRDKNSWC/HD-0207-08*.

[10] Wyatt D. C., P. A. Chang (1994) Development and assessment of a total resistance optimized bow for the AE36, *Marine Technology* 31, 149-160.

[11] Fjalstrom, P. O. (1993) Evaluation of a Delaunay-based methods for surface approximation. *Computer-Aided Design*, Vol. 25, No. 11, 711 – 719.

[12] Fong, P., H.P. Seidel (1993) An implementation of triangular B-spline surfaces over arbitrary triangulations. *Computer - Aided Geometric Design*, Vol. 10, 267 – 275.

[13] Hoppe, H. et al., (1992) Surface reconstruction from unorganised points. *ACM SIGGRAPH Computer Graphichs*, Vol. 26, No. 2, 71 – 78.

[14] Hoppe, H. et al., (1994) Piecewise smooth surface reconstruction. *Proceedings of SIGGRAPH'94*, *ACM SIGGRAPH*, 295 – 302.

[15] Oblonšek, Č., N. Guid, (1998) A fast surface-based procedure for object reconstruction from 3D scattered points. *Comput. Vis .& Image Underst.*, Vol. 69, No. 2, 185-195.

[16] Calladine, C. R. (1984) Gausssian curvature and shell structure. *Proceedings of the conference Mathematics of Surfaces*, 179 – 196.

[17] Gurunathan, B. and S. G. Dhande (1987) Algorithms for development of certain slasses of ruled surfaces. *Computers & Graphics*, Vol. 11, No. 2, 105-112.

Naslov avtorjev: mag. Simon Kolmanič
prof.dr. Nikola Guid
Univerza v Mariboru
Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova 17
2000 Maribor
simon.kolmanic@uni-mb.si
guid@uni-mb.si

Authors' Address: Mag. Simon Kolmanič
Prof.Dr. Nikola Guid
University of Maribor
Faculty of Electrical Engineering
and Computer Science
Smetanova 17
2000 Maribor, Slovenia
simon.kolmanic@uni-mb.si
guid@uni-mb.si

Prejeto:
Received: 14.12.2001

Sprejeto:
Accepted: 29.5.2003

Odprt za diskusijo: 1 leto
Open for discussion: 1 year