



HAL
open science

Processing free form objects within a Product Development Process framework

Bianca Falcidieno, Franca Giannini, Jean-Claude Léon, Jean-Philippe Pernot

► **To cite this version:**

Bianca Falcidieno, Franca Giannini, Jean-Claude Léon, Jean-Philippe Pernot. Processing free form objects within a Product Development Process framework. John Michopoulos and David Rosen and Chris Paredis and Judy Vance. Advances in Computers and Information in Engineering Research, ASME, 2014. hal-00922950v1

HAL Id: hal-00922950

<https://inria.hal.science/hal-00922950v1>

Submitted on 31 Dec 2013 (v1), last revised 25 Nov 2016 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contents

1	Processing free form objects within a Product Development Process framework	3
	<i>Bianca Falcidieno, Franca Giannini, Jean-Claude Léon and Jean-Philippe Pernot</i>	
1.1	Introduction	3
1.2	PDP and free form surfaces in product aesthetic and engineering designs	5
1.3	Modeling and sketching free form surfaces	10
1.4	Towards advanced models, methods and tools for modeling free-form surfaces	18



1

Processing free form objects within a Product Development Process framework

Bianca Falcidieno

IMATI-GE CNR, Genova, Italy

Franca Giannini

IMATI-GE CNR, Genova, Italy

Jean-Claude Léon

Grenoble University - INRIA, Grenoble, France

Jean-Philippe Pernot

Arts et Métiers Paris Tech, Aix-en-Provence, France

1.1	Introduction	3
1.2	PDP and free form surfaces in product aesthetic and engineering designs	5
1.3	Modeling and sketching free form surfaces	10
1.4	Towards advanced models, methods and tools for modeling free-form surfaces	18
	References	23

1.1 Introduction

Creativity is the generation of new ideas, by proposing new ways of looking at existing problems, seeing new opportunities or by exploiting emerging technologies or changes in markets. Innovation is the successful exploitation of new ideas. Design is what links creativity and innovation.

Creative industries are based on these three processes. Combined with Information and Communication Technologies (ICT) they are an emerging lead market in the European knowledge economy and a dynamic motor for economic and social innovation, outlining a common vision of future ICT-driven creative industries, where the fusion of creativity and technology is considered a crucial factor of success.

We are assisting to a huge impact of ICT on several sectors of these industries: music composition and production; film, television and video; animation and computer games; writing, publishing and print media; advertising and marketing; architecture and visual arts; product design and manufacturing.

Activities in all these sectors require a high level of technical knowledge to create and manipulate content. The computer is now the workbench for making digital content, and creators need to feel at home with digital technology.

In most of these contexts the advances of technology pave the way to new ICT services, where dynamically evolving applications are continuously adapting to changing users needs and to user-generated content. This paradigm is supporting inventive and creative practices in the arts, science,

engineering, education, and business by enabling entirely new types of creative digital productions. It pushes advances in user-centric content creation systems and the convergence of roles of professional and non-professional users, who become the major content producers, whose focus is not only on the technologies but, even more, on the applications where users are working.

Recently, the content sector has experienced a massive transformation due to changes in user behavior, showing a growing desire to participate actively in the creation process, stimulated by new ICT capabilities. This is particularly true in fields like entertainment, digital media and visual arts, where a product or a response will be judged as creative to the extent that it is both a novel and appropriate, useful, correct or valuable response to the task in hand, and the task is heuristic rather than algorithmic [7]. Also in product design, new technologies such as low cost scanners or fast prototyping give the impression that everybody can become a creative person allowing the creation of his/her own objects. However, besides those trends promising prosperous connections between creative sectors and ICT tools, several obstacles prevent this vision from turning into reality. Designing is a cyclic, iterative, loop of phases or steps, a form of systematic problem solving, finding a solution and integrating the procedural aspects of design with the structural aspects of design problems. Design is also a heuristic interactive process that can be adapted to the particular requirements of a task [18]. It is therefore crucial that computer-aided tools adopted during industrial product development can support users in dealing with these paradigms. In particular, in designing industrial products to capture and render emotion and intuition in the aesthetic of shapes is crucial for the success to the market. Industrial design includes the specification of all those products that are somehow subject to visual judgment and appreciation. It ranges from transport vehicles, not only cars but also boats, buses, trains . . . , to home appliances, furniture, cosmetic containers and thousands of other objects that we deal with in our daily life. Therefore, it is not surprising that the industrial design process is not unique and may vary deeply from company to company, depending on several factors such as the complexity of the product to be designed, the available financial resources, the equipment used, the team of specialists involved and so on. For simple manufactured objects a single designer can represent the design team, but for complex products a team of specialists is usually required. In the latter case, interaction with the engineering team, which is in charge of guaranteeing the fulfillment of planned functionality and performances together with product feasibility, is much more complex. It is clear that the more technologically complex a product, the more problems will arise when the stylist's idea is coupled with engineering and ergonomic constraints. Existing interfaces and tools, though functional and very powerful, appear cumbersome and daunting to a stylist; creating a complex model still requires considerable technological knowledge and an unnecessary effort in specifying unknown details, which are not necessary in the early phases of design and limit freedom and freshness in conception.

To a larger extent, this is true when designing objects using free-form surfaces, where other functional issues are prominent rather than aesthetics, and it is impossible to precisely specify a shape at the beginning due to both an uncertainty of the idea itself (not fully refined at the conceptual phase) and the lack of information not yet known about engineering conditions and simulation results. The consequence is that the process requires too many back and forth attempts before obtaining acceptable outcomes, thus resulting still far from the stylist's needs of intuitive systems that improve interactivity without reducing their creativity.

The purpose of the chapter is then to review: 1) the stages of a PDP (Product Development Process) where free-form shapes appear and are subjected to aesthetic and functional requirements; 2) the modeling, sketching and modification activities illustrating how free-form surfaces can be processed and what are the corresponding difficulties faced during these tasks; 3) the contributions of our community to solve some of these issues and the problems which are still open.

PDPs have to incorporate a large amount of product requirements involving many actors, tasks and activities (CAD, CAE, CAM, optimization processes, reverse engineering, . . .). These processes often rely on DMUs (Digital Mock-Ups) integrating multi-representations and multi-resolutions of components to shape complex components and products containing notably the needed free-form

surfaces. Free-form surfaces and free-form shapes appeared long ago in CAD/CAS systems based on fairly sophisticated modeling functions. Little progress has been obtained since then because free-form shapes are still obtained tediously even if reverse engineering processes have become rather common practices. There are still several challenging limitations and difficulties that need to be addressed to improve the complete definition of products incorporating free form shapes, ranging from the initial model creation to its final assessment.

Problems and solutions related to this topic are presented in the rest of the chapter: section 1.2 deals with PDP and free-form surfaces in product (aesthetic and engineering) design; section 1.3 focuses on modeling and sketching free-form surfaces; section 1.4 explores advanced models, methods and tools for modeling and processing free-form surfaces; conclusive remarks and open issues are outlined in the last section.

1.2 PDP and free form surfaces in product aesthetic and engineering designs

1.2.1 PDP structure

PDPs, as experienced by companies, have not a unique structure but vary in accordance with the products they deliver, their structure and their complexity. The development of domestic appliances significantly differs from that of an aircraft because the complexity of the latter, i.e. its number of components, its number of functions, the diversity of physical phenomena it is subjected to, is much greater than the former. This can strongly influence their interaction with free-form surfaces processing whenever this category of surface is relevant for a given product. Still, a generic structure of a PDP can be derived as a reference scheme that can be adapted to specific companies and classes of products. Anyhow, it is assumed that a digital model of the product stands as a reference product model. Figure 1.1 summarizes the corresponding structure of a reference PDP. In Figure 1.1, the proposed task decomposition of the PDP has to be tuned for each component of the product, i.e. the reverse engineering task may be applicable to a few components only or not at all. Similarly, the re-engineering task performed using a preexisting product can address a sub-assembly of the product or no component at all. Also, tasks like preliminary design may nearly vanish when products are simple. Referring again to product complexity, structuring a PDP is more and more critical because complexity is increasing. Often better products require incorporating more functions, e.g. the evolution from initial cell phones to smart phones, hence increasing their number of components. Also, better products require a more accurate evaluation of their behavior in a context performance optimization is becoming increasingly important. Consequently, behavior optimization covers shape optimization, which refines shapes, increasing their complexity in the sense of number of modeling features required to describe them. Optimization processes call for more structure in a PDP to tightly connect simulation and design updates when component shapes are evolving.

The task breakdown used here is classical and it is compatible with concurrent engineering approaches when considering that components are not uniformly moving from preliminary design to embodiment design and so on. This means that some sub-assemblies or components of a product are not at the same design stage than others at a given time of a PDP. For example, a car dashboard can be at the embodiment design stage whereas the internal areas of a car body trunk, defined by a set of sheet metal components, can be at the level of detailed design.

Regarding products incorporating components with free-form surfaces¹, several observations can be made depending on the type of product:

¹other than blending radii

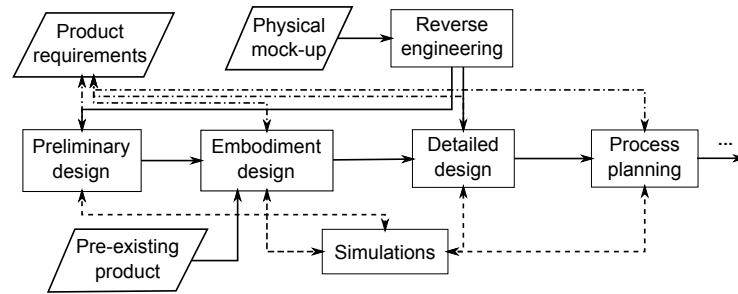


FIGURE 1.1: Reference structure of a PDP.

- As a first example, regarding an outer car body, preliminary design is still based on hand sketches to shape the car without the use of a digital model, which comes later on in the PDP at a stage that can be rooted at the end of the preliminary design stage;
- Because of the above observation, reverse engineering is a typical process free-form shape components are subjected to. This can be highlighted again in the context of car design where clay models are the origin of a reverse engineering phase leading to a digital model of a car at the preliminary design phase;
- The ‘pre-existing’ product item in Figure 1.1 expresses configurations often designated as product re-engineering where a company reconsider design and manufacture solutions to better fit the current economic context. There, assuming that the pre-existing product owns a digital representation, the modification processes are central to the re-engineering task. If CAD, CAx, softwares are ascendant compatible over minor revisions, they are not over major ones and over long term periods. Consequently, model parameterization and construction trees are frequently not available in these settings;
- 3D printing is a process not inserted in Figure 1.1 though it is becoming more and more frequent as an evaluation means of design solutions, often at preliminary design stages but not only.

The previous observations already show some variants influencing PDP pipelines, showing how PDP pipelines can be derived from the reference one proposed in Figure 1.1 to match practices in different companies.

The reference structure of PDPs incorporates also one important source of input data: the product requirements, and one important process: the simulation that can take place at different levels of PDPs. The product requirements refer to the functional and aesthetics specifications that lead to criteria for the evaluation of design variants and the selection of the one that performs best when the product is in use. AS illustrated on Figure 1.1, requirements can contribute to various stages of a PDP from preliminary design to process planning. Also, PDP stages anterior to a given one can set requirements on the subsequent stages. Taking again the example of car body design, the outer car body shape is defined at the preliminary design stage and sets dimensional requirements regarding doors, headlights, . . . , the inner body structure that will be designed at later stages of a PDP.

Similarly, the simulation process designates a generic activity that takes place all along a PDP and refers to different categories of simulations. Classically, simulations refer to behavioral evaluation of components. Digital behaviors have been extensively developed and cover now a wide range of configurations and phenomena. From structural behavior, acoustics, thermal behavior, fluid flow, . . . , to user-centered simulations, i.e. simulations using virtual reality technology to immerse a user in a virtual world with the digital product and interact with it, their diversity is still increasing as

well as their number in order to reduce the number of prototypes and speed up PDPs. Regarding the above observations, the 3D printing process can be regarded as a particular type of simulation where objectives can be to assess qualitatively the visual effect of free-form shapes, their size in a real environment, their color, . . . The higher the complexity of a product, the greater the number of simulations tend to be to obtain a designed product that behaves correctly right from the first product occurrence manufactured.

Simulations can interact both ways with the different stages of design, it is not only a process derived from design stages with components' or assemblies' geometric models but it is also processes that can feed the design stages with updated components' or assemblies' geometry when the simulation turns into an optimization process where 3D shapes can be part of the design parameters of an optimization process.

Overall, a PDP cannot be reduced to a simple process flow extracted from Figure 1.1. Even if a main objective is to get a product right at the first time, addressing one or several PDP processes at a finer resolution, some requirements may not be achieved right with the first design. Consequently, shape modifications, dimensions adjustments are common practices that must be supported by the digital models of products and their components. These notions of modifications and adjustments are tightly related to the concept of compromise that is taking place all along the various processes of a PDP.

As mentioned at the beginning of this section, there is no unique scheme for a PDP, which impacts the digital models that take part to this PDP. The diversity in PDP schemes triggers another one about the digital models. If B-Rep CAD models are commonly regarded as product reference data, other digital shape representations get an increasing interest, e.g. finite element or finite volume models for simulations. Other digital shape models like point sets, images, take part to reverse engineering processes. This diversity of digital representations is a key to setting up a PDP tailored to a given product, which enforces the diversity of scenarios that can be derived from the reference process structure in Figure 1.1.

The various PDP processes generate or modify not only shape models but also engineering data, e.g. component names, material designation and mechanical properties, stress fields, functions, . . . Some of these data can be tightly related to shape models, i.e. to geometric information. Here, considering geometric information as a reference, these data are collectively designated as semantic information. Most often, semantic and geometric data are loosely connected and their connection is not robust, i.e. semantic data are assigned interactively to a whole component rather than the appropriate geometric entities or some of its sub domains but it is error prone and tedious. Producing and maintaining tighter connections between is also a key to an improved efficiency of PDP processes.

1.2.2 Free-form shape and PDP relationships

First of all, it has to be observed that the focus is placed on the digital representations of a component or a product. Digital models exist at most stages of a PDP. The preliminary design phase can be considered as the only one where digital models may not be available because this phase produce only hand sketched representations of parts of a product.

As mentioned in the previous section, hand sketched drawings are frequently used to define free-form shapes such as the external shape of a car, the shape of home appliances, . . . because their shape is hard to define directly as a digital representation using current industrial software whether they are dedicated, i.e. Computer Aided Styling, or not, i.e. Computer Aided Design ones. Figure 1.2 illustrates into which extent hand sketched documents contain uncertainties showing that they cannot be interpreted as sharp product requirements and can hardly be computer processed straightforwardly to produce a 3D model directly from sketches [40]. More precisely, projection lines conveyed by feature points are not converging consistently toward a vanishing point in the perspective view of Figure 1.2a and vanishing points that contribute to the perspective views of Figure 1.2a and b cannot be located accurately.

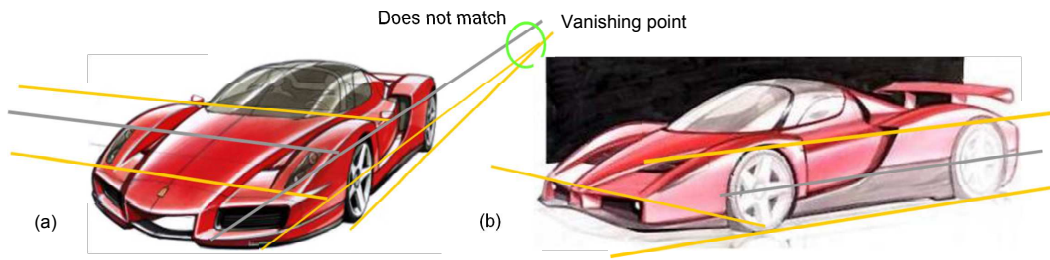


FIGURE 1.2: An example configuration where sketched shape requirements are not accurate enough to be used as geometric prescription. a) an analysis of a sketch showing that lines converging toward a focal point, are inconsistent, other lines cannot define precisely the focal point because they are close to be parallel, b) lines can hardly define the focal point showing that the sketch can contain inaccuracies (courtesy Pininfarina).

Similar observations can be performed about functional requirements where specifications may not be always regarded as consistent between them and dimensional values may not be consistent with material strength prescriptions, for example.

The above configurations reflect the uncertainties due to the need for compromises (see Section 1.2) inside or between PDP processes and these compromises take place throughout a PDP. This is particularly easy to address when requirements can be expressed only qualitatively, e.g. aesthetics, and are regarded as subjective rather than objective like a power produced, a velocity, . . . To express the concept of compromise at the level of digital shape models, it is a means to set requirements on these digital representations. Their capabilities to be easily modified, i.e. add/remove shape details, modify dimensions, apply geometric constraints, . . . , are among these requirements. Because modifying a shape is also comparable to exploring a solution domain, letting know the user that some shape modifications are just not possible is as important as providing solutions. This is particularly important with free-form surfaces where generating a fillet of arbitrary radius between two free-form surfaces has no systematic solution. When seeking a compromise during a PDP process, computing the maximal radius compatible with a configuration of free-form surfaces is as important as looking for the maximal mechanical stresses in a component.

Anyhow, it can be observed that free-form surface models do not benefit from construction tree structures that ease modifications and associate a process to a digital representation of an object (see Figure 1.9.b). Classically, industrial CAD software rely on an incremental modeling paradigm where a complex shape is generated through a sequence of simple shape modeling operations (see Section 1.3.4). Commonly, the chronology of these operations is at the basis of a history tree describing the construction process of an object.

Consequently, free-form shape modifications are generally tedious because operator parameterization and construction tree are often missing, or partly missing and this combines with frequent update failures because free-form surfaces may not exist, e.g. see the above example of existence of a fillet between two free-form surfaces. It is always the case when geometric models are transferred between different software as it is often required to follow the processes contained in a given PDP (see Figure 1.1). Indeed, during a model transfer between CAD and CAE software, whatever the standard format used, i.e. STEP, IGES, . . . , the construction tree of an object is not transferred, hence shape modifications become tedious anyway. To overcome this difficulty, shape modifications based on deformation processes is an approach that has been extensively addressed (see Section 1.3). However, defining a structure that can be an intrinsic history representation of a free-form surface generation is still a means for significantly improving their generation, modifications and shape optimizations.

Regarding components subjected to functional requirements, when free-form surfaces are directly

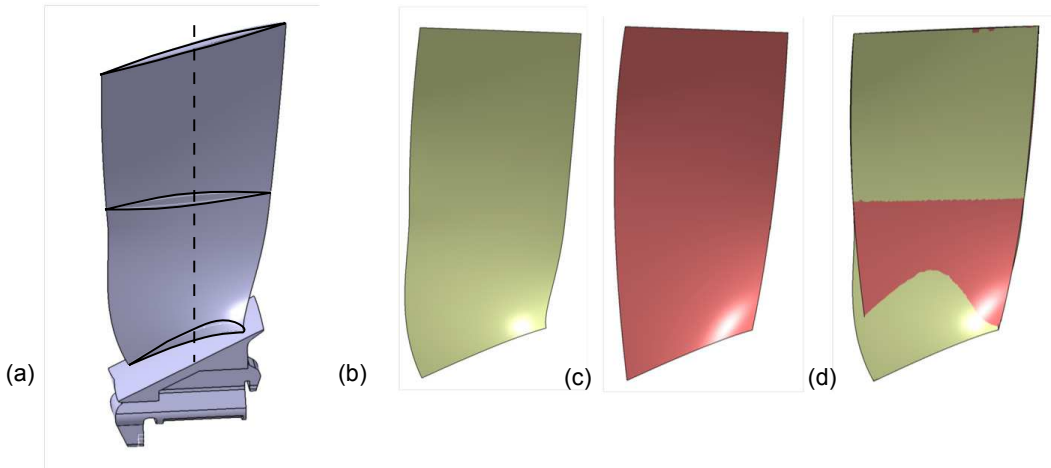


FIGURE 1.3: Examples of a fan blade subjected to functional parameters during a shape optimization process. Influence of shape generation algorithms across different software. a) Initial fan blade model with the three sections used to optimize the blade shape, b) aerodynamic area modeled with CAD software A, c) aerodynamic area modeled with CAD software B and the same input parameters as software A, d) superimposed aerodynamic showing their shape differences.

involved in these requirements, is another category of illustrative configurations involving free-form surfaces processing during a PDP. A characteristic configuration where functional requirements can be exemplified is the definition of fan blades (see Figure 1.3). Apart from the generation process of the aerodynamic area of this component, what illustrates the relationship between functional requirements and shape is its performance optimization. Free-form surfaces contribute to the definition of its aerodynamic part and a construction tree may not be available for the generation of this area but this lack of structure has been already discussed. Here, performance optimization can be reduced to shape optimization acting on the air flow around the blade. Such an optimization process requires automated shape modifications, hence a parameterization. Consequently, the corresponding parameterization and the choice of design parameters input to the optimization process can be regarded as functional parameters and may differ from those used during the initial model generation process. Consequently, the lack of construction tree reduces to configurations that are similar to the ones described above.

At the difference of general purpose shape modifications, parameterization for shape optimization incorporates the use of several models; often geometric models in a CAD environment and finite element (FE) ones or finite volume ones in CAE environments. Such needs identify the geometric element of a model parameterization as reference data that monitor the shape transformations. As such, they could be regarded as intrinsic to the functional requirements and the targeted component, independently of any software as long as similar modeling processes can be applied to generate the object from different software, i.e. CAD or CAE. However, this is subjected to the influence of geometric modeling algorithms and it can lead to fair differences between models generated from the same geometric elements and the same control parameters as shown on Figure 1.3. In Figure 1.3d, the superimposed aerodynamic surfaces show a maximum distance of 3.2% of the total blade height between them. Consequently, shape parameterization related to free-form surfaces in a PDP is essentially valid within one software environment.

As pointed out above, B-Rep NURBS models of free-form surfaces are not the only digital representation of a component during a PDP, piecewise linear, i.e. faceted representations, are also frequently used across a PDP for FE simulations, immersive simulations using virtual reality tech-

niques for usage or maintenance scenarios, . . . In these configurations, component shapes for each category of simulation may differ from each other because simulation hypotheses differ across a PDP. It is often the for FE simulations where detail removal is common task used to take incorporate hypotheses that produce different shapes of the same component. Other simulations derive different shapes and processing free-form surfaces across all the simulations and PDP stages increases significantly the amount of tedious tasks and reduces the efficiency of a PDP. Even though the previous observations don't apply solely to free-form surfaces, the difficulties currently attached to free-form shape processing add to the reduction of efficiencies in PDPs. On a complementary basis, this analysis shows how improvements in free-form shape modeling and processing could improve some tasks and even more globally the efficiency of PDPs.

The diversity of PDP process schemes highlighted at Section 1.2.1 is tightly linked to the diversity of digital shape representations encountered throughout each of them. Over the past years, the concept of B-Rep Nurbs as central representation of a product has progressively evolved into the concept of multiple representations of a component or an assembly across a PDP, i.e. B-Rep Nurbs, Finite Element meshes, faceted representations as used in virtual reality simulations, . . . Currently, these multiple digital shape representations and others like subdivision surfaces, T-splines get also incorporated in modeling processes and increase further the diversity of representations. PDP processes are often centered on one of these representations because of its well suited aptitudes and a component model differs in shape between processes, e.g. a CAD model of a component within the detailed design process is getting simplified and transformed into a mesh when adapted to a finite element simulation. Performing the shape transformations to move a digital shape between two processes of a PDP is still an ongoing research topic awaiting significant progress. Underlying issues are the possible relationships between the geometric models, the possible mapping between geometric models when the shape of components differ between PDP processes.

Section 1.2.1 has highlighted the connection between geometric information and semantic one. Regarding shapes, this connection refers to the concept of feature. Numerous works have addressed feature modeling and feature recognition for mechanical components. However, free-form surfaces embedded in these components has not significantly evolved and up to integration into industrial software. Similarly, setting up a tight connection between semantic information and free-form information is part of scientific issues and robust connections between semantic information and simple geometric entities, i.e. planes, cylinders, . . . is improving slowly developing. Indeed, added semantic information is expanding but its tight connection with geometry is not yet benefiting shape modeling systems, i.e. functional modeling is still part of perspectives of the computer aided tools.

1.3 Modeling and sketching free form surfaces

The use of digital tools for the creation of 3D models is now current practice. Different tools and approaches are now possible thank to the decades of research and efforts in various sectors including Computer Aided Geometric Design, Computational Geometry, signal and data acquisition and processing, electronic engineering. In fact, while at the beginning the only possibility to create a model was through Computer-aided Drafting or Design systems, currently new methods exists for the digital cloning of physical objects. They range from the well-established laser scanning methods to the recent digital photos elaboration means. Each methodology better applies to specific users and exploits different geometric representations. In the following sub-sections a brief overview of the main methods, models and functionalities for shape creating and modification currently existing is provided.

1.3.1 Computer-Aided Design System for Shape generation

The birth of computer supported drafting and modeling system dates at the beginning of the '60s in the last century. The first computer-aided system supporting the creation of digital models is the pioneering Sketchpad defined by Ivan Sutherland in his PhD thesis at MIT [42]. This system provided the possibility to draw 2D objects through the interaction of a light pen with the computer monitor. The interest in 3D digital representation of objects emerged firstly in automotive and aerospace industries, for the production and for the model reuse for such complex products made of a huge amount of recurrent components. de Casteljaou at Citroen and Bézier at Renault defined almost in parallel methods for the definition and manipulation of 3D curves and surfaces which turned out to be equivalent and are still the basis of most of the current commercial CAD systems. The first commercial CAD systems have been launched in the '70s and have had a larger development and distribution starting from the '80s thanks to the availability of UNIX workstations and later on to the capabilities of Personal Computers and laptops allowing the processing and rendering of the models. These systems have been mainly developed having in mind engineers as final users. Therefore the capabilities for defining and changing the object shapes are more reflecting the underlying geometric representation of the surfaces than the natural capabilities and intent of the designers. To try to get closer to the specific user needs, according to a specific product domain, more specialized tools have been defined in successive years. In the field of free form modeling, we assisted to the creation of those referred as Computer-Aided Styling (CAS) or Computer-aided Industrial Design (CAID) systems. They are specifically equipped with sophisticated functionalities for the creation, visual simulation and rendering of 3D surface-based objects. This includes, for instance, the rendering of the reflection of the lights, e.g. the so called zebra strips (see fig. 1.4), and of the environment on the object for analyzing the quality of the surfaces and the effect that object can evoke when inserted in real world. These systems were mainly thought for industrial stylists and were largely used in the automotive sector in the preliminary design phase. Despite their large usage, they were not directly used by stylists but by the so-called surfacers, i.e. persons trained in the use of the systems with the capability of correctly understanding the stylists drawings. In fact designers still preferred the use of pen and paper for expressing their ideas and for achieving their final proposal. Indeed designers are mainly using scribbles to draw the initial ideas which are then modified by successive refinement correcting the scribbles and adding new elements for emphasizing dark and lighted areas to provide shape details and appearance. This is a quite fuzzy and unpredictable process to browse and sharpen up potential solutions to their product ideas. Computer-aided systems weakly support such fuzzy process requiring the insertion of precise information since the beginning. Thus the most applied approaches to create industrial design objects are either the creation of surfaces by exploiting specific sections curves created by tracing the hand-made drawing of the product or by using Reverse Engineering techniques applied to the clay model created from the hand-made sketch [15] (see Figure 1.5).

Moreover the models created with CAID systems very often required a conversion/refinement/re-creation in more engineering oriented systems to provide the correct mathematical representation suitable for the analysis and production activities in the embodiment design phase. Such re-generation in CAD systems was due to various reasons both cultural and technical. In fact, creative users are mainly concerned with the definition of beautiful shapes, not paying attention to the model specification, i.e. the underlying mathematical surface decomposition, thus long and narrow surfaces can be used which may bring problems in later analysis and manufacturing. From the technical point of view CAID systems did not provide engineering analysis and planning tools so a change of system was required with the related problems in data transfer process. To overcome the latest limit, commercial system providers realize integrated environments in which the specialized modules can better communicate, see for instance the CATIA portfolio [2].

Nowadays the main efforts of current 3D surface modeling systems is devoted to better support creative users at large: either in industrial manufacturing sectors and in the new entertainment in-

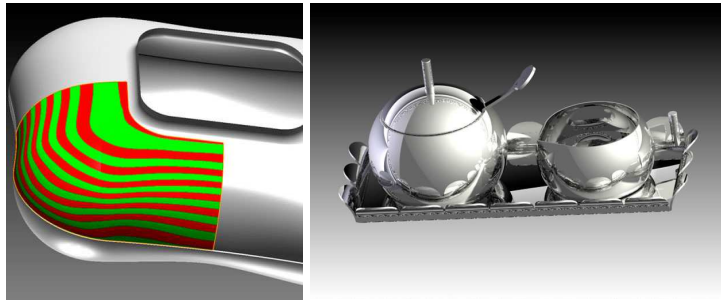


FIGURE 1.4: Example of zebra strips and of rendering (courtesy by Alessi).



FIGURE 1.5: Example of physical prototypes from hand-made sketches (courtesy by Eiger).

dustries, e.g. video games.

Initial solutions are emerging supporting 3D sketching to be used for the creation of the 3D model directly from the sketched elements are emerging. For instance, CATIA Natural Sketch [2] is re-applying Sutherland's approach allowing the designer to directly sketch with a pen on the monitor. Even if more advanced than traditional tools they are still not so immediate in directly achieving the 3D model of any complex shapes. These tools are mainly interpreting the sketched scribbles as potential construction curves for simple surfaces.

Recently, to support designers in browsing new ideas the so-called morphogenesis approach has been inserted in commercial tools, e.g. solidThinking [5], such as trying to give the most efficient design shapes, provides suggestions on possible shapes by proposing alternative shape solutions according specific behavior boundary conditions, e.g. applied forces. This approach is very useful for suggesting unforeseeable solution very organic, i.e. resembling natural elements, but still needs to provide designers with suitable tools for re-adjusting the proposed shapes according to the taste and intent of the designer.

Other example of domain oriented 3D systems for creative users are those developed for the entertainment industry. Initial 3D animation dates in mid '80,s with the very famous short movie Luxo Jr. [4]. Nowadays 3D models are now very used in movies and video games. In addition, being 3D recognized a good mean for communication information and simulation, larger attention has been devoted in the media and communication sectors. Needs in these sectors are a bit different from the industrial engineering domain. Here what is important is the visual appearance and not the precise and detailed shape representation: textures are normally used to complete the visual details very frequently not modeled. Additionally, animation of the produced models can be another important action to be performed of the created models, thus requiring fast transformations. Due to these

characteristics these systems allow also the use of alternative discrete geometric representations in addition to parametric surfaces, i.e. NURBS or B-Splines as adopted in engineering oriented systems, as described in the following subsection.

Despite all these efforts, mainly devoted to providing more user oriented simplified modeling functionalities, these systems are still mainly using traditional modeling operations, such as lofting, sweeping, curve interpolation, filleting and blends. At the same time they are still missing, or at the best weakly supporting, the association of semantic information to the elements of the modeled objects. Such an association is crucial for easy modification, consistency maintenance within and between PDP processes, automation of analysis and planning of activities, model re-use and retrieval.

1.3.2 Digital Shape generation from real objects

An alternative way to create digital objects is through processing of the data acquired from real objects. This approach gained a widespread usage in various disciplines ranging from medicine to archeology, from entertainment to site planning [12, 27, 14]. Even within engineering it has various usage. A typical usage is the creation of the digital representation of products for which it is not available, either because the CAD model is not anymore representing the real object (e.g. changes on the physical object occurred) or it does not exist. This typically happens in maintenance of structures and equipment, and in design where a physical prototype is created either for analyzing the initial idea or for evaluating the achieved solution before the final production. It is also used to assess competitors' products for the understanding of the underlying components, behavior and costs.

The type of technology adopted for real data acquisition strongly depends on the type of objects to be reconstructed and on the planned use. The physical objects can be measured using 3D scanning technologies like CMMs, (Coordinate-Measuring Machines) laser scanners, structured light digitizers, or industrial CT (computed tomography)scanning. For instance for medical purposes as well as for the reconstruction of underground equipment, computed tomography is adopted, which by using X-rays allows the capturing of 3D images of internal elements.

More recently cheap solutions are appearing using technologies from the game industry, e.g. Kinect, or plain photographs from portable cameras, e.g. 123D Catch available on iPad [1] and processed by the Autodesk cloud or 3DSOM Pro [3]. These solutions are still at the embryonic phase and are generally providing rough models in terms of geometry visually enriched with textures corresponding the acquired pictures, thus currently not usable for professional design. Other approaches include the use of calibrated cameras placed in exact positions and then employing photogrammetry methods for reconstruction or the combination of digitalized points and 2D images for the recovery of additional information [33].

In engineering object reconstruction has been traditionally obtained through the use of 3D scanners and it is referred as Reverse Engineering approach. The measured data, usually represented as a point clouds plus additional images, need to be further processed to obtain the watertight 3D CAD representation useful for design and production purposed. Normally the overall process involves the use of intermediate object representation in terms of discretized triangular-faced meshes obtained after various (semi)automatic steps: registration of the acquired data; reduction of the number of points; mesh creation and repairing [10, 47]. The generation of the CAD model from the obtained mesh is a semi-automatic process too. It normally requires several actions from the user in order to obtain the most appropriate decomposing in terms of bounding CAD surfaces. In some cases, i.e. when the surface does not incorporate sharp features for instances, this process can be automatic [20].

1.3.3 Object Representations

As we have seen in the previous sections, depending on the considered Product Development Process phase and on the specific purposes different analysis and tools are applied, each of them is better working on specific object description. Therefore different representation schema are used for the complete project development of new products. As already mentioned, the first representation schema for free form surfaces, named as its developer Bézier, has been defined during the '60s in automotive sector. It exploits matricial calculus. This provides the advantage of handling transformations, such as rotation or translation through simple and efficient operations on matrices. They express the surfaces as a combination of polynomials, also called blending functions, which specify the contribution of the defining control points matrix on each point on the surface. Points on the surfaces can be evaluated at specific values of the two parameters specified in a rectangular domain. Bézier surfaces present an important limit for complex object definition: no local modification capability is provided; that is the modification of any control point has effect on the overall surface. The effect can be mitigated by using a very large number of control points. This is unfeasible because the degree of the surface is directly linked to the number of control points. To overcome this serious limit B-splines and NURBS (Non-Rational Uniform B-Splines) have been introduced [35, 23] which are the de-facto standard in current CAD/CAS systems. They provide a polynomial piecewise definition of the surface which automatically maintain a certain continuity degree conditions among the pieces. In addition, differently from Bézier and B-Splines, NURBS are able to represent also functional surfaces (planes, cylinders, spheres,) thus guarantee the use of a single representation for all the possible geometries. In all these schemes the modification of the surface can be indirectly obtained through the modification of its control points (see Figure 1.6). This can be quite non trivial and requires a certain understanding on how changes of the control points reflect on the surface changes. To this aim methods for more direct shape modifications have been developed and are still under investigation as discussed in the next section of this chapter. Moreover, the definition of complex shaped objects cannot be done in terms of a single parametric quadrangular surface (called patch). Thus the description of complicated shapes needs to be decomposed into several patches, possibly trimmed (i.e. only a part of the parametric domain is considered for the specific object). For example the detailed specification of the car body can require the definition of more than 6000 patches (see Figure 1.7). To represent high quality and globally smooth shapes, these patched have to be smoothly connected bringing to additional continuity constraint conditions which should be preserved during all the shape modification activities, making them more complicated.

Another possibility to representation surfaces is through implicit methods. An implicit surface consists of those points in the 3-D space that satisfy a particular requirement. Thus, it is a function linking the three coordinates of the points: $f(x, y, z) = 0$. The function f can contain any mathematical expression, when it is polynomial it gives rise to the so called algebraic surfaces, which have gained interest in the CAD domain to overcome problems related to specific types of computation, such as intersection, and to data exchange [19, 9]. Nevertheless they have not been widely used in CAD systems because of the difficulty in converting traditional representations in algebraic forms with manageable degree surfaces and of the difficulty in defining suitable means for controlling their shape.

As alternative to the above continuous representations, tessellated models are used in several Product Development Process phases, such as Virtual Reality visualization, rapid prototyping, FE analysis, and are the first model created when creating the digital cloning of physical objects. They allow an explicit, even if approximated, representation normally in terms of triangular planar elements and in this way, even if not unique, they overcome the problem of decomposing the object boundary in suitable patches. They gained large attention in recent years also for modeling purposes due to the increased storage and computational capabilities of computers to treat large data. Thus analysis and modeling capabilities traditionally defined for continuous models are being defined for discrete models as well [11]. The major drawback of such models is related to the missing intuitive

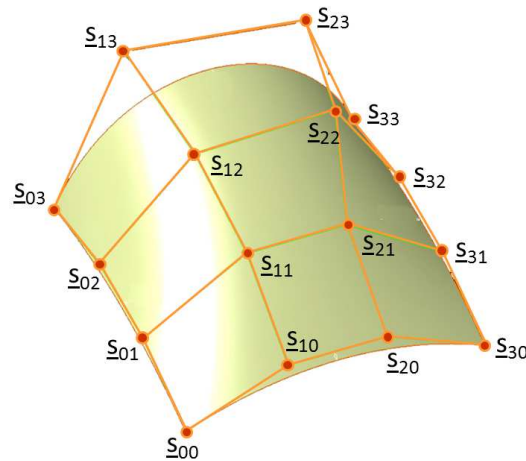


FIGURE 1.6: Example of B-spline surface with its control point network.

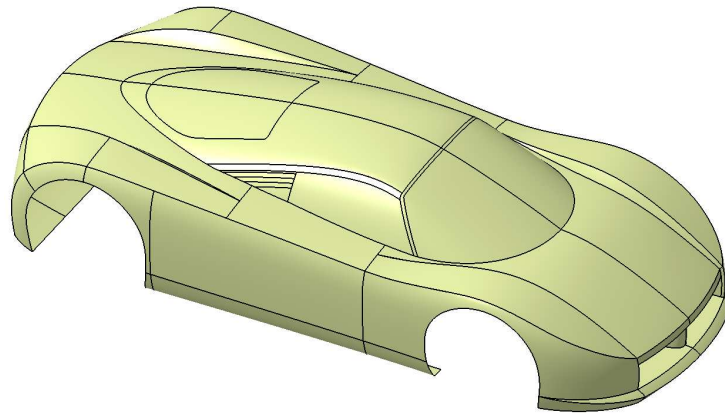


FIGURE 1.7: Example of trimmed B-spline surfaces used to create the car body (courtesy by Fiores II [24]).

shape control parameters which makes difficult the fine control of the shape during its definition and editing.

Other representations that are recently introduced in CAD systems and can be considered in between continuous and discrete representations are sub-division surfaces. Subdivision surfaces initially defined in the '70 have gained major attention in recent times. They can be considered as a generalization of spline surfaces, since they are also controlled by a coarse control mesh, but in contrast to spline surfaces allow to represent surfaces of arbitrary topology. Subdivision surfaces are generated by repeated refinement of control meshes: after each topological refinement step, the positions of the (old and new) vertices are adjusted based on a set of local averaging rules. Figure 1.8 shows an example of different levels of iterative refinements of the subdivision surface control mesh [11].

Subdivision surfaces offer the advantage of representing the object with a single surface, thus there is no need to manage additional topological and continuity conditions as in the case of spline. In contrast, they cannot represent surfaces for which the triangulation is not the result of repeated refinement of a coarse control mesh. These arbitrary surfaces have to be remeshed to subdivision

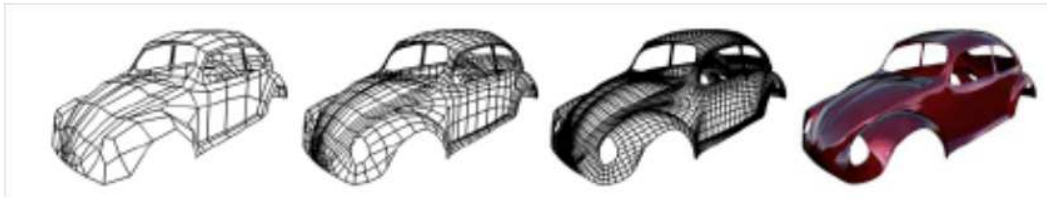


FIGURE 1.8: Subdivision surface generated by iterative refinement of the original control mesh [11].

FIGURE 1.9: Examples of object modeling using: (a) modeling operations and construction tree as available in current CAD modelers, (b) general free-form surfaces where parameterization and construction tree concepts are not available.

connectivity in a preprocessing step. Such a remeshing corresponds to a resampling of the surface, and it usually leads to sampling artefacts and loss of information. Artefacts are unexpected features on the subdivision surface which cannot be intuitively controlled by repositioning the vertices of the original coarse mesh. Additionally they are missing a pipeline of a constructive approach as available for NURBS models, This is mainly due to their insufficient shape predictability and thus difficulty in the control net choice. Moreover, the insertion of G1 discontinuities, i.e. sharp edges, or functional surfaces are difficult to include and handle. Therefore, subdivision surfaces better apply when the main modeling objective is the creation of smoothly appearing objects or when shapes have to be transmitted with low band, as for animation, than in the cases where a strong control on the surface is needed.

An alternative to subdivision surfaces in creating non-rectangular surface and at the same time allowing local modifications without the insertion of additional control points are the so-called T-splines. T-splines are now being inserted in some commercial tools like Rhino [?, 38, 37]. Anyhow several issues are still to be tracked to get the same operational maturity as traditional representations. Other representations, not treated here, are adopted in other processes during the PDP, such as tetrahedral meshes for simulation analyses and voxels for topological optimization. This generates difficulties in the adaptation and linking of the involved representations and resolution to the various processes' need. Associating semantic information to the product shape may facilitate such an integration and adaptation.

1.3.4 Shape modification

Shapes can be created from scratch, or through modifications of existing models since most products are refined versions of existing ones. As seen in the previous section, the final solution of a product is normally achieved through successive shape modifications. It is then clear the importance of having efficient shape modification tools. To support easy editing of shapes, in mechanical engineering the parametric modeling and the concept of feature have been introduced and widely adopted for shape design. Features correspond to recurrent shape primitives that can be instantiated through few intuitive parameters having a predictive shape behavior (when parameter values change) and are providing the way to associate some application semantics, e.g. manufacturing or assembly operation, finishing [39]. Examples of mechanical features are holes, pockets, fillets. To satisfy application and product family design, current CAD systems provide the user capabilities for the specification of feature classes in addition to those already available. This approach is further enriched with the possibility to directly editing object elements (i.e. faces, edges and vertices) to

stretch their shape, through the so-called direct modeling approach.

In mechanical engineering designers benefit from construction tree structures that ease modifications and associate a process to a digital representation of an object (see Figure 1.9.b). Industrial CAD software rely on an incremental modeling paradigm where volume modeling is performed iteratively using planar sketched contours subjected to either extrusion or revolution operations, as generative parameters, and to either material addition or removal, as shape sculpting parameters. The construction tree structure is based on the reference plane containing the sketched contours and primitive shape defined from the generative and shape sculpting parameters. Constructive solid geometry (CSG) fits also in this approach where regularized Boolean operators act as shape sculpting parameters. Parameterized contours and parameterized extrusion lengths or rotation angles are helpful to propagate shape modifications throughout a construction tree to modify an object shape (see Figure 1.9.a).

Indeed, these modeling operations can be all aggregated as gesture based interactions and, more precisely, single trajectory based operations. Effectively, extrusions relate to simple translational movements, revolutions to rotational ones. Even more complex modeling functions like blending radii, that can produce complex surfaces, can be reduced to single trajectory operations when they are decomposed into a connected sequence of edges forming a chain or a single loop. The sequence of edges form the trajectory that can be assigned to these blending operations. Extrusions with draft surfaces or blending with a variable radius are extensions of the previous functions where the initial contour used in the primitive definition is assigned an evolution law, often a simple one. However, the concept of single trajectory is still applicable and the concept of gesture that follows this trajectory, even if it is not the effective user interaction, is also the common concept that aggregate all these modeling operations. When it comes to more general free-form surfaces that are not covered by the previous categories, the concept of gesture based modeling operations tend to vanish and it becomes more difficult to define a construction tree and its corresponding modeling process. Similarly, the diversity of input parameters used by the previous operators have not led to a generic structure of modeling process. This is also due to the intrinsic level of degree freedom in the specification of free form shapes which cannot be characterized by few parameters and rules. Consequently, free-form shape modifications are generally tedious because operator parameterization and construction tree are often missing, or partly missing. Free-form shapes are still defined as a set of free-form surfaces resulting from low-level modeling and modification operations. Approaches for the manipulation of free form surfaces can be classified according to the type of operations and associate control parameters. Thus, two main approaches can be distinguished depending on whether they use:

- operators that do not apply a deformation to an initial shape/surface. This category can be further refined depending on whether they use a set of :
 - numerical and/or geometric parameters (e.g. curves and points) to control classical procedures like sweep, loft, extrusion, intersection, trimming, projection, blending, filleting and so on. These operators can be used for the definition of the overall structure of the objects' shape as well as for local refinements.
 - points to find the surfaces that best fit the points with respect to specific criteria such as the least-squares minimization. This is the method applied in RE approach for the digital model creation.
- operators deforming an initial shape/surface [28, 36]. This category can be further refined depending on whether the deformation is considered as a:
 - global deformation controlled by global parameters. These methods are widely used in animation where there is no need to have a very accurate shaping of the objects.

- local and global deformations through the use of some geometric parameters and constraints like curves and geometric continuity to be preserved within and between the deformed area and the unchanged part. The free form feature approach we developed belongs to this category and is further described in subsection 1.4.2.

This categorization could be further refined depending on the type of underlying surface representation: NURBS surfaces, subdivision surfaces, meshes, voxels and so on. Due to the space limit, we mainly focus on free form surfaces by mean of NURBS curves and surfaces since this type of representation is at the base of the CAD models definition. Furthermore, subcategories could also be defined depending on whether they can also use data and parameters extracted from images [ref]. Even if this is a promising way to reconstruct shapes from real objects, these approaches won't be detailed here [ref]. Finally, the control parameters can refer to classical ones (dimension, coincidence and so on) or to higher level parameters liking aesthetic as well as emotional properties to the underlying geometric models through a set of constraints and dedicated objective functions. These aspects are further developed in the next section.

Despite the above described types of operators, fine shapes adjustments can still require trial and error moving the control points of the underlying surfaces. This means acting on non-intuitive surface parameters linked to the way the surface is defined mathematically but with no mean from the user perspective. This is made more complex when changes are performed on models created by others: the model decomposition in surfaces is not unique, it depends on the designer choices during the model creation which may not reflect the shape interpretation. As a consequence designers cannot express modifications in terms of the wished shape changes but need to be deal with the defining elements and control points of the adopted surface decomposition. Therefore, frequently the user needs to add new control points or surfaces and to perform trimming operations (with the needed continuity conditions specifications) to be able to sufficiently localize the shape changes or to add specific functional surfaces and/or sharp edges. These modifications have the effect of creating more and more complex models, which make propagation of modification a critical issue in free form domain, due to the patch continuity conditions required and to the possible unpredictable effects. Additional difficulties are faced by engineers when geometric operators fail, e.g. no solution to a blend generation, self-intersection when generating a surface through interpolating sections.

In conclusion we can say that despite the big improvements in free form design tools, modifications are still too linked to the underlying geometry than to the shape. Therefore they are generally time consuming and require a lot of expertise in knowing the corresponding effects on the surface of the control point changes. Thus new methods have to be developed to allow the user should to interact with the shape through more intuitive and meaningful elements independent on the underlying patch representation.

1.4 Towards advanced models, methods and tools for modeling free-form surfaces

In the previous sections, various criteria have been introduced to evaluate and compare the numerous free-form surface modeling approaches that have been developed during the last decades. From this analysis, it is clear that the so-called feature-based approaches can play a key role for the modeling of shapes within a PDP framework. Therefore, in this section, we review the recent advances that have tried to extend the feature concept to the free-form domain thus giving rise to several free-form features modeling strategies. More precisely, section 1.4.1 reviews the state-of-the-art of those free-form feature-based approaches and section 1.4.2 presents a clear and detailed analysis of the open issues and future trends.

1.4.1 Free-Form Features - A promising way to the modeling of free-form surfaces within a PDP

Even if today's CAD systems can easily represent free-form shapes by means of NURBS surfaces [35], their definition and modification still require a deep knowledge and a great skill in the manipulation of the underlying mathematical models. The implemented free-form shapes design operators are time-consuming and do not enable fast modifications. To overcome these limits, some researches have been undertaken to try to adapt the feature concept in the free-form domain.

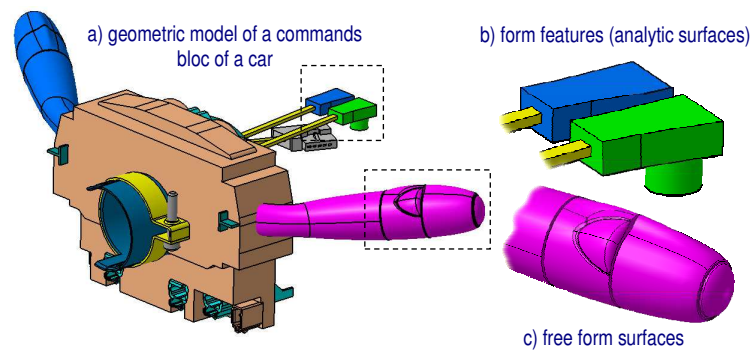


FIGURE 1.10: (a) Geometric model of a commands bloc of a car designed with (b) form features (analytic surfaces) and (c) free-form surfaces (courtesy PSA).

The feature concept has been successfully adopted for the design of shapes defined by analytic surfaces such as planes, cylinders, spheres and so on [39]. Following the feature-based approach, the geometric model is not anymore perceived as a collection of vertices, edges and faces but as a well-organized set of features corresponding to slots, ribs, stiffeners and so on. Figure 1.10.b shows an example on which there exist form features. All the properties related to a feature type are specified within a feature class that defines a template for all its instances. This always includes the generic shape of the feature, and a number of parameters, e.g. length, width, and constraints, e.g. parallel, perpendicular, that characterize this shape. By specifying values for the parameters, an instance of the feature class can be created and then be added to a feature model [13]. The feature model usually contains the feature instances, the information related to their mutual dependencies and the chronology of construction. Thus, the user do not manipulate directly the geometric entities but the features themselves through a set of control parameters that can be modified later on. The use of feature is also crucial when considering all the information that have to be associated to the parts all along the PDP. Two approaches are classically adopted for the definition of a feature model: the design by features and feature recognition strategies [44].

For several years, many researches have tried to extend the feature concept to the free form domain thus giving rise to a set of free-form features modeling strategies. To carry out appropriate analyses and comparisons between the existing feature-based approaches, suitable evaluation criteria and an adapted ranking system have to be adopted. On one hand, the various criteria define the framework for the analyses. They depend on the application domain needs and characteristics. Considering the aesthetic and engineering design of styling products, four main categories of criteria have been identified [30]:

- the criteria related to the adopted geometric models and features which characterize the nature of the manipulated representations (parametric, polyhedral and so on), the type of preservation (models are different or identical), the type of discontinuities that can be handled (pre-

scribed or free) and the dependencies to the underlying geometric models to identify approach that really work on the shape whatever the underlying representation is;

- those attached to the shapes and functions which distinguish approaches that have a shape classification or not, the fact that curves can be used or not, the possibility to parameterize the shape, the number of control parameters as well as the way validity constraints can be checked;
- those related to the families of shapes to highlight the capacity to handle local as well as global shape modification, the level of freedom in the shape definition, the level of control of the DOF including the possibility to impose heterogeneous behaviors;
- and those concentrating on the interactivity provided by the approaches to characterize approaches whose manipulations are tedious as well as approaches which do not incorporate high-level interactive operators.

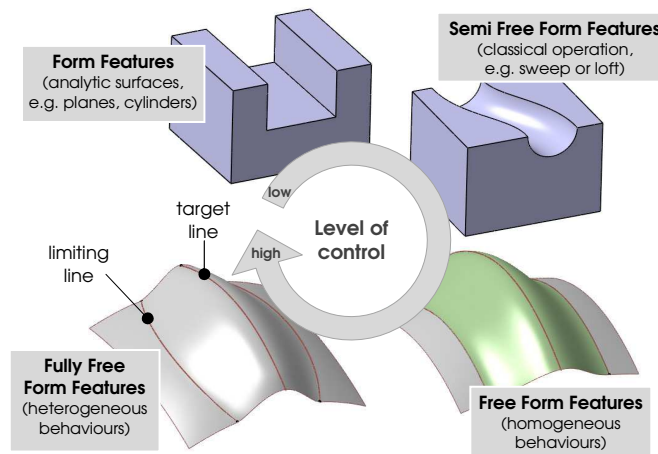


FIGURE 1.11: Features classification based on the concept of level of control [30].

Based on top of those evaluation criteria, three main categories can be distinguished depending on the level of control they provide on the resulting shapes (fig. 1.11):

- Semi free-form features defined by free-form surfaces, obtained by classical rules such as loft/sweep operations, interpolation rules or specific relationships directly expressed between the control points. The user often has few possibilities to control the resulting shapes. Actually, this category can be further refined depending on the type of constraints used to define the features. Point-based approaches [8, 41, 48, 49] define semi free-form features on top of points which often directly correspond to the control points of the underlying Bézier patches. Curve-based approaches [34, 43, 45] define parameterized free-form surfaces from a set of parameterized free-form curves. Finally, surface-based approaches [17, 16, 46] define semi free-form features directly from a bivariate function $z = f(x, y)$ which significantly reduce the applications;
- Free-form features defined by free-form surfaces, obtained through the use of adequate techniques expressing a homogeneous behavior over the whole surfaces [24]. There is more

freedom but handling the modified area is not always as free as possible. The insertion of the features inside the initial model often requires topological changes, e.g. trimming operations, which is always risky [25, 26];

- Fully free-form features characterized by a higher level of freedom in the definition of the shapes obtained through the use of techniques prescribing heterogeneous behaviors over different areas of free-form surfaces. The modified area can be bounded by any free-form curve while preserving the surface initial topology (no trimming operations nor patches insertion). Such features take advantage of the NURBS description power [35] and fit well the stylists' requirements.

More precisely, a fully free-form deformation features (δ -F⁴) corresponds to an area of a free-form surface that results from a deformation under specific constraint lines: the target, the limiting and driving lines [29]. The target lines give the global directions of the shape whereas the limiting lines are free-form curves used to give a shape around the target lines and bound the deformation area (fig. 1.12.a). In this approach, the shape is inserted without modifying the surface topology. It uses an automatic fixation algorithm to block the control points that have an influence outside the deformation area. Driving lines correspond to curves used to drive the repetition of more simple shapes (fig. 1.12.b). A restricted set of high-level parameters are used to control both the shape of the curves and the shape of the surface around these curves. There are the numerical parameters (e.g. position, dimensions and orientation), the shape control parameters (type of multi-minimizations) and the type of discontinuity (G0, G1). The numerical parameters are used to parameterize the features, the multiple minimizations enable the specification of heterogeneous shape behaviors [32] and the discontinuities can be inserted along any free-form directions. Additionally shape constraints, such as internal planar areas within a specific feature can be included. The various parameters used to control the constraint lines can be instantiated in two ways using the concept of feature template: either the designers directly specify their own constraint lines or they can instantiate numerical parameters (dimensions, angles, shape behaviors and so on) used to shape the constraint lines with the generic deformation engine. Once defined over the surface, δ -F⁴ can be manipulated in two ways: either in a prescriptive manner while modifying the different parameters that have been instantiated (dimensions, shape behaviors and so on) or in a qualitative manner while using a set of interactive deformation operators, e.g. push, pull. To select the δ -F⁴ to be instantiated and the associated template [31], a complete feature classification has been proposed.

The strength of the proposed approach lies in the perfect harmony between degree of flexibility and shape control. The concept of δ -F⁴ template enables mastered manipulations of the free-form curves that are imposed as constraints for the free-form surface deformation process. This double level of deformation (curve deformation followed by surface deformation) enables an easy access to a wide variety of shapes without limiting the designers' creativity. Moreover, the use of multi-minimizations initiates a new level of shape control thus justifying even more the term of fully free-form deformation features in comparison to the other approaches dealing with semi free-form features. The main drawback of this approach lies in the fact that solely some basic validity conditions can be defined and managed through a rudimentary checker.

Even if such an approach is already a good mean to help designers and stylists specifying shapes, there is still a gap between the available tools and the way people think. This has been widely studied within the Aim@Shape European Network of Excellence [6] from which a structured decomposition of shapes has been proposed (fig. 1.13). Actually, shapes can be decomposed in three distinct levels which interact the ones with the others: the geometric level defining the mathematical model of the shape, the structure level which helps decomposing the shape into areas on which specific information can be associated through the third level gathering together the semantic information. Therefore, and as described on figure 1.13, there is a need for developing top-down approaches that would manipulate shapes by mean of semantics. To this aim, several approaches have been imagined [21, 22] to try to set up a new approach where designers and stylists manipulate curves through

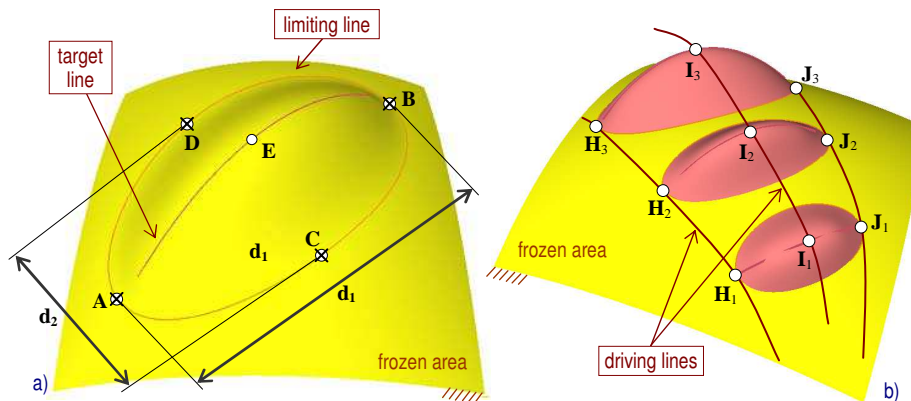


FIGURE 1.12: A basic shape feature (a) defined by target and limiting lines parameterized with few numerical parameters and heterogeneous deformation behaviours. A complex shape feature defined as a repetition of BSF along several driving lines (b).

a set of higher level control parameters like the straightness, the convexity or the acceleration of the curves [24]. Also, new approaches that would take into account the emotional aspects are foreseen.

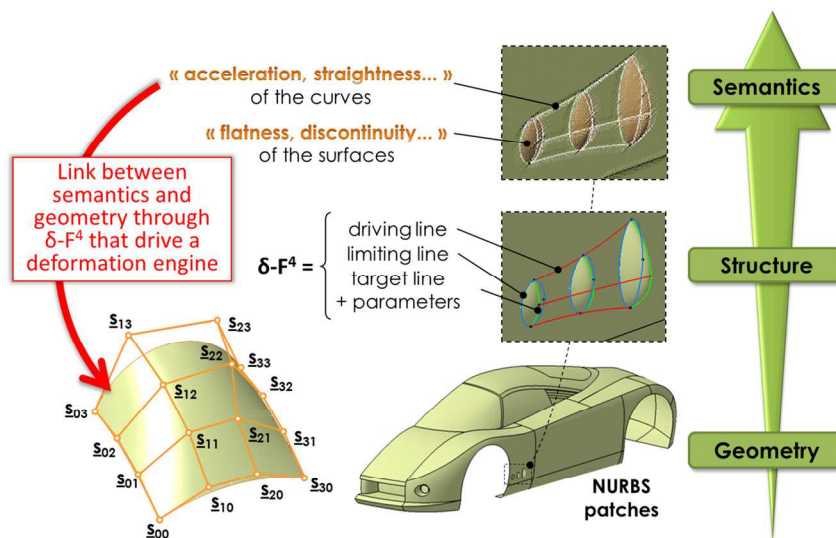


FIGURE 1.13: Structured decomposition of shapes and link between semantics and geometry through the use of $\delta-F^4$ that drive a deformation engine.

As a conclusion, adopted right from the beginning within an integrated semantic-based approach, the use of $\delta-F^4$ is very promising. Not only it helps the designers and stylists interacting with shapes without focusing on the underlying mathematical models, and the feature concept is also a good way to support many of the semantics used all along the PDP. However, there are still open issues and future trends that are developed in the next subsection.

1.4.2 Open issues and future trends

Even if the above mentioned approaches have deeply improved the way engineers can interact with free-form surfaces within a PDP framework, there are still many open issues as well as future trends.

Clearly, if some real improvements have been proposed in this domain, it is clearly still difficult to take into account the uncertainties the users and designers have when defining their shapes. Actually, the final shapes result from the integration of numerous constraints defined steps by steps by the different actors of the PDP. However, the global optimization problem is often not completely formalized and the final shapes often results from tedious manipulations and exchanges between the actors. Over-constrained configurations are not treated and it is therefore difficult to handle validity conditions.

Moreover, even if the feature concept is a real improvement in the way designers interact with the shapes, its use is often circumvent to the modeling of surfaces and its extension to the multi-representations of the product is not yet fully supported. Thus, there exist difficulties to interoperate and move from one representation to another, which is mandatory all along the PDP. In the free-form domain, the integration of structural and detail features is not fully supported and it is still difficult to have an efficient building tree. Effectively, when the building tree is available, very few modifications are possible since most of the time the curves and surfaces are dead, i.e. disconnected from the past history.

Actually, many of the actual limits can be ascribed to the underlying well-known NURBS mathematical model whose capabilities are not fully exploited. The weights, the knot sequences as well as the trimming lines are so many parameters often considered as fixed during the modification steps. Moreover, there exist many issues regarding the parameterization of those free-form surfaces as well as the decomposition in patches. Thus, the available modification mechanisms are often time-consuming and far from the users' needs. Of course, this breaks the development of a PDP framework where all the steps could be played in real-time to fit the end-user requirements. Such a new way for designing shapes require the definition of a completely new approach much more oriented towards the semantics and emotions rather than to the low-level constraints and underlying mathematical models. At the end, shapes should really be seen as a result of an emerging process integrating the numerous constraints manipulated by the different actors of the PDP.

In parallel, these new developments also have to take into account the new challenges arising from the use of new interesting low-cost Virtual Reality devices putting the end-user at the center of the PDP. Therefore, the scope of the PDP is widened and new applications are foreseen to do so that everyone can become a designer !

References

- [1] <http://www.123dapp.com>.
- [2] <http://www.3ds.com/products/catia/welcome/>.
- [3] <http://www.3dsom.com/>.
- [4] http://www.pixar.com/short_films/theatrical-shorts/luxo-jr.
- [5] <http://www.solidthinking.com/>.
- [6] Aim@Shape. <http://www.aimatshape.net>. *Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge embedded in multi-dimensional digital objects. European Network of Excellence, Key Action: 2.3.1.7 Semantic-based Knowledge Systems, VI Framework, 2004.*

- [7] T. Amabile. *Creativity in Context: Update to the Social Psychology of Creativity*. West View Press, Colorado, USA, 1996.
- [8] C.K. Au and M.M.F. Yuen. A semantic feature language for sculptured object modelling. *Computer-Aided Design*, 32(1):63–74, 2000.
- [9] C. Bajaj, J. Bloomenthal, B. Guo, J. Hart, and G. Wyvill. Implicit surfaces for geometric modelling and computer graphics. In *SIGGRAPH'96 Course Notes*, 1996.
- [10] P. Benk, R.R. Martin, and T. Varady. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 11:839–851, 2001.
- [11] M. Botsch, M. Pauly, C. Rossl, S. Bischoff, and L. Kobbelt. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, 2006.
- [12] C. Bradley and B. Currie. Advances in the field of reverse engineering. *Computer-Aided Design and Applications*, 2(5):697–706, 2005.
- [13] W.F. Bronsvort, R. Bidarra, and P. Nyirenda. Developments in feature modelling. *Computer-Aided Design and Applications*, 3(5):655–664, 2006.
- [14] M. Callieri, M. Dellepiane, P. Cignoni, and R. Scopigno. *Processing sampled 3D data: reconstruction and visualization technologies*. Taylor and Francis, 2011.
- [15] C.E. Catalano, B. Falcidieno, F. Giannini, and M. Monti. A survey of computer-aided modeling tools for aesthetic design. *ASME JCISE*, 2:11–20, 2002.
- [16] J.C. Cavendish. Integrating feature-based surface design with freeform deformation. *Computer-Aided Design*, 27(9):703–711, 1995.
- [17] J.C. Cavendish and S.P. Marin. *A procedural feature-based approach for designing functional surfaces*. H. Hagen (Ed.), Philadelphia, USA, 1996.
- [18] N. Cross. *Engineering Design Methods*. John Wiley, Chichester, third edition, 2000.
- [19] T. Dokken and B. Juttler. *Computational Methods fo Algebraic Spline Surfaces: An exploratory workshop*. Springer Berlin Heidelberg New-York, 2005.
- [20] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *ACM SIGGRAPH 1996*, pages 325–334, 1996.
- [21] Giannini F., Montani E., Monti M., and Pernot J-P. Semantic evaluation and deformation of curves based on aesthetic criteria. *Computer-Aided Design and Applications*, 8(3):449–464, 2011.
- [22] Giannini F., Monti M., Pelletier J., and Pernot J-P. A survey to evaluate how non designers perceive aesthetic properties of styling features. *Computer-Aided Design and Applications*, 10(1):129–138, 2012.
- [23] G. Farin, J. Hoschek, and M.-S. Kim. *Handbook of Computer Aided Geometric Design*. Elsevier Science B. V. Amsterdam, 2002.
- [24] Fiore. <http://www.fiore.com>. *Formalization and integration of an optimized reverse engineering styling workflow*. Brite-Euram Project N. BE96-3576, 1997-99, 1997.
- [25] M. Fontana, F. Giannini, and F. Meirana. A free form features taxonomy. In *Proceedings of Eurographics'99*, volume 3, pages 107–118, 1999.
- [26] M. Fontana, F. Giannini, and F. Meirana. Free form features for aesthetic design. *International Journal of Shape Modelling*, 6(2):273–302, 2000.

- [27] X.W. Gao, Y. Qian, and R. Hui. The state of the art of medical imaging technology: from creation to archive and back. *Open Medical Informatics Jou.*, 5(1):73–85, 2011.
- [28] Gain J. and Bechmann D. A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics*, 27(107):1–21, 2008.
- [29] Pernot J-P., Falcidieno B., Giannini F., and Léon J-C. Fully free-form deformation features for aesthetic shape design. *Journal of Engineering Design*, 16(2):115–133, 2005.
- [30] Pernot J-P., Falcidieno B., Giannini F., and Léon J-C. Incorporating free-form features in aesthetic and engineering product design: State-of-the-art report. *Computers in Industry*, 59(6):626–637, 2008.
- [31] Pernot J-P., Giannini F., Falcidieno B., and Léon J-C. Parameterised free form feature templates. In *Proceedings of the IEEE Shape Modelling International Conference (SMI'09)*, 2009.
- [32] Pernot J-P., S. Guillet, Léon J-C., Falcidieno B., and Giannini F. Shape tuning in fully free-form deformation features. *Journal of Computing and Information Science in Engineering*, 5(1):95–103, 2005.
- [33] Panchetti M., Pernot J-P., and Véron P. Towards recovery of complex shapes in meshes using digital images for reverse engineering applications. *Computer-Aided Design*, 42(8):693–707, 2010.
- [34] P.J. Nyirenda, M. Mulbagal, and W.F. Bronsvort. Definition of freeform surface feature class. *Computer-Aided Design and Applications*, 3(5):665–674, 2006.
- [35] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer Verlag, 1997.
- [36] R. Raffin. *Free Form Deformations or Deformations Non-Constrained by Geometries or Topologies*. Hidalgo et al. (Ed.). Springer Verlag, 2013.
- [37] T.W. Sederberg, G.T. Finnigan, X. Li, and H. Lin. Watertight trimmed nurbs. *ACM Transactions on Graphics*, 27(3), 2008.
- [38] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.
- [39] J.J. Shah and M. Mäntylä. *Parametric and feature-based CAD/CAM*. Wiley-Interscience Publication, John Wiley Sons, Inc., 1995.
- [40] C. Shao, A. Bousseau, A. Sheffer, and K. Singh. Crossshade: Shading concept sketches using cross-section curves. *ACM Transactions on Graphics, SIGGRAPH*, 2012.
- [41] Y. Song, J.S.M. Vergeest, and I. Horváth. Reconstruction free-form surface with parameterized features. In *ASME DETC'02*, 2002.
- [42] I.E. Sutherland. *Sketchpad: A man-machine graphical communication system*. PhD thesis, MIT, 2013.
- [43] E. van den Berg and W.F. Bronsvort. Parameterised, constraint-based wrapping of freeform shapes. *Computers & Graphics*, 31(1):89–99, 2007.
- [44] E. van den Berg, W.F. Bronsvort, and J.S.M. Vergeest. Freeform feature modelling: concepts and prospects. *Computers in Industry*, 49(2):217–233, 2002.
- [45] E. van den Berg, H.A. van der Meiden, and W.F. Bronsvort. Specification of freeform features. In *ACM Symposium on Solid Modelling and Applications*, pages 56–64, 2003.

- [46] P.A. van Elsas and J.S.M. Vergeest. Displacement feature modelling for conceptual design. *Computer-Aided Design*, 30(1):19–27, 1998.
- [47] T. Varady, M.A. Facello, and Z. Terek. Displacement feature modelling for conceptual design. *Computer-Aided Design*, 39(5):379–388, 2007.
- [48] J.S.M. Vergeest and I. Horváth. Reconstruction free-form surface with parameterized features. In *Proceedings of the IEEE Shape Modelling International Conference*, pages 20–28, 2001.
- [49] G. Vosniakos. Investigation of feature-based shape modelling for mechanical parts with free form surfaces. *Int. Jou. of Advanced Manufacturing Technology*, 15(3):188–199, 1999.