Towards a new generation of geometrical constraints in parametric sketches

Pedro Soza
Universidad de Chile. Chile.
Georgia Institute of Technology.USA.
psoza@uchile.cl -psoza@gatech.edu

Abstract: *Geometrical constraints are an open issue because the current technology only provide basis to defining constraints when sketching. Sketching with constraints requires the ability to define and visualize future directions in the design process. Therefore, decision making along design processes relates to the possibilities the constraint allows for the designer. This proposal seeks a way to improve constraints election and definition. We propose a tool to help designers see in advance the future states of their designs when defining constraints. The tool will also register constraint definitions, allowing designers to reuse cognitive effort and knowledge by constructing a reusable library of cases for design intent.*

Palabras clave: Geometrical Constraints, Parametric Modeling, Design Processes

Problem Definition

Parametric modeling provides mechanisms to embed domain expertise and design intent into a set of topological and geometric relationships, allowing automation of low-level tasks and assisting and informing designers with relevant knowledge about the strength of current design alternatives (Eastman, Lee, & Sacks, 2003). However, the current state of development in this technology fail in capturing ambiguity, complexity, and sharing information along the design process. Usually there is more than one way to solve a design and to build a model. This idea leads to the notion of ambiguity. Moreover, design and modeling problems usually become different facets of the problem. Some solutions are driven by the design intent while other are driven by that the application allows or the user knowledge and expertise about it.In relation to the notion of complexity, an unsolved problem is that the amount of data grows exponentially. The number of parameters can be very large. For instance, (Anderl & Mendgen, 1996) provide an example of a gearbox composed by 665 features, 1363 parameters, and 1291 geometric constrains between features. Even when in the future this could be faced with enough computational power, the cognitive load the designer can face makes almost impossible to handle all this complexity. This seems to be a processing problem and not a representational one. Also, product of this complexity, the risk of propagation error increase exponentially.Sharing is the last limitation depicted by (Eastman, Lee, & Sacks, 2003) and is grounded on the idea that ambiguity and complexity becomes an obstacle to share design intent, building information, and tacit knowledge embedded on parametric models.

Cognitive scope

Gero and Kelly stated that designers interpret the world through their expectations (Gero & Kelly, 2009). Indeed, according to them expectations are derived from the situation in which designers are immersed. Following this idea, constraints definition should play an important role while defining the first sketches of a design problem. Design problems are ill-defined (Eastman C. , 1969), therefore we must consider that the first sketches of a design problem play an important role redefining the design problem. Moreover, the initial states of the design process are related to constraints definition. According to (Aish, 2005) design is fundamentally about the creation of rules. These rules are the definition of proper relationships between design components. Aish and Eastman describe design as a process in which decisions are made with incomplete information. Thus, the designer needs to be predictive in order to anticipate what the consequence of the 'making' or 'doing' will be. "How can we progress from intuition to precision? How can we augment the cognitive processes? How can we record the progression of ideas?" are some of the questions raised by (Aish, 2005). Moving forward, there is a lack of definition of the characteristics a computational design tools should provide to facilitate problem solving in design. Also, we are far from defining a method to use parametric modeling tools or to define how these tools are internalized and operated by designers.

Constrains Definitions

Constraints are explicit dimensions of distances and angles, as well as constraints of parallelism, incidence,

perpendicularity, tangency, concentricity, and prescribed radii. The user specifies a rough sketch and adds to it geometric and dimensional constraints. According to (Anderl & Mendgen, 1996) the designer sketch the topology of a shape by picking point and drawing lines and arcs on the screen and then applying linear and angular dimensions and dependencies such as horizontality or parallelism or others to the topology of the objects on the screen. This process is called constraining the sketch. The dimensions and dependencies are referred to as the constraints. Internally, on the application, all the constraints refer to topology and geometry objects, most of them to the vertices and their geometric coordinates. The geometric coordinates are called parameters, which are constrained. The main aspects of modeling with constraints are structuring a solid model as a history of features, using topology objects and their geometric coordinates as parameters and applying constraints to these objects.

Researchershave defined constraints in different ways. Some examples of those definitions can be found in (Gross, 1978), (Anderl & Mendgen, 1996), and (Hoffmann & Kim, 2001).Gross define *"design as a set of constraints, or relations on a set of variables"*(Gross, 1978). For him constraints are rules, requirements, relations, conventions, and principles that define the context of design. Some are imposed externally, while others are imposed by the designer. Some are site-specific, others not. Some are the result of higher-level design decisions; some are universal, a part of every design as gravity for instance (Gross, 1978).Anderl&Mendgen recognized the influence of constraints definitions in design processes as Gross, but they narrowedthe scope of their work to the use of geometrical and dimensional constraints in constructive solid geometry (CSG). They observed that *"designer sketches the topology of a feature's shape by picking points and drawing lines and arcs on the screen and then applying linear and angular dimensions and dependencies (or properties) such as horizontality or parallelism to the topology-objects"*(Anderl & Mendgen, 1996). They call this process "constraining the sketch" and the dimensions and dependencies defined by the designer are referred to as the constraints.For Hoffmann & Kim,failure to renew or explore new designs possibilities is related to the parameterization of the shapes. They pose the general question: "Given a parametric solid and its constraint schema, what are the valid ranges for its dimensional constraints and parameters?" (Hoffmann & Kim, 2001). Clearly, design methods can help, but they are not enough to solve all the problems raised. On their work they posed a series of rules. The first one states that a polygon is valid in its constraints if the resulting contour is closed and simple. Simple refers its topological relation and states the polygon is no overlapped with itself. They develop an algorithm to solve dimensional constraints on polygons, allowing changing the shape, yet maintaining the topology. They also come up with a tree graph to check if the constancy of the constraints is well done. For doing this, they replace each edge of the polygon with a node in the tree, and joint the nodes with the constraint representation (just dimensional). In general, polygons are under-constrained if the graph is not connected, is over constrained if the graph is closed, and if the polygon is well connected the graph is a free tree.

(Bouma, Fudos, Hoffmann, Cai, & Paige, 1995)concentrate their efforts on sketch representation as the basis for archiving sketches in neutral format, with the ability to retrieve the archived sketch and edit them later. Their constraint solver determine the geometric elements that are to be found, and processes the constraints to determine each geometric element such that the constraints are satisfied. On the other hand, (Hoffmann & Kim, 2001) stated that failure to renew or explore new designs possibilities is related to the parameterization of the shape. They develop an algorithm to solve dimensional constraints on polygons' allowing changes on shapes while maintaining the topology.

As summary we find two approaches to defining constraints. The constructive approach and the rule based approach. In addition, the internal representation of constraints follows two alternatives. It can be as a set of predicates or rule-based, or it can be represented by algebraic equations. Algebraic equations are the most popular way to represent constraints, due the economy of applying the dimensional and geometrical relations to the topology as equations. Finally, to solve the constraints the literature gives account principally of three methods. These are the numeric, the symbolic, and the rule-based methods. The numeric approach solves the constraint network based on an iterative, numerical algorithm. This solution was implemented by Sutherlands on his Sketchpad system in 1963. It may be used when the constraints are expressed as algebraic equations in the implicit form (f(x) = 0). The symbolic approach it uses symbolic calculation of equations. Based on a constraint network, a symbolic calculation of the equations evaluates all possible solutions for the coordinates of the characteristic points. Multiple solutions are very common because many constraints are expressed as quadric

equations. For the rule-based approach, the constraints are expressed by rules or predicates. This is the most common method is use today.

Technical Approach

We propose solving constraintsusing A.I. techniques based insearch and optimization algorithms. These techniques will offer possible design solutions according to the constraints being selected. The general scheme of our proposal is as follows:

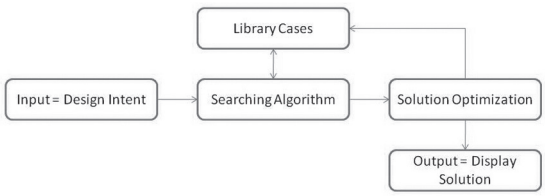

Fig. 1 General scheme

On our general scheme, the user gives an input to the system when applying a geometric constraint to some object or feature on the screen. Once accepted that constraint, the system, if is active, will search possible futures states on the library of cases (database). When solutions are found, they will show up as future states of the design to the user on the screen. When the user accept a recommendation, that solution should be optimized and write to the database, increasing the number of solutions, and keeping track of the design states. These design states are design decisions. Then, the final solution will be displayed on the screen. We have chosendifferent tree search algorithms to explore our proposal because on them each node of the graph is visited in a systematic way. Such visit occurs following the order of the tree from the initial state up to the goal state.

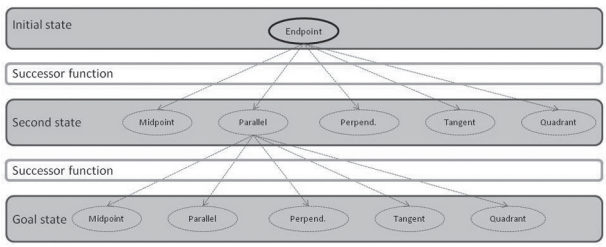

Fig. 2 Diagram of tree navigation with initial state, goal state and functions layers

There are several tree search algorithms and search strategies. The explanation of the options already pre-elected by our team is presented below.

Evaluation uniformed search strategies:

Goal-based agents can succeed on their search task by considering future actions and the desirability of their outcomes. Indeed, search algorithms take a problem as input and return a solution in the form of an action sequence. In our work we'll be evaluating five well known search algorithms:breadth-first, uniform-cost, depth-first, depth-limited, iterative deepening, bidirectional search.

However, any of these algorithms needs to be updated to find the goal state when the information is incomplete. These are called search with partial information algorithms. They use heuristic strategies running over one of the previous strategies to inform the search task and solve the problem.

Informed search or heuristic strategies

We selected five approaches to explore the solution space to the proposed problem:Best-First Search, A* search, Alpha-Beta Pruning, Beam search, Hill Climbing, and Genetic Algorithms. A key component of these algorithms is called the *heuristic function*, denoted $h(n)$. In this function $h(n)$ is equal to the estimated cost of the cheapest path from anode n to the goal state. Heuristic functions are the most common way in which additional knowledge of the problem is imparted to the search algorithm. Heuristic search is at the core of our problem formation, thus is the main reason to explore this various algorithms.

Expected significance

Parametric models are not reusable neither recyclable and we lose them and the knowledge embedded in them after the design process. Our proposal will tackle this situation.

The progression of operations using hierarchies and allowing changing parameters is a useful tool considering different design options. In addition, this register can serve as basis for analysis of the design process and the generation and use of knowledge by architects, becoming a tool for improve performance in design process and a knowledge repository. A efficient way to simplify the process of constraint definition imply reviewing the redundancy of operations. Also, is difficult to understand and rely on the constraints defined by others. Indeed, the exchange of constraints based models is an open issue. The degrees of freedom in sketches and evaluation of design alternatives are also and open issue. Automated

reasoning and design generation provide powerful capabilities for design practice, but that is not possible in large scales or complex design proposals yet. Our proposal seeks to tackle these issueson early stages of the design process.

Generative systems offer the potential for radically changing the way designers interact with their designs, allows check for consistency, generate alternatives and modifying design. We consider this relevant because up to date there is a lack of analysis in the life cycle of parametric elements, as well as in the project life versus the design process. We assume that when these elements are stabilized on the design phase, no one wants to add more changes to them. Yet we do not know which were those properties allowing stabilization of the project being design, and if those properties could be changed. Our proposal should reveal and expose those states and register them on the library of cases. Through this way, our proposal will approach to a repository of design intends and methods.

References

- Aish, R. (2005). From Intuition to Precision.

Anderl, R., & Mendgen, R. (1996). Modelling with constraints - theoretical foundation and application. *Computer Aided Design, 28* (3), 155-168.

- Barker, S. M. (1995). Towards a topology for computational geometry. *Computer Aided Design, 27* (27), 311-318.

- Bouma, W., Fudos, I., Hoffmann, C., Cai, J., & Paige, R. (1995). Geometric constraint solver. *Computer Aided Design, 27* (27), 487-501.

- Eastman, C. (1969). Cognitive processes and ill-defined problems: a case study from design. *International Joint Conference on Artificial Intelligence* (pp. 669-690). Washington D.C.: Donald E. Walker, Lewis M. Morton.

- Eastman, C. M., Lee, G., & Sacks, R. (2003). Parametric 3D modeling in building construction with examples from precast concrete. *Automation in Construction, 13*, 291– 312.

- Eggink, D., Do, E., & Gross, M. (2001). Smart Objects: Constraints and behaviors in a 3D design environment. *eCAADe.*

- Gentry;, A. R. (2009). *Rich Knowledge Parametric Tools for Concrete Masonry Design.*

- Gero, J., & Kelly, N. (2009). Constructive Interpretation in Design Thinking. *eCAADE*, (pp. 97-104).

Gourtovaial, A. M. (2003). Towards Integrated Performance-Based Generative Design Tools.

- Gross, M. (1978). *Design as Exploring Constraints.* MIT.

- Heisserman, J. (1994). Generative Geometric Design. *Computer Graphics and Applications, 14:2*, 37-45.

Heisserman, J., Callahan, S., & Mattikalli, R. (2000). A Design representation to support automated design generation. In J. Gero (Ed.)., (pp. 545-566).

- Hoffmann, C., & Kim, K. J. (2001). Towards valid parametric CAD models. *Computer Aided Design, 33*, 81-90.

- Kalay, Y. (1999). Performance-Based Design. *Automation in Construcction, 8*, 395-409.

- Kannengiesser, J. U. (2004). The situated function–behaviour–structure framework. *Design Studies, 25*, 373–391.

- Kilian, A. (2005). Design Innovation through constraint modeling. *eCAADe*, (pp. 671-678).

- M., C. C. (1995). On editability of feature-based design. *Computer Aided Design, 27* (12), 905-914.

- Martini, K. (1995). Hierarchical geometric constraints for building design. *Computer Aided Design, 27*, 181-191.

- Medjdoub, B. (1999). Interactive 2D Constraint-Based Geometric Construction System.

- Meiden.;, A. v. (2010). Tracking topological changes in parametrics models. *Computer Aided Geometric Design, 27*, 281-293.

- Ramin;, R. A. (2009). Physics-Based Generative Design.

- Sacks;, G. R. (2005). Specifying Parametric Building Object Behavior (BOB) for a Building Information Modeling System. *Automation in Construction, 15*, 758 – 776.

- Shelden, J. D. (2004). A parametric strategy for free-form glass structures using quadrilateral planar facets. *Automation in Construction, 13*, 187– 202.

- Thabet;, K. W. (2003). Building assembly detailing using constraint-based modeling. *Automation in Construction, 12*, 365– 379.

- W., E. B. (1999). Solving Form-Making Problems Using Shape Algebras and Constraint Satisfaction., (pp. 620-625).

- Wright;, S. D. (2000). From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge. *Computer Aided Design, 32*, 851-866.

- Xiao-Shan, G., & Shang-Ching, C. (1998). Solving geometric constraint systems. I. A Global propagation approach. *Computer Aided Design, 30* (1), 47-54.

- Yeon-suk;, J.-m. J. (2009). Automatic rule-based checking of building designs. *Automation in Construcction, 18*, 1011-1033.