Surface Fairing for Ship Hull Design Application

P. Brunet A. Vinacua
M. Vivó N. Pla
A. Rodríguez

Report LSI–98–6–R

Surface Fairing for Ship Hull Design Application

N.Pla, A.Vinacua, M.Vivo, P.Brunet
Univ Politecnica de Catalunya, Barcelona
A. Rodriguez
Sener, Madrid, Spain

The paper describes the design and implementation of a fairing module for a ship hull design application. This module is part of a large project on ship design, including facilities for piping, hull design and virtual reality tools for interactive navigation. The surface hull model is based on trimmed NURBS. A discussion of existing fairing algorithms and related criteria for this particular application is presented. The proposed algorithm, based on filtering parallel plane sections of the hull is presented and discussed.

1. Introduction

In ship building the fairing of the ship hull surface is of great relevance, as it has an important impact in the ship's performance. Because of this, the ship industry has always spent considerable efforts in achieving smooth, adequate surfaces for the hull. Traditionally, this process was done by craftsmen who would bend thin wood sticks in the shape of sections of the projected hull by hanging weighs, and then study the tension along the stick to decide if the shape was good enough. When the design of hulls moved to be computer-based, methods akin to this evolved, where a large family of sections were drawn by the computer orthogonally projected onto the screen (see Figure 1),

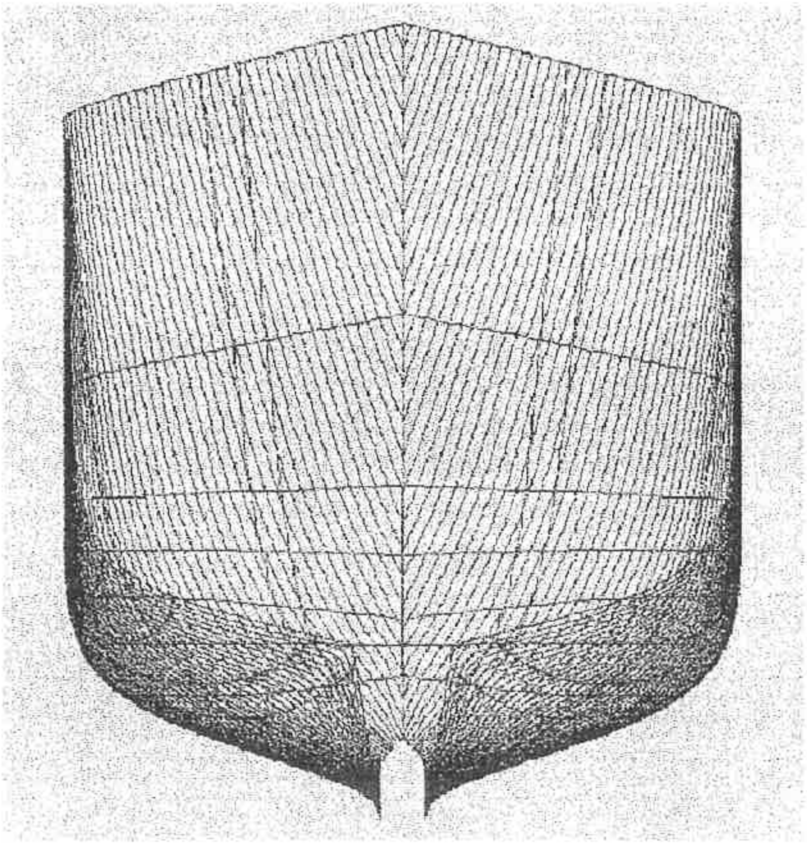

Figure 1: A standard sections plot.

where a specialized technician would look for artifacts or indications of bad shape (in the uneven spacing of these cures) and painfully edit those curves to improve the shape. The process would still take a well trained operator many weeks to complete.

Nowadays the ship hull's shape is generally modeled with NURBS, and surface-based interrogation techniques are thus available and not uncommon: Gaussian curvature plots, Mean curvature plots, sections with arbitrary families of planes,...Some fairing mechanisms proposed in the literature deal with these surfaces as such, and have therefore an advantage over the traditional method and the potential to reach a solution much faster. However, they do not meet yet all the user criteria for such procedures, and have therefore not yet been adopted throughout.

In this paper we discuss what are these user requirements, and propose new methods mid-way between the traditional curve-based methods and the abstract surface-based methods, in that they operate over a surface, but interact with it through sets of curves.

In the next section we enumerate the user requirements on ship hull fairing algorithms, and on section 3 we review some of the previous work in the area. Section 4 introduces the new method we propose and present its results.

2. User requirements for hull fairing

The method for fairing a ship hull depends on the CAD product used to define the hull. The best way to reach a closed description of a ship hull is to use a three dimensional surface modeling method. In our case, the hull form design software is based on trimmed NURBS surfaces. Thus, the fairing method here has to deal with this model.

Most of the ship computer applications define the hull form by two sets of grid lines [KCh93], thus the fairing method used in those cases can not be applied to our ship hull model.

The ship fairing method is required to be an automatic version of the traditional shipyard fairing method. The traditional method is performed manually by a very skilled operator who checks the discontinuities by inspection of the drawings generated by planar sections. In this case, many weeks and sometimes months must be spent in order to obtain a fair model. The planar sections that are usually used are the waterlines, the (transversal) sections or other defining set of curves. The desired fairing method must allow the user to use the planar sections to achieve the faired hull.

Another requirement of the fairing method is the use of local tolerances. The surface obtained by application of the fairing process has to be close to the original surface. In fact, a tolerance parameter will be given by the user, which controls the maximum distance between the faired surface and the original one at some discrete points.

The fourth requirement is that the fairing has to be global. Most of the existing fairing methods deal with a single patch, but the desired method has to fair the whole ship hull shape.

Finally, besides having a fair surface which approximates the original ship shape, some constraints have also to be satisfied by the faired surface, which are the main features of the ship: length overall, length between perpendiculars and water lines among others. Because of this reason, the fairing method has to keep a number of constrains given by the user.

3. Discussion of existing fairing algorithms

Ship hull fairing goes back almost to the beginning of the CAGD literature. Nowacki and Reese discussed it in [NoR83], where they discuss the whole process of designing the hull shape from initial naval requirements, through fairing of its shape, and includes references on automatic fairing dating back to 1967. In that paper they propose to express the shape of the hull to be faired as a collection of patches, to be chosen as large as possible for best results. They consider the boundary curves and fair them independently, and then fill in the holes with new patches. Appropriate twists have to be chosen at the corners to be successful. They propose the development of criteria of fairness, and put forward the basic ones: thin plate energy and its approximations.

Moreton and Sequin, in [MoS95], characterize several fairing metrics for curves and surfaces. Basic metrics include minimum energy curves (MEC) and surfaces (MES), together with minimum variation curves (MVC) and surfaces (MVS). It is concluded that minimizing the variation of the curvature (MVC, MVS) leads to smoother and more pleasant surfaces. The authors present a MVS-based surface design scheme in which a minimum variation network (MVN) of curves is first computed. The MVN interpolates the data points and inherits the topology of its connecting graph. The MVN is consistent with a unique second fundamental form in each data point, the final shape being improved when G^2 continuity is imposed in curve segments arriving the vertex from opposite directions. In a second step, a MVS surface is obtained from the MVN network by an optimization through a constrained conjugate gradient algorithm. The proposed scheme is computationally very expensive but leads to visually good results.

Cheng and Barsky, in [ChB91], propose a method for constrained fairing of curves. Constraints are introduced in the form of interpolation of points with small tolerances. By using a coordinate separation, the problem can be reduced to functional interpolation of values with predefined tolerances. The authors propose a minimization of the MEC functional with constraints. After some manipulations, the problem is reduced to a constrained quadratic optimization problem, that is solved using standard quadratic programming algorithms.

Taubin ([Tau95]) proposes a fairing algorithm for discrete surfaces based on the iterative application of a low-pass filter. The initial discrete surface consists of a set of points with triangular topology and neighbor information for every vertex —edge connected vertices. The filter is local, two-step, and guarantees a non shrinkage of the discrete surface. It is based on a Laplacian operator that operates on the neighbor vertices of the filtered point. The algorithm generates good faired surfaces and presents linear time and space complexity.

On the other hand, Welch and Witkin [WeW94] present a variational approach for solving the same fairing problem for discrete surfaces. They use a local parameterization based on a planar approximation of the triangles converging to the visited vertex, and a local parametric representation of the surface. The variational functional is a discrete approximation of the minimum energy MES criterion, which is optimized using a constrained conjugate gradient optimization. The surface topology is recomputed after every iteration by means of a constrained triangulation. The behavior of the overall algorithm is acceptable if the initial surface is close enough to the final one.

J. Roulier, T. Rando([RR95]) present a study of different fairness metrics for the design of curves and surfaces. From this study, an algorithm to design fair curves or surfaces is deduced. The main contribution of this paper is that several fairness metrics are defined to measure the different levels of fairness. Given a curve or a surface, a fairness metric is defined as the length of the *derived* curve or the area of the *derived* surface. The derived curve or surface depends only on a geometric invariant of the original curve or surface. Several derived curves (or surfaces) are presented. Once the fairness metric μ has been chosen the algorithm can be described as follows:

> Determine the free variables for the problem at hand, and express μ as a function of these free variables.
>
> Choose a method for the evaluation of the function μ .
>
> Choose a minimization algorithm and suitable starting values for the algorithm to minimize μ, and perform the minimization using the evaluation method chosen in the previous step.

Finally a discussion about the method to evaluate the length or the area is presented, as well as about the selection of a non linear optimization code to use.

Sapidis and Farin [SF90] present a fairing algorithm for B-Spline curves. The algorithm has the following properties: is automatic, local and preserves the local convexity properties of the initial curve.

The method proposed by the authors is based on removing and reinserting knots of the spline. The point at which the spline curve has to be faired is identified automatically.

The fairness of the curve is measured by the continuity of the curvature plot. Thus, the curve is faired at the knot corresponding to the largest discontinuity of the derivate of k.

The Eck and Jaspert paper, [EJ95], is related to [SF'90]. The aim of the authors was to deduce an automatic and local fairing process according to [SF'90] for given planar or spatial point sets, which is based on direct manipulation of the data. The fairness criteria used is similar to the one used in [SF'90]. The fairing process is iterative, fairing only one point or inner vertex in each iteration. The new position of the modified point is in a neighborhood of the original point, moreover it is situated on a straight line defined previously. This is a heavy constrain for the fairing process, but makes the computation of new position of the point more easy and effective. Discrete curvature and discrete torsion are used to define the line. The obtained algorithm has to the following properties: the faired point in within a given tolerance form the original point and a global fairness criterion is also decreased. In the planar case, the algorithm also guarantees shape preservation.

Greiner discussed in [Gre94] the optimization involved in all variational schemes. He shows how the intrinsic surface's geometry can be used to find good quadratic approximations to functionals used in fairing, thus making their minimization inexpensive. This paper is of great relevance in the context of variational design, as it turns many theoretical approaches into computationally feasible approaches for real world applications. Greiner also mentions the problem of tolerances in variational design, although his approach is a lot less formal and complete than the rest of the paper, essentially truncating the minimization perturbation to be within tolerances, which in no way assures that we obtain the minimum under the tolerance constraints.

This results were applied by Kolb, Pottmann and Seidel in [KPS95] to the problem of surface reconstruction. There they start with a polyhedron and construct smooth interpolants to the vertices. They do so by first finding a curve network through the vertices which is smooth and optimizing the global shape. Although not a paper on fairing or on ship hull design, the method is relevant also to the design of fair shapes in general, and can be applied in this context too.

Bloor and Wilson [BlW94] propose a method for the design of fair surfaces based on the solution of a biharmonic partial differential equation. A closed, parametric solution of the PDE is computed by imposing boundary conditions on the positions and tangents. The solution is evaluated on a rectangular grid (u, v) for visualization purposes. PDE surfaces are proposed by the same authors [BWH95] for the fairing of an initial continuous surface $P(u, v)$. In this case, the variational approach minimizes a functional including the distance to the initial surface P and an integral term depending on squared derivatives. A discrete solution on a rectangular grid is obtained by iteratively applying a local filter to the initial surface. The filter can be parametrized in order to remove high frequencies in P.

Vassilev, [Va95], presents a method for fair spline interpolation and approximation for curves and surfaces which can be applied to large amounts of data. The algorithm chooses a parameterization of the curve, and the curve is determined by using interpolation or by a least squares fit. To increase the fairness, additional data points are used as degrees of freedom. The Additional Data Points (ADP) are inserted only where a lack of data is observed. The fairness criterion for curves is the energy functional proposed by Celniker and Gossard. A new energy functional is considered in the case of surfaces. Some ADP are inserted to the given data and the minimization of the fairness measure gives a linear system to solve, using least squares in the approximation case.

Sapidis and Koras , [SK97], present some results about the analysis of a graphical plot to get information about the fairness of a curve, although they don't propose new methods for curve fairing. Usually a plot of a mathematical measure, as the curvature, reveals the fairness of a curve. In this paper, it is proved that a non uniformly scaled curvature plot often includes sharp corners at knots where the discontinuity in the derivative if the curvature is relatively small.

The authors propose to apply these results in expert-system like procedures to use curvature plots correctly by a stylist.

A short summary of the discussed algorithms is presented in Table 1. The main conclusion is that there is a lack of fairing algorithms for continuous surfaces allowing geometric constraints and tolerances in the location of certain interpolation points. Moreover, there is no explicit use of the planar sections of the surface. The conclusion is that specific algorithms for ship fairing that take into account the user requirements must be developed. In this context, a new ship design oriented fairing algorithm has been derived. The algorithm is presented and discussed in the next two sections.

	Curves or Surf.	Discrete or Cont.	Tolerances Yes/No	Local or Global
[NoR83]	S	C	N	G
[SF90]	C	C	partial	L
[WeW94]	S	D	N	G
[ChB91]	C	C	Y	G
[Gre94]	S	C	Y-	G
[Tau95]	S	D	N	L
[EJ95]	C	D	N	L
[RR95]	C,S	C	N	G
[MoS95]	S	C	N	L
[KPS95]	S	C	N	G
[BlW94]	S	C	N	G
[Va95]	C,S	D	N	G

Table 1: Summary of fairing papers discussed

4. The proposed fairing algorithm

A specific fairing algorithm to fulfill ship design application requirements has been designed and implemented. The algorithm refines the internal NURBS surface model of the hull while visualizing it through the standard set of planar sections. User interaction on any of these sections is also allowed during the smoothing process.

The main diagram of the proposed algorithm is shown in figure 2.

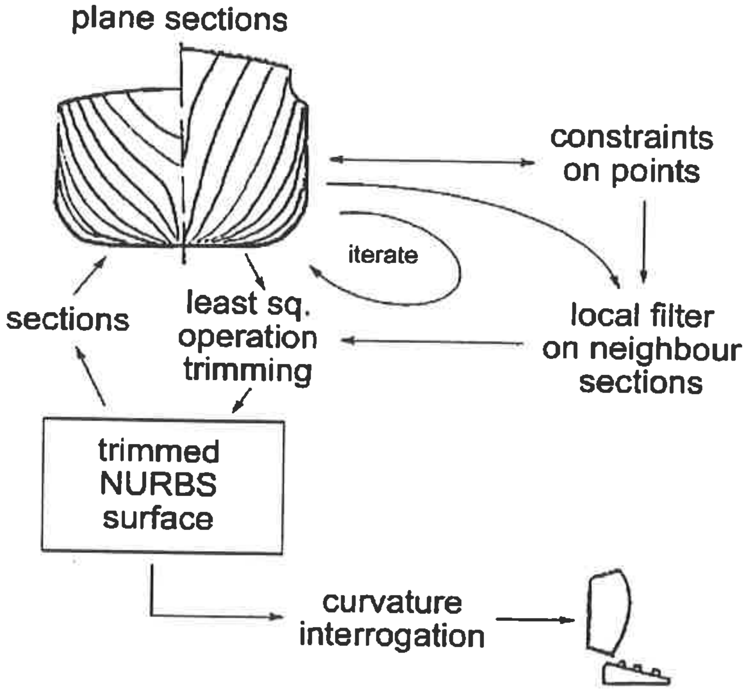

Figure 2:The proposed algorithm.

In a first step, the set of planar sections is automatically computed from the internal trimmed NURBS surface model. A modification of the algorithm [KPW92] is used for this purpose. The hull surface model can also be used for surface quality interrogation (curvature plots and arbitrary direction sets of sections). Then, the fairing algorithm is based on a local filter that works on neighbor curve sections, in a way that reproduces the classical ship fairing scheme. Finally, after section curves have been faired, a new modified hull surface model is obtained using a least squares approximation. This procedure is repeated, and a new set of planar sections is computed and faired until the ship hull is sufficiently smooth.

The fairing algorithm works as follows,

```
While NonSmooth (Surface) do
    C := ComputePlaneSections (Surface)
    For iter in [1..MaxIter] do
        For Every plane section C[i] in C do
            F[i] := Filter ( C[i], C )
        EndFor
        k := ComputeMaximumBias ( F, C )
        C[k] := C[k] + $\lambda$ * ( F[k] - C[k] )
    EndFor
    Surface := SurfaceModification ( Surface, C )
EndWhile
```

The algorithm iterates until the surface is sufficiently smooth regarding the user requirements. Every iteration starts by computing the set C of planar sections $C[i]$ of the internal NURBS surface model. This set C is computed by the procedure ComputePlanarSections, following [KPW92]. Then, the inner loop computes a filtered section $F[i]$ for every plane section $C[i]$ in C. In the present implementation we use a polygonal representation for the sections $C[i]$ and the filtered section $F[i]$ is computed as the piecewise linear curve being equidistant from $C[i-1]$ and $C[i+1]$, except for the first and last sections where $F[1]=C[1]$ and $F[N]=C[N]$. At this moment, the Euclidean norm of the distances from the vertices of every $F[i]$ to the corresponding polygonal section $C[i]$ can be used to compute a measure of the maximum discrepancy between C and F: the maximum discrepancy occurs at the least faired contour $C[k]$, where,

$$Norm(F_k - C_k) \geq Norm(F_j - C_j) \qquad \forall\ j$$

Being C_k the least faired contour, it is chosen as the contour to be smoothed in the present iteration. The smoothing operation works by modifying C_k and morphing it to F_k. The interpolation value λ is a positive, real parameter not greater than one. It is computed in order to guarantee that constraints are fulfilled: every fairing iteration tries to uniformly move C_k towards the filtered target F_k while the set of fairing constraints are satisfied.

The algorithm repeats the identification and fairing of the least faired contour a number of times (MaxIter) before computing the new NURBS surface model. The surface computation is performed by the SurfaceModification routine, that obtains the new hull surface model from the new polygonal contours through an SVD algorithm similar to [HuS'97]. The parameter MaxIter tunes the approximation of the final surface: A small MaxIter value results in a better approximation in the SurfaceModification procedure but also slows the overall fairing algorithm, as the number of calls to ComputePlaneSections will increase accordingly. On the other hand, a greater value for MaxIter will have the reverse effect.

The algorithm has been implemented and tested in real ship surfaces. Some results are presented and discussed in the next section.

5. Results

Figure 3 shows results of applying the proposed algorithm to a sample surface. The original surface has been perturbed with random noise below the tolerance given to the algorithm (1% of the surface's scale). Figure 3 a) depicts the Gauss curvature plot of the surface with noise, whereas

Figure 3 b) shows the Gauss curvature plot after smoothing.

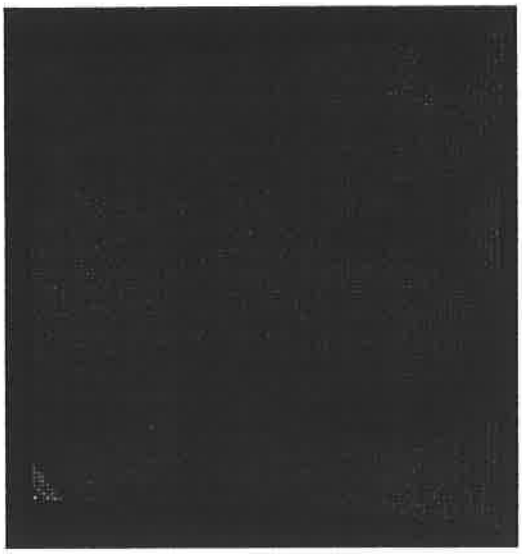

Figure 3: a) Original surface.

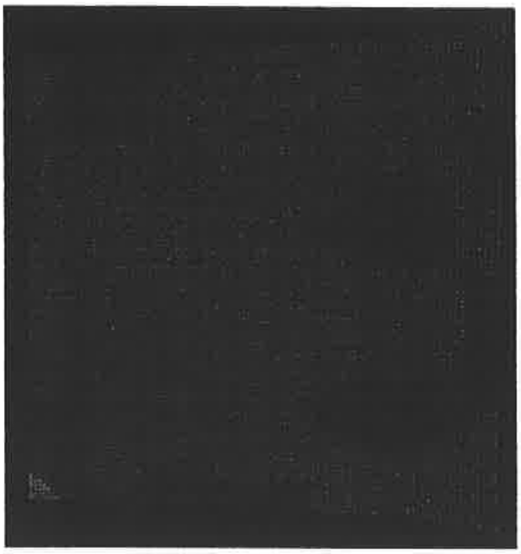

Figure3: b)Faired surface.

The algorithm manages to remove or improve conflictive areas while guaranteeing that the modifications to the surface remain under the given tolerance. The algorithm is insensitive to the original patch structure of the surface, and deals with it globally, although in doing so through families of sections it is more amenable to ship designers, who can likely interact with the inner aspects of the algorithm more confortably.

6. Acknowledgements

This research has been partially supported by grant TIC-95-0630-C05-03 of the CICYT and by contract CTT-C2191. The third author has been supported by grant 1996FI-3100PG of the *Generalitat de Catalunya.*

7. References

[BlW94] Bloor,M.I.G., and Wilson,M.J., Interactive Design Using Partial Differential Equations, in "Designing Fair Curves and Surfaces", Nick Sapidis Ed., SIAM, 1994, pp 231-251.

[BWH95] Bloor,M.I.G, Wilson,M.J., and Hagen,H., The Smoothing Properties of Variational Schemes for Surface Design, Computer-Aided Geometric Design Vol 12, 1995, pp 381-394.

[ChB91] Cheng,F., Barsky,B.A., Interproximation: Interpolation and Approximation Using Cubic Spline Curves, Computer-Aided Design, Vol 23, 10, 1991, pp 700-706.

[EJ95] Eck, M., Jaspert,R., Automatic fairing of Point Sets, in Designing Fair Curves and Surfaces, N.Sapidis Ed., SIAM, 1994, pp.

[Gre94] Greiner, G., Variational Design and Fairing of Spline Surfaces, Computer Graphics Forum Vol 13, 3, 1994, pp C-143–C-154.

[HuS97] Hu, Y-P., Sun, T-C., Moving a B-Spline Surface to a Curve - A Trimmed Surface Matching Algorithm, Computer-Aided Design Vol 29, 6, 1997, pp 449-455.

[KCh93] J.S. Kouh, S.W. Chau, Computer-aided geometric design and panel generation for hull forms based on rational cubic Bézier curves. Computer Aided Geometric Design V. 10, (1993).

[KPS95] Kolb, A., Pottman, H., Seidel, H-P., Fair Surface Reconstruction Using Quadratic Functionals, Computer Graphics Forum Vol 14, 3, 1995, pp C-469–C-479.

[KPW92] Kriezis,G.A., Patrikalakis,N.M., Wolter,F.E., Differential Equation Methods for Surface Intersections, Computer-Aided Design Vol 24, 1, 1992, pp 41-55.

[MoS95] Moreton,H.P., Sequin,C.H., Minimum variation Curves and Surfaces for Computer-Aided Geometric Design, in Designing Fair Curves and Surfaces, N.Sapidis Ed., SIAM, 1994, pp 123-159.

[NoR83] Nowacki, H., Reese, D., Design and Fairing of Ship Surfaces, in “Surfaces in CAGD”, Barnhill and Boehm (eds.), North-Holland, Amsterdam, 1983, pp 121–134.

[RR95] Roulier,J., T. Rando,T., Measures of Fairness for Curves and Surfaces, in Designing Fair Curves and Surfaces, N.Sapidis Ed., SIAM, 1994, pp

[SF90] Sapidis,N., Farin, G. Automatic fairing algorithm for B-Spline curves. Computer Aided Design V. 22, N. 2, (1990).

[SK97] Sapidis, G. Koras Visualization of curvature plots and evaluation of fairness: an analysis of the effect of scaling. Computer Aided Geometric Design V. 14, (1997).

[Tau95] Taubin,G., A Signal Processing Approach to Fair Surface Design, ACM Computer Graphics, Siggraph 1995, pp 351-358.

[Va95] Vassilev,T.I., Fair interpolation and approximation of B-splines by energy minimization and points insertion. Computer Aided Design, V. 28. N. 9. pp 753–760.

[WeW94] Welch,W., Witkin,A., Free-Form Shape Design Using Triangulated Surfaces, ACM Computer Graphics, Siggraph 1994, pp 247-256.

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1998

LSI–98–1–R "Optimal Sampling Strategies in Quicksort and Quickselect", Conrado Martinez, Salvador Roura.

LSI–98–2–R "Query, PACS and simple-PAC Learning", J. Castro and D. Guijarro.

LSI–98–3–R "Interval Analysis Applied to Constraint Feasibility in Geometric Constraint Solving", R. Joan-Arinyo and N. Mata.

LSI–98–4–R "BayesProfile: application of Bayesian Networks to website user tracking", Ramón Sangüesa and Ulises Cortés.

LSI–98–5–R "Some reflections on applying Workflow Technology to Software Process", Camilo Ocampo and Pere Botella.

LSI–98–6–R "Surface Fairing for Ship Hull Design Application", P. Brunet, A. Vinacua, M. Vivó, N. Pla, A. Rodríguez.

LSI–98–7–R "Trust Values for Agent Selection in Multiagent Systems", Karmelo Urzelai.

LSI–98–8–R "The use of SAREL to control the correspondence between Specification Documents", Núria Castell and Àngels Hernández.

LSI–98–9–R "Intervalizing colored graphs is NP-complete for caterpillars with hair length 2", C. Àlvarez, J. Diaz and M. Serna.

LSI–98–10–R "A unified approach to natural language treatment", Jordi Àlvarez.

LSI–98–11–R "Collision Detection: Models and Algorithms", Marta Franquesa and Pere Brunet.

LSI–98–12–R "Height-relaxed AVL rebalancing: A unified, fine-grained approach to concurrent dictionaries", Luc Bougé, Joaquim Gabarró, Xavier Messeguer and Nicolas Schabanel.

LSI–98–13–R "HyperChromatic trees: a fine-grained approach to distributed algorithms on RedBlack trees", Xavier Messeguer and Borja Valles.

LSI–98–14–R "Asynchronous Interface Specification, Analysis and Synthesis", Michael Kishinevsky, Jordi Cortadella, Alex Kondratyev and Luciano Lavagno.

LSI–98–15–R "Heuristics for the MinLA Problem: Some Theoretical and Empirical Considerations", Josep Diaz, Jordi Petit i Silvestre and Paul Spirakis.

Hardcopies of reports can be ordered from:

Núria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6
Jordi Girona Salgado, 1–3
08034 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, http://www-lsi.upc.es/