

Desenvolvimento de partidos geométricos para construções verticais utilizando de geometrias complexas, ferramentas paramétricas e generativas

Development of geometric parties for vertical constructions using complex geometries, parametric and generative tools

Pedro Oscar Pizzetti Mariano

PosArq, UFSC, Brasil
pedro.pm@hotmail.com

Alice Theresinha Cybis Pereira

PosArq, UFSC, Brasil
acybis@gmail.com

Abstract

In order to provide a range of complex forms, justifiable to be selected in different aspects, the present work developed and evaluated a parametric and generative algorithm that allows to create diversified geometric options. The development of this work occurred in four stages, initially with the collection of information in the areas of architecture, parametric design and tools. Then the algorithm was developed with the aid of visual programming software. After, the geometric result was recreated in a physical model. Finally, we verified the potentialities of the use of the algorithm, and the use of a generative method.

Keywords: Generative Architecture; Parameterization; Complex Geometry; Grasshopper.

Introdução

Com o surgimento de teorias, ferramentas e metodologias, em diferentes campos do conhecimento, como arquitetura, design e engenharia, novas potencialidades e áreas de atuações foram criadas, integrando diferentes disciplinas, como programação, computação, etc. Como exemplo da ampliação das tecnologias na arquitetura pode-se citar o uso de softwares paramétricos, generativos e programas BIM. Que além de organizar e ampliar a capacidade dos projetistas, possibilitam o aprimoramento da criação de formas geométricas diferenciadas para determinadas situações. (Alvez 2012).

Esta expansão e integração de conhecimentos em diversas disciplinas, trouxe à tona ferramentas computacionais que potencializaram a atuação do arquiteto e urbanista. Oxman (2008), descreve duas destas linhas de ferramentas, o DAD e o CAD. O DAD (*digital architecture design*), possui a ideia da utilização das ferramentas digitais no desenvolvimento de projetos. E o CAD (*computer aided design*) exprime a ideia de introduzir ferramentas que antes eram utilizadas em papel para uma interface digital.

Tais ferramentas ampliaram a utilização dos computadores nas etapas de concepção do partido e nas tomadas de decisões de projetos. Possibilitando que a geometria não esteja somente associada a construção por seu apelo estético, mas também por uma interpretação de seu entorno, permitindo um design eficiente em relação as condicionantes onde a construção se situa. Para o desenvolvimento de um projeto que leve em conta o seu entorno e demais condicionantes, diversos parâmetros devem ser analisados como o custo, funcionalidade, gastos, entre outros. (Rashad e Afaris 2010),

E com o auxílio de softwares generativos e de programação visual simplificada, arquitetos e designers podem ampliar sua

biblioteca formal. Pois em diversas situações a capacidade de se gerar formas complexas é limitada pelos conhecimentos da geometria ou de programas modeladores. (Riam e Asayama, 2016; Gürbüz, çagdas e alaçam 2010).

Assim com o intuito de se proporcionar uma nova gama de opções para projetistas e designers se apropriarem de formas complexas, e permitam que as mesmas possam ser simuladas e avaliadas, o presente trabalho desenvolveu um algoritmo paramétrica e generativo que permite criar opções diversificadas em designs verticais com geometrias complexas.

Procedimentos Metodológicos

Inicialmente foram buscadas informações pertinentes a plug-ins que auxiliaram no desenvolvimento do algoritmo, junto com a busca e leitura de trabalhos vinculados ao tema. Após se iniciou a montagem do algoritmo em um plug-in de programação visual. Para este procedimento se utilizou o modelo interativo incremental, método que é descrito posteriormente no trabalho.

Com o código consolidado uma outra sequência foi desenvolvida e anexada, utilizando de um algoritmo genético. Com isto se aplicou um padrão desejado, e se selecionado a forma topológica que melhor condizia com os aspectos quantitativos (referentes a radiação), e subjetivos (juízo estético) desejáveis.

A partir desta definição do volume, a forma teve suas diferentes partes planificadas digitalmente e cortadas com o auxílio de uma cortadora a laser, na escala 1:100, permitindo a visualização e avaliação do resultado de maneira coesa. Por último se descreve o algoritmo e modelo físico, demonstrando as potencialidades da construção e emprego *Generative Performance Based Design*, para construções de partidos

arquitetônicos verticais. A utilização destes métodos pode ser identificada na sequência de ações pela figura 1.

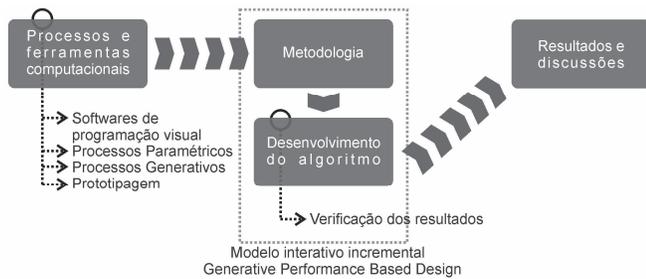


Figura 1: Grafo com os procedimentos do trabalho.

Fonte: Mariano P.O. P. (2017).

Processos e Ferramentas

Antes da introdução das ferramentas digitais no campo do desenvolvimento de projeto e construção, AEC (*Architecture, Engineering and Construction*), a utilização de programas paramétricos já era utilizada pela indústria da construção aeroespacial, desenvolvimento náutico e automobilístico. Este uso que antes era exclusivo destas indústrias, atualmente também auxilia designers e arquitetos a desenhar e modelar seus projetos de uma maneira simplificada. Estas ferramentas permitem fazer mudanças e variações no projeto, acarretando em um refinamento da qualidade final da proposta, pois possibilitam que designers e projetistas avancem e retrocedam etapas em diversas situações de projeto, aprimorando o modelo, a cada nova solução proposta. (Barrios 2004; Salim, Burry 2010 e Dantas, 2010)

Softwares de Programação Visual

Para Ercan e Elias-Ozkan (2015) o desenvolvimento de programações paramétricas e generativas em trabalhos que envolvam simulações e necessitam seguir uma série de performances, requer uma gama de habilidades para a construção desta programação. A formulação de um uma área de trabalho personalizada (*workflow*), pode requerer leitura, escrita, interpretação e manipulação de um banco de dados de determinado programa. E com o desenvolvimento de novas interfaces amigáveis, (*user-friendly*) e plug-ins para programações paramétricas, profissionais tendem a explorar ferramentas sem a necessidade de especialistas no assunto os acompanhando.

As programações visuais ou (*visual programming languages*) VPL's, são uma amostra destes ambientes com interfaces amigáveis para a programação. Os VPLs são organizados e desenvolvidos a partir de elementos gráficos já existentes, que são movimentados, ajustados e conectados, formando o código desejado. A forma de se criar os códigos em VPL's assemelha-se analogamente a organizar caixas e conectar fios, (*box-and-wire*) que se interligam. (Celani e Vaz, 2012).

Para este estudo se utilizou a ferramenta de programação visual Grasshopper, este plug-in foi introduzido no software Rhinoceros 3D a partir de sua versão 4, possibilitando uma programação de linguagem denominada (*Explicit History*). Sua *interface* não necessita de profissionais especializados em programação, para desenvolver suas regras e algoritmos, no software é possível adicionar inputs, outputs, parâmetros e outras regras. (Ercan e Elias-Ozkan 2015).

Processos Paramétricos e Generativos

Para Gürbuz, Çagdas e Alaçam (2010) o papel da computação no processo do design arquitetônico é um tema que vem se discutindo a mais de três décadas, e pode ser um precursor para que projetistas consigam alterar suas metodologias de desenvolvimento do partido arquitetônico. Uma das abordagens ainda comumente usadas por arquitetos, urbanistas e designers é de se utilizar ferramentas computacionais como um finalizador do processo, principalmente como uma ferramenta de visualização ou desenho. Uma tendência é tentar modificar a utilização das ferramentas computacionais no processo da formulação da proposta, pois novas abordagens na prática da arquitetura e design começam a se expandir, permitindo a inserção de informação no modelo, uso de recursos paramétricos e generativos.

Para os autores Salim e Burry (2010) e Barrios (2004), a principal manipulação que os softwares paramétricos podem disponibilizar para o processo de design, está associado a linearidade do desenvolvimento do mesmo ou IPO, (*input process and output*). Este processo difere dos comumente usados em projetos, como as ferramentas de manipulação direta na geometria. Nestes processos os inputs e outputs estão constantemente abertos, as diferentes tarefas, disciplinas e dimensões conseguem sofrer real colaboração e interação. Com estas ferramentas é possível se observar diferentes variações e prever e reconfigurar possíveis mudanças dentro do projeto.

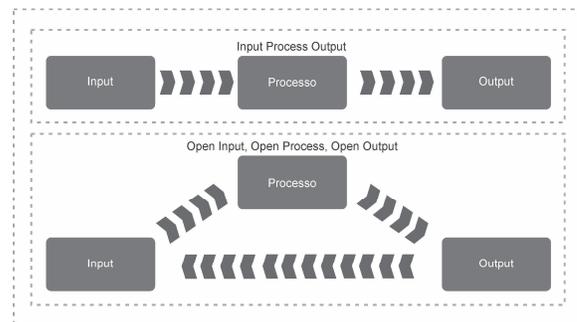


Figura 2: Modelo de processo paramétrico.

Fonte: Mariano P.O. P. (2017).

Já no processo generativo, o projetista tende a limitar a geração das formas através da definição de uma série de parâmetros. Isto ocorre pois dentro dos sistemas generativos pode-se definir uma série possibilidades abstratas, que permitem um conjunto de potenciais soluções no processo final do produto. Estes métodos generativos não geram o produto em si, eles constituem ferramentas que possibilitam diversas tarefas que resultam em produtos. (Fisher E Herr, 2000).

Este processo se mostra vantajoso na concepção de projetos em arquitetura, pois o designer e arquiteto pode manipular inúmeras ideias geradas a partir da combinação destas duas ferramentas. A utilização de algoritmos generativos também permite o desenvolvimento de um conjunto de solução que poderiam não ser alcançadas devido a capacidade de intuição do projetista ou seus conhecimentos prévios sobre os fatores que o software está simulando. (Ercan E Elias-Ozkan 2015).

Tais processos paramétricos também possibilitam parametrizar modelos desenvolvidos e a combinação de diversas peças modeladas e parametrizadas a partir de uma regra ou código desenvolvido. Fisher et al (2003), complementa este pensamento, afirmando que os modelos desenvolvidos por ferramentas digitais podem ser testados de maneira instantâneas a partir de ferramenta e máquinas que possibilitam a prototipagem, como corte a lasers, CNCs, impressoras 3D em diversos materiais, o uso da realidade aumentada, realidade virtual, entre outras.

Metodologias

O modelo utilizado no trabalho é modelo de programação iterativa e incremental, o MII foi escolhido por uma gama de vantagens em seu uso, esta metodologia permite a visualização e verificação das etapas prontas do modelo antes da finalização total da programação. Assim os testes conduzem um feedback ao projetista em diversas etapas da programação, evitando que sugestões e mudanças importantes sejam recebidas somente na conclusão do desenvolvimento. (BONA 2002).

O modelo iterativo e incremental é dividido em quatro etapas organizadas em uma sequência de atividades. Segundo Cantor (1998) e Bona (2002), o início de uma etapa seguinte não tem o significado que a etapa anterior tenha sido finalizada, fazendo com que o projeto possa avançar mesmo que algumas das etapas anteriores necessitem de novas validações.

Cantor (1998) e Bona (2002), também dividem o modelo iterativo e incremental nestas quatro etapas: Concepção, elaboração, construção e transição.

A concepção (começo), é definição de uma predefinição do produto que será entregue. Na segunda etapa, elaboração, é alcançada a forma (compreensão), é desenvolvido o que é necessária para se ter o produto final, e tem-se o objetivo de que a programação fique organizada. Na fase de construção é iniciado a compilação do produto, onde são feitas as principais codificações do sistema e também as tarefas referentes a análise e projeto. E a última etapa é a transição, onde ocorre a entrega do produto e se verifica se foi atendido os objetivos inicialmente planejados. Estes passos podem ser compreendidos no desenvolvimento da proposta do algoritmo pela figura 3.

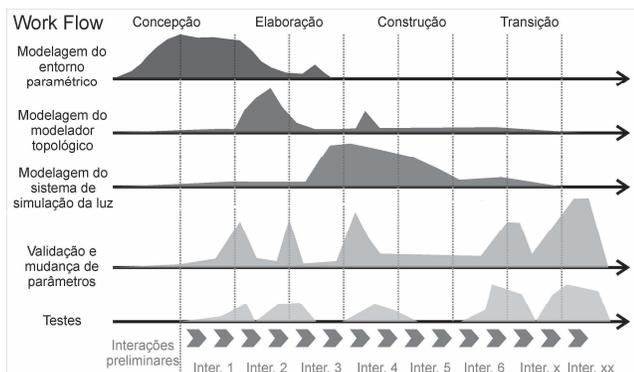


Figura 3: Modelo do MMI para o algoritmo.

Fonte: Mariano P.O. P. (2017).

Desenvolvimento do Algoritmo

A partir do modelo MMI comentado anteriormente, o algoritmo se constrói em 3 modelagens principais, a criação do entorno paramétrico, do modelador topológico e o sistema de simulação da radiação.

A primeira sequência de construção do entorno paramétrico, tem como objetivo criar um ambiente cabível de ser simulado, podendo referenciar ambientes construtivos reais. Para isso se desenvolveu um entorno de 9 quadras que podem ter uma série de parâmetros alteráveis para se aproximar a um ambiente real, como representado pela figura 5.



Figura 5: Algoritmo de entorno e seu resultado

Fonte: Mariano P.O. P. (2017).

Após se desenvolver um entorno randômico, se selecionou um dos lotes da quadra central para se criar a estrutura complexa. Para o desenvolvimento desta geometria se utilizou dois plug-ins para o Grasshopper, o Millipede e o DIVA.

No lote selecionado é gerado uma estrutura de dimensões parametrizáveis, com uma determinada área em x e y delimitada pelo tamanho do lote, peso próprio, e altura em relação a cota 0 do lote. A altura que esta estrutura tem em relação ao terreno é o que irá definir a altura da estrutura da construção, que para o trabalho foi dimensionada com 50m. Junto com esta estrutura, 6 pontos são criados na área de projeção desta primeira estrutura mencionada. Os pontos são postos de maneira aleatória a partir de um numero pré-estabelecido, e pelo plug-in Millipede, uma forma geométrica topológica é gerada, ligando estes pontos a estrutura elevada do solo. Como demonstrado na figura 6.

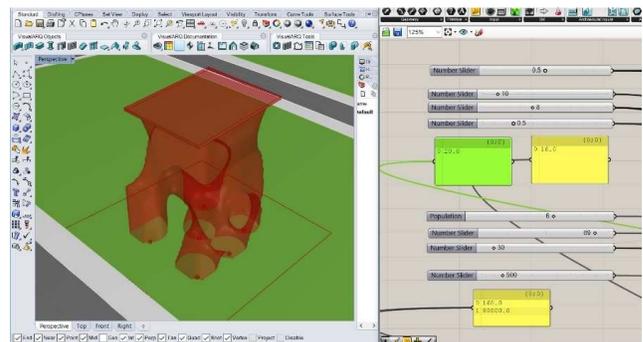


Figura 6: Forma geométrica criada pelo Millipede

Fonte: Mariano P.O. P. (2017).

Esta forma então é fatiada em alturas representativas para um andar habitável, e a partir destas curvas é desenvolvido uma estrutura, compreendendo fechamento e lajes. Estes fechamentos então podem ser avaliados pelo plug-in DIVA, permitindo se identificar uma série de comportamentos da luz natural em relação a forma. Para este trabalho se decidiu utilizar a radiação, avaliada de maneira dinâmica, tendo assim uma média anual da radiação incidente na fachada, conforme demonstrado na figura 7.

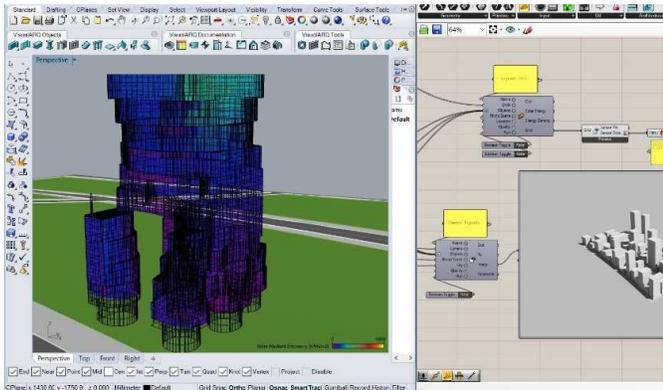


Figura 7: Avaliação da radiação pelo DIVA
Fonte: Mariano P.O. P. (2017).

Por meio dos resultados dessa avaliação da radiação solar, se consolidou uma métrica máxima onde pôr meio de um algoritmo genético, se desenvolveria a forma que mais se adequasse a este resultado, no caso buscando-se a forma que mais recebesse radiação. Com isto o processo do *Generative Performance Based Design* se consolidou, pois, a partir deste limitador (radiação), o algoritmo genético, Galápagos, alterava o padrão da ordem dos 6 pontos de maneira randômica, fazendo com que uma nova forma topológica se desenvolvesse, e assim uma nova avaliação da radiação se identificasse. Este processo ocorreu o número de vezes estabelecido pelo limite de variações de pontos pré-definidos pelo parâmetro, (60 vezes), até o algoritmo demonstrar a forma que mais recebia anualmente radiação. Como resultado se gerou a forma 52 demonstrada pela figura 8. Na sequência está a figura que representa algoritmo completo com todas as etapas, figura 9.

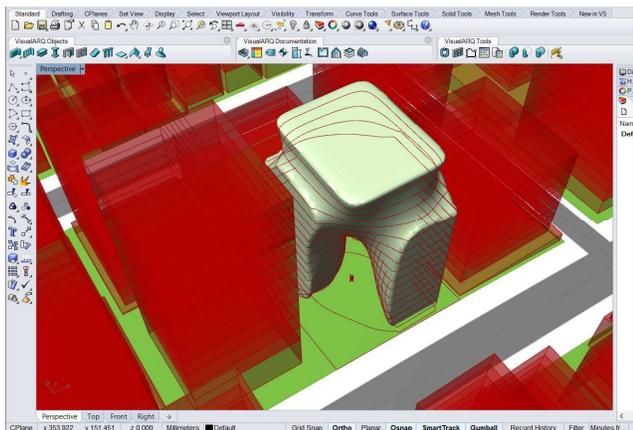


Figura 8: Forma geométrica selecionada gerada pelo Galápagos
Fonte: Mariano P.O. P. (2017).

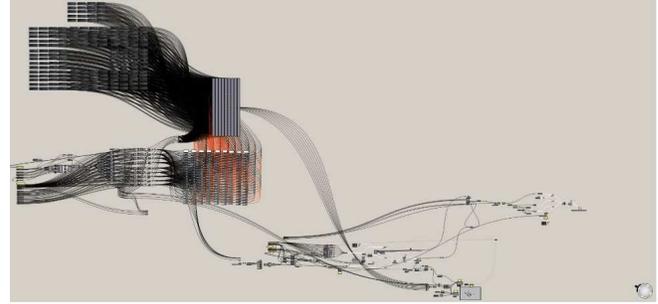


Figura 9: Algoritmo completo
Fonte: Mariano P.O. P. (2017).

Com as definições virtuais e simulações já contestadas em um ambiente digital, ainda restava a verificação real da geometria desenvolvida pelo algoritmo. Assim a forma criada foi dividida em suas respectivas lajes e planificada, tomando-se o cuidado de estruturar dois volumes inseridos no modelo, para representar as circulações verticais. As peças foram dimensionadas para serem reproduzidas em uma cortadora a laser em uma escala de 1:100, como resultado da planificação e montagem temos a figura 10, a estrutura de um edifício com 17 lajes e 50m de altura.



Figura 10: Modelo físico
Fonte: Mariano P.O. P. (2017).

Resultados e discussões

O algoritmo desenvolvido para o experimento cumpriu as propostas averiguadas. A principal, a respeito da sua capacidade de se gerar formas complexas, apresentou um resultado potencialmente satisfatório, produzindo uma gama de formas que poderiam não ser concebidas a partir dos conhecimentos geométricos euclidianos sem a aplicação de uma ferramenta digital. O código proporcionou uma gama de

opções formais para construções de designs verticais, que podem ser justificadas por uma série de parâmetros desenvolvidos.

O resultado apresentado se mostrou satisfatório pois o código desenvolvido foi posto em teste, modificando-se e verificando-se os parâmetros de modo generativo, a fim de se constatar os diversos modelos plásticos criados pela ferramenta.

A programação generativa demonstra que caso ela seja desenvolvida aliada a códigos de simulação pode resultar em formas que exprimem a capacidade máxima do edifício para algumas características, como desenvolver uma forma que capte a maior quantidade de radiação, podendo-se assim otimizar a utilização de painéis fotovoltaicos na fachada, ou se distribuir as aberturas de maneira coerente com esta medição. Algumas interpretações negativas a respeito do trabalho também foram perceptíveis, sendo a demora para a o desenvolvimento e geração da forma topológica um empecilho no decorrer dos testes da programação. A capacidade de processamento dos computadores para o desenvolvimento de algoritmos com determinada complexidade, é uma característica importante a ser considerada, pois além de um conhecimento prévio e domínio dos programas utilizados, a capacidade de hardware do computador pode influenciar negativamente ou limitar o desenvolvimento da metodologia.

Agradecimientos

Gostaríamos de agradecer ao CAPES pelo suporte durante o desenvolvimento do trabalho, a Universidade Federal de Santa Catarina – UFSC, em especial aos laboratórios Hiperlab e Pronto 3D, por fornecerem os ambientes e ferramentas necessários para a construção deste artigo.

Referencias

ALVEZ Gilfranco Medeiros; PRATSCHKE, Anja. Processos de Criação, Emergência e Parametrização em Arquitetura. Cadernos de Pós-Graduação em Arquitetura e Urbanismo. V. 12 n. 1. 2012.1 ISSN 1809-4120.

- BARRIOS, Carlos Roberto, Hernandez. Parametric Gaudi. VIII Congresso Ibero americano de gráfica digital – SIGraDI. Universidade do Vale dos Sinos, São Leopoldo, Brasil. 2004
- BONA. Cristiana. Avaliação de processo de software: um estudo de caso em XP e Iconix. Florianópolis 2002, Dissertação
- CANTOR, Murrey R. Object-Oriented Project Management with UML. New York: Ed. John Wiley & Sons, 1998. Cap.03, p.93-95
- CELANI, Gabriela. VAZ, Carlos Eduardo Verola. CAD Scripting And Visual Programming Languages For Implementing Computational Design Concepts: A Comparison From A pedagogical Point Of View. International Journal of Architectural Computing. Volume 10, 2012. Pags. 121-137. ISSN 1478 0771.
- DANTAS, R. Jorge. The End of Euclidean Geometry or Its Alternate Uses in Computer Design. SIGRADI 2010, disrupcion, modelacion y construccion: Diálogos cambiantes. De 17 A 19 de Novembro Bogota, Colombia, Universidade de los Andes.
- DANTAS, R. Jorge. The End of Euclidean Geometry or Its Alternate Uses in Computer Design. SIGRADI 2010, disrupcion, modelacion y construccion: Diálogos cambiantes. De 17 A 19 de Novembro Bogota, Colombia, Universidade de los Andes.
- ERCAN, Burk. ELIAS-OZKAN, Soofia Tahira. Performance-based parametric design explorations: A method for generating appropriate building components. Design Studies. Vol. 38, 2015. Pags. 33-53. ISSN: 0142-694X.
- FISHER, Thomas. HERR, Christiane M. Teaching Generative Design. Design Research Technology Centre, 2000.
- GÜRBÜZ, Esra. ÇAGDAS, Gülen. ALAÇAM, Sema. A generative Design Model for Gaziantep's Traditional Pattern. Future cities: 28th eCAADe Conference Proceedings. Zurich, Switzerland: ETH Zurich, 2010
- RASHAD, Ahmed. ALFARIS, Anas. A Performance Based Generative Design System Methodology for Sustainable Design in Practice. Conference On Technology & Sustainability in the Built Environment. Kings saud university – College of Architecture and Planning. 2010.
- RIAM, M. Iasef. ASAYAMA, Shuichi. Computational Design of a nature-inspired architecture structural using the concepts of self-similar and random fractals. Automation in Construction Vol. 66 2016. Pags. 43 – 58. ISSN: 0926-5805
- SALIM, Flora Dilys. BURRY, Jane. Software Openness: Evaluation Parameters of Parametric Modeling Tools to Support Creativity and Multidisciplinary Design Integration. ICCSA International Conference Computational Science and Its Applications, Fukuoka, Japan, março de 2010. Parte III. Pags. 483-497. 2010.