

**DESIGN E TECNOLOGIA: TRANSPOSIÇÃO DE VOLUMETRIAS ENTRE
SUPERFÍCIES POLIGONAIS E POLYSURFACES NURBS**

***DESIGN AND TECHNOLOGY: VOLUMETRIC TRANSPOSITION BETWEEN
POLYGONAL SURFACES and NURBS POLYSURFACES***

Felix Bressan¹

Wilson Kindlein Júnior²

Fábio Gonçalves Teixeira³

Resumo

O presente estudo se divide em dois caminhos construtivos: conversão de volumetrias constituídas de superfícies poligonais (*quads*) em superfícies *NURBS*; e conformação de padrões ou outras entidades poligonais quaisquer sobre essas volumetrias. Existem inúmeros casos onde a topologia destino não pode ser convertida para somente uma superfície *NURBS* inteira, procedimento que facilitaria enormemente a tarefa de transposição de volumetrias e que já estaria resolvido com o emprego de ferramentas prontas de modelagem paramétrica. Para executar as tarefas deste artigo, foram adotadas estratégias de sistemas de modelagem generativa e parametrizações associativas. Os conhecimentos de construção de sistemas concebidos para a presente pesquisa poderão ser empregados em campos distintos que possuam, como paradigma, a criatividade projetual como uma de suas balizas, como, por exemplo, as Artes Visuais, a Arquitetura e *Design*. O presente trabalho visa possibilitar a utilização da fabricação digital através das tecnologias de corte a laser, impressão 3D ou usinagem CNC, ampliando estratégias e conceitos referentes à pesquisa de superfícies e construção de volumetrias de objetos através da busca de inovação das suas possibilidades expressivas.

Palavras-chave: *design* e tecnologia; estratégias generativas; superfície; volumetria; escultura; design de superfície.

Abstract

The present study is divided in two constructive ways: conversion of volumetric bodies constituted by polygonal surfaces (*quads*) into surfaces *NURBS* and conformation of patterns or other polygonal entities on such volumetric bodies. There are numerous cases where the target topology cannot be converted to a single and unique *NURBS* surface, a procedure that would make the task of volumetric transposition easier and which would already be solved by the use of ready made parametric modeling tools. To carry out this article's tasks, generative modeling systems and associative parametrization strategies were adopted. The knowledge for the construction of the systems used in the present research can be used in different fields that have, as a paradigm, the design creativity as one of its beacons, such as Visual Arts, Architecture and Design. The present work aims to make use of digital manufacturing through laser cutting, 3D printing or CNC milling technologies, at the same time expanding strategies

¹Doutorando, Programa de Pós-Graduação em *Design* – UFRGS, felixbressan@gmail.com

²Professor Doutor, Departamento de Materiais da Escola de Engenharia – UFRGS, wilsonkindleinjuniorgmail.com

³Professor Doutor, Departamento de *Design* e Expressão Gráfica – UFRGS, fabiogt@ufrgs.br

and concepts related to surface research and volumetric object construction through the search for innovation of its expressive possibilities.

Keywords: design and technology; generative strategies; surface; volumetric; sculpture; surface design.

1. Introdução

A evolução e consolidação da computação gráfica para aplicação no design de produtos, na engenharia, na arquitetura e nas artes permitiu a geração de ferramentas que tornam as possibilidades de modelagem geométrica praticamente infinitas. A modelagem paramétrica, onde são definidas regras e controles para a geração de geometrias, trouxe ferramentas onde um processo de programação visual orienta a modelagem geométrica. É neste contexto que este trabalho foi desenvolvido. O objetivo é gerar volumetrias complexas a partir do mapeamento de um volume 3D sobre outro volume 3D, similar ao mapeamento de texturas. No entanto, o processo aqui proposto é muito mais complexo pois envolve o mapeamento em dois contextos tridimensionais, ao contrário do mapeamento de texturas que realiza um mapeamento 2D para 3D.

As estratégias desenvolvidas neste trabalho possibilitam construir volumetrias sobre objetos formados por superfícies poligonais conformando padrões tridimensionais ou outros objetos poligonais quaisquer através de modelagem paramétrica e processos generativos digitais. Isso permite a criação de uma ampla gama de objetos que podem ser inteiros ou vazados, podem ser padrões repetidos em qualquer sistema de repetição ou ainda objetos aleatórios quaisquer. É importante ressaltar que o conceito desenvolvido visa gerar uma configuração de objetos pensada para a criação de uma nova volumetria, que pode sobrepor a volumetria destino ou substituí-la, ou gerar uma distribuição interna à volumetria sempre acompanhando a forma da mesma. Poderíamos supor que seria uma textura aplicada a um objeto. Mas este trabalho vai além disso. Estamos gerando objetos que não são somente uma texturização 3D, mas a construção de um novo objeto com forma distinta.

Sobre o estado da arte, existem várias possibilidades de criação de texturas e assemelhados em volumetrias. Como, por exemplo, técnicas que utilizam canais *alpha*, *displacement*, *softwares* como Zbrush, 3ds Max, entre outros. Porém, esses sistemas não executam o que é proposto neste trabalho. Nessas técnicas, a superfície da volumetria destino é conformada segundo a forma do módulo e, no processo descrito neste artigo, é gerada uma cópia fiel ao modelo original, distribuída sobre a volumetria destino. Isso pode ser muito importante em casos, em que serão transpostos objetos que não podem ter sua geometria alterada, porque já estão elaborados para serem posteriormente planejados para corte a laser, por exemplo. Em relação a distribuição direta de objetos poligonais (*meshes*) sobre superfícies poligonais, como apresentado nesse artigo, não foi encontrado até o momento, procedimentos que apresentam o mesmo resultado, porém existem possibilidades parecidas. Uma delas é a distribuição de objetos *mesh* unitariamente em cada polígono na superfície destino. Fica-se assim limitado ao tamanho dos polígonos. E temos de considerar que os ângulos entre polígonos podem estar acentuados, o que ocasionaria uma quebra de continuidade dependendo do modelo. Isso não ocorre no método apresentado aqui pois além de não ficar limitado a polígonos individuais os mesmos são convertidos em superfícies *NURBS* (*Non Uniform Rational Basis Spline*) sistema de modelagem muito utilizado na indústria e que permite alcançar uma grande precisão (ISSA, 2013; MURDOCK, 1012). Isso origina uma superfície contínua entre polígonos vizinhos, conforme descrito posteriormente. Temos

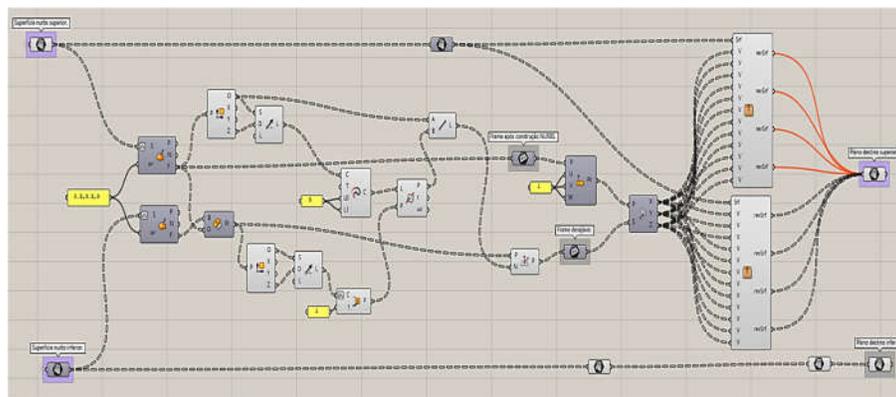
também comandos como ou semelhantes ao *Scatter* do *software* 3ds Max. Eles distribuem objetos, mas sem conformá-los e muitas vezes, os elementos se transpassam na distribuição. O procedimento mais próximo são ferramentas nativas encontradas como no *plug-in Grasshopper* do *software* *Rhinoceros*, que distribui objetos poligonais sobre superfícies destino confeccionadas de somente uma superfície *NURBS*. É um excelente método com ótimos resultados, porém, como não é possível gerar uma grande gama de objetos com somente uma superfície *NURBS*, ficaríamos circunscritos a objetos construídos dessa forma. Já com superfícies poligonais no objeto destino, podemos desenvolver uma maior variedade de formas. Como ponto negativo do presente trabalho, podemos dizer que não é em qualquer superfície *mesh* que podemos aplicar os procedimentos aqui descritos. A superfície não pode ter partes sobrepostas e ser formada de polígonos de quatro lados. Como regra geral, sua topologia tem de permitir que seja planificada mesmo com algumas distorções como descrito posteriormente. E outro ponto negativo, decorrente disso, é que: como a superfície, em muitos casos, será planificada com distorções dimensionais, os objetos distribuídos também vão apresentar alguma alteração de dimensão em relação ao seu formato original.

2. Materiais e Métodos

Para o desenvolvimento dos algoritmos utilizados neste trabalho, foi usado um sistema de modelagem paramétrica denominado *Grasshopper* que é uma linguagem de programação criada em 2007 por David Rutten, na empresa *McNeel & Associates*. O *Grasshopper* é um *plug-in* que trabalha sobre a plataforma do *software* *Rhinoceros*. A escolha dessa linguagem de programação está cada vez mais presente em vários trabalhos publicados atualmente. Existem outras linguagens de programação, mas a escolha do *Grasshopper* deve-se, entre outros motivos, à possibilidade de desenvolver códigos de programação de forma "visual", ao contrário de outras possibilidades em que seria necessário escrever os códigos literalmente. Porém, se for necessário, é possível desenvolver ferramentas não encontradas nativas no *software* através das linguagens de programação *script*, dentro do próprio *plug-in*.

Na Figura 1 vemos como exemplo uma parte do algoritmo desenvolvido. Esta imagem serve somente para ilustrar como se apresenta o desenvolvimento do algoritmo de forma visual. São pequenas "caixas" que contém os códigos de programação que vão sendo distribuídos na tela conforme a necessidade. As linhas tracejadas são as ligações executadas pelo programador ligando o resultado dos procedimentos com os próximos componentes.

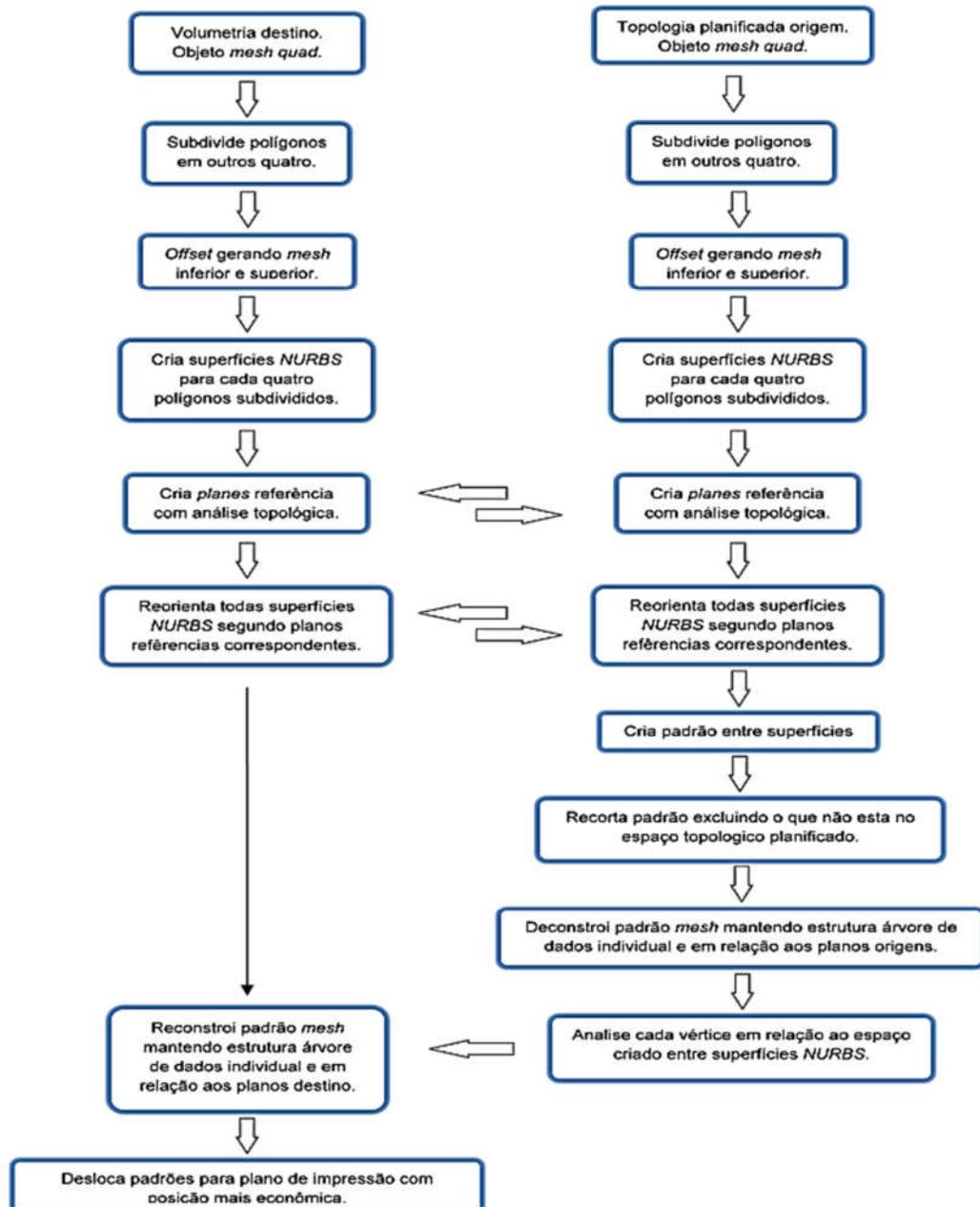
Figura 1: Exemplo de programação no *plug-in grasshopper*. Pequenas "caixas" que contém os códigos de programação são ligadas nas suas saídas de dados aos próximos componentes.



Fonte: Elaborado pelos autores.

O algoritmo completo desenvolvido para executar as tarefas deste artigo segue o fluxograma da Figura 2, onde todos os procedimentos desenvolvidos estão descritos para possibilitar a reprodução dos resultados utilizando-se qualquer linguagem de programação. É importante frisar que toda tarefa ocorre automaticamente do início ao fim somente sendo necessário ajustar parâmetros como, por exemplo, quantidade de módulos, espessura final da peça etc.

Figura 2: Fluxograma do algoritmo desenvolvido para possibilitar a reprodução dos resultados através de qualquer linguagem de programação.



Fonte: Elaborado pelos autores.

Basicamente todo o procedimento pode ser resumido da seguinte forma: Começamos com módulos poligonais ou formas poligonais que serão distribuídas sobre uma superfície destino que é a forma final do trabalho. A partir da superfície destino é feita uma cópia totalmente planificada da mesma mas, topologicamente, idêntica. É como se achatássemos todos seus polígonos sem nenhuma sobreposição num formato que lembra uma folha de papel recortada. Tanto a superfície planificada como a superfície destino são convertidas para várias superfícies *NURBS*, o que gera uma continuidade harmônica em ângulos acentuados que poderiam existir. Os objetos que serão distribuídos na superfície destino são inicialmente distribuídos sobre a superfície planificada e recortados segundo seus limites. Todos os elementos que compõem os objetos ou módulos são analisados em relação a sua localização nas superfícies *NURBS* planificadas individualmente. Na superfície destino é criado um espaço gerando-se outra superfície sobreposta que seria a espessura do trabalho final. Nesse espaço, considerando a análise dos elementos anterior, são reconfigurados os módulos poligonais finalizando o trabalho.

2. Modelo Poligonal e Topologia Planificada

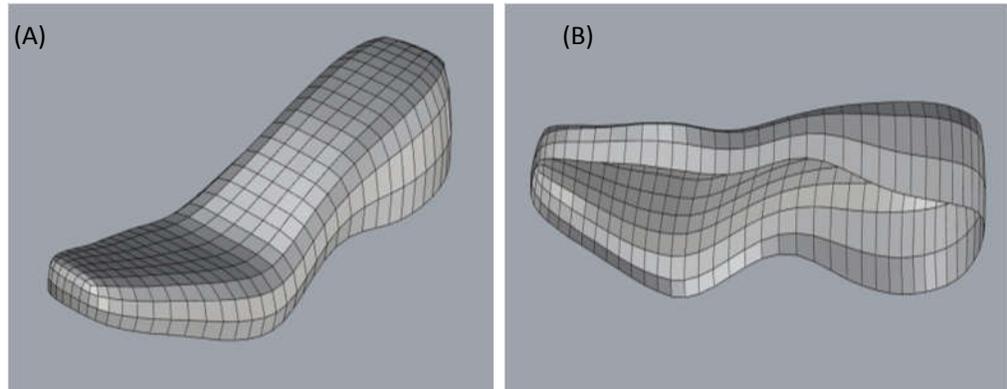
Os procedimentos aqui descritos podem ser aplicados numa gama grande de objetos poligonais com os mais diversos fins. Não somente objetos de *design*. Para este trabalho, desenvolveu-se um modelo de calçado referenciado no trabalho de Cardoso (2009). A escolha desse modelo ocorreu por ele já ter sido estudado em seus aspectos de planificação e por ser um produto utilizado na indústria. Ela descreve procedimentos para mapeamento e estampagem de imagens sobre superfícies não planificáveis aplicadas a objetos de *design*. São analisados procedimentos técnicos para configurar imagens sobre objetos e são estudadas superfícies planificáveis e não planificáveis, para um mapeamento com pouca distorção usado na fabricação de produtos. Usaremos um modelo de estudo semelhante, que também não pode ser planificado sem ter distorções, pois ele teria de ter partes seccionadas.

Podemos citar duas diferenças básicas apresentadas no presente artigo. A primeira é que trabalharemos com construção de volumetrias por modelagem poligonal e não com mapeamento de imagens sobre a volumetria. E o segundo diferencial é o desenvolvimento de estratégia de conversão do objeto destino para superfícies *NURBS*, o que viabiliza algumas vantagens como termos uma superfície sem necessidade de aumentarmos enormemente a quantidade de subdivisões de polígonos. Nesse modelo a sua superfície superior e lateral não podem ser planificadas sem distorção, o que é um ótimo exemplo para esta aplicação. O modelo poligonal foi desenvolvido com polígonos de quatro lados, técnica muito utilizada na construção de *meshes* (AZEVEDO e CONCI, 2003; GOMES e VELHO, 2008). Se o modelo original fosse modelado diretamente em *NURBS* ou se já existisse nessa técnica, ele seria muito provavelmente constituído de *polysurfaces*. Também nesse caso poderiam ser executados facilmente os procedimentos aqui descritos, mas não por outras técnicas existentes com as mesmas características abordadas.

O ideal é que os objetos fossem formados por somente uma superfície *NURBS*, o que, em muitos casos, não é possível de ser executado. Encontramos muitos objetos utilizados na indústria com modelagem *NURBS*, mas feitos de várias *polysurfaces*. Isto é, são formados pela união de várias superfícies. Se o objeto for construído em varias superfícies *NURBS*, podemos convertê-lo para um objeto poligonal e posteriormente reconvertê-lo para *NURBS* na técnica aqui apresentada. Na Figura 3 vemos o modelo poligonal utilizado. Ele seria a representação do corpo de um calçado sem a parte inferior. Sobre as laterais e a parte superior é que seriam distribuídos os nossos módulos apresentados posteriormente. É importante frisar que essa

distribuição de módulos será contínua sem interrupções nas passagens do topo para as bordas laterais.

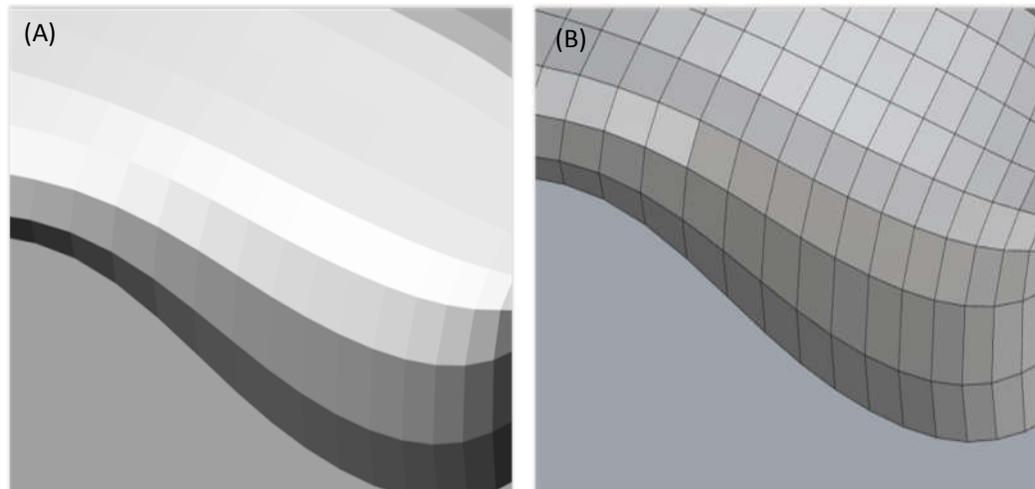
Figura 3: Modelagem poligonal do protótipo. (A) Vista Perspectiva. (B) Vista lateral/inferior.



Fonte: Elaborado pelos autores.

Vemos na Figura 4 os polígonos constituintes do objeto destino. Suas faces não tem uma continuidade satisfatória para a execução deste trabalho. Foi então desenvolvido um método de conversão desses polígonos individualmente para *NURBS*. Cabe aqui uma ressalva; eles poderiam ser facilmente convertidos mantendo-se exatamente sua forma e, por conseguinte, com ângulos acentuados entre eles. Mas o que será apresentado é a sua conversão, deixando a superfície perfeitamente lisa, com leve arredondamento nos seus limites, sem quebras visuais.

Figura 4: (A) Detalhe da malha poligonal do objeto destino demonstrando os ângulos acentuados entre polígonos vizinhos. (B) Detalhe da malha poligonal com *edges* a mostra.

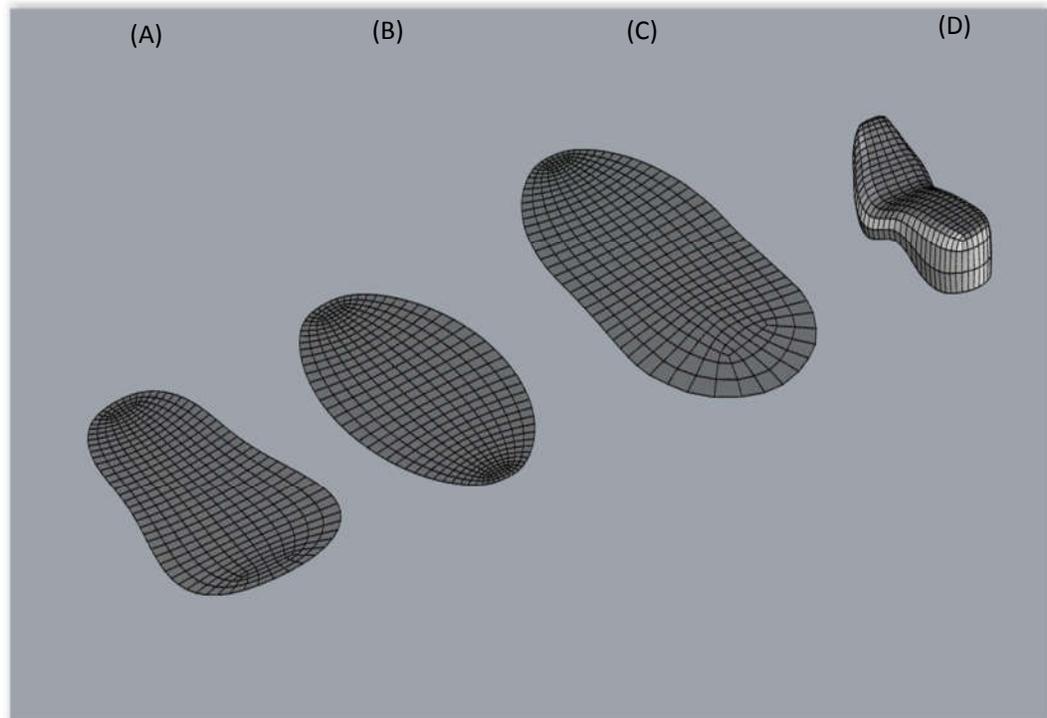


Fonte: Elaborado pelos autores.

Essa superfície destino foi totalmente planificada utilizando-se o sistema de geração de mapas planificados do *software* 3ds Max. Na realidade, foram implementadas três possibilidades de topologias planificadas do modelo inicial. Todas elas são equivalentes topologicamente e poderiam ser utilizadas. O ideal é que cada polígono da superfície planificada em relação ao seu equivalente no objeto original tivessem as mesmas dimensões proporcionalmente para não ocasionar distorções. Quanto mais forem parecidas as dimensões dos polígonos planificados com seus equivalentes proporcionalmente, maior será a precisão da reprodução.

O modelo planificado, em princípio, pode ter qualquer variação de forma. Na Figura 5, vemos três possibilidades topológicas de superfícies planificadas geradas a partir do mesmo modelo poligonal inicial. Se fosse possível termos no modelo em 3D as mesmas dimensões dos polígonos equivalentes do mapa planificado, não teríamos nenhuma distorção dimensional na transposição dos *meshes* entre as duas superfícies. Mas como numa grande quantidade dos casos será impossível ter essa equivalência de dimensão, vai haver uma variação de escala no resultado final. A variação aqui, ocorrerá em relação à diferença de tamanho entre polígonos equivalentes do modelo planificado e seu correspondente no volume destino.

Figura 5: (A), (B) e (C) Três possibilidades de superfícies planificadas correspondentes à superfície original. Qualquer uma delas poderia ser utilizada, topologicamente são iguais mesmo com formas e dimensões diferentes. (D) Volumetria destino.

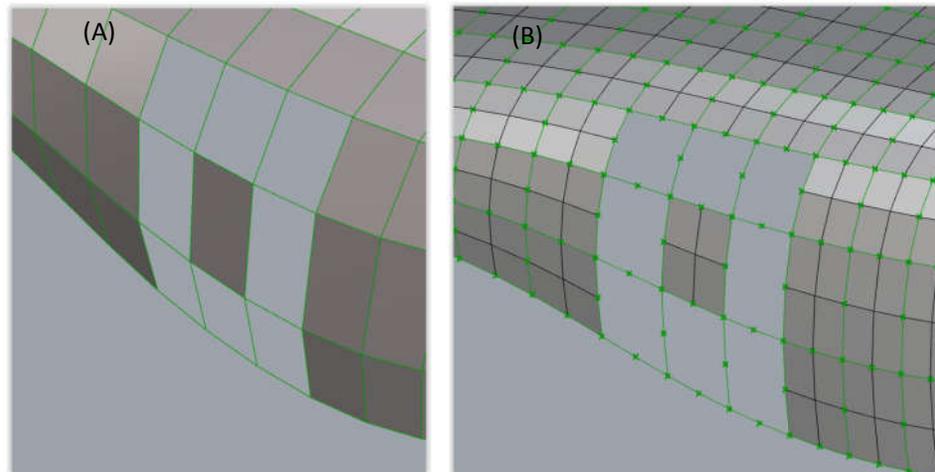


Fonte: Elaborado pelos autores.

2.1. Conversão de Volumetrias para NURBS

Será apresentada aqui a estratégia de conversão da topologia planificada e da volumetria destino para *NURBS* convertendo cada polígono individualmente. Essa técnica divide cada polígono em múltiplos de quatro. As divisões ocorrem "arredondando" os polígonos e não simplesmente subdividindo os mesmos. Este algoritmo é referenciado no trabalho de Catmull e Clark (1978) e encontra-se em outro *plug-in* instalado dentro de próprio *Grasshopper* que se chama *Weaverbird*. A ferramenta é *Weaverbird's Catmull-Clark Subdivision*. Se os polígonos fossem triangulares também poderíamos convertê-los para *NURBS*, mas não teríamos o ganho de suavização entre superfícies que teremos aqui. Na Figura 6, vemos os polígonos originais e depois de subdivididos. Na imagem, foram retirados alguns polígonos para melhor visualização do processo.

Figura 6: Exemplo de polígonos antes e após a divisão. (A) Polígonos originais. (B) Polígonos após a divisão pelo algoritmo *Weaverbird's Catmull-Clark Subdivision*.

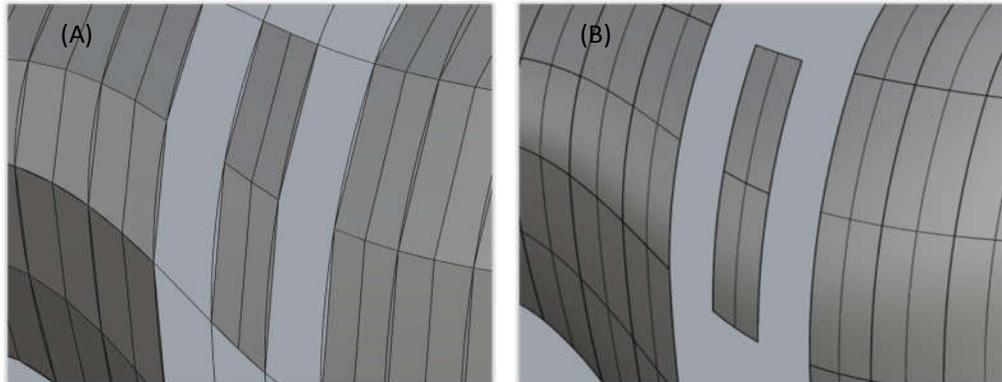


Fonte: Elaborado pelos autores.

Com a divisão dos polígonos e por consequência o aumento do número de vértices é possível convertê-los em curvas *NURBS* e com isso gerar as superfícies *NURBS*. Na Figura 7, vemos os polígonos após a divisão e as curvas *NURBS* geradas a partir de seus pontos de controle e a criação das superfícies *NURBS* a partir das curvas *NURBS*.

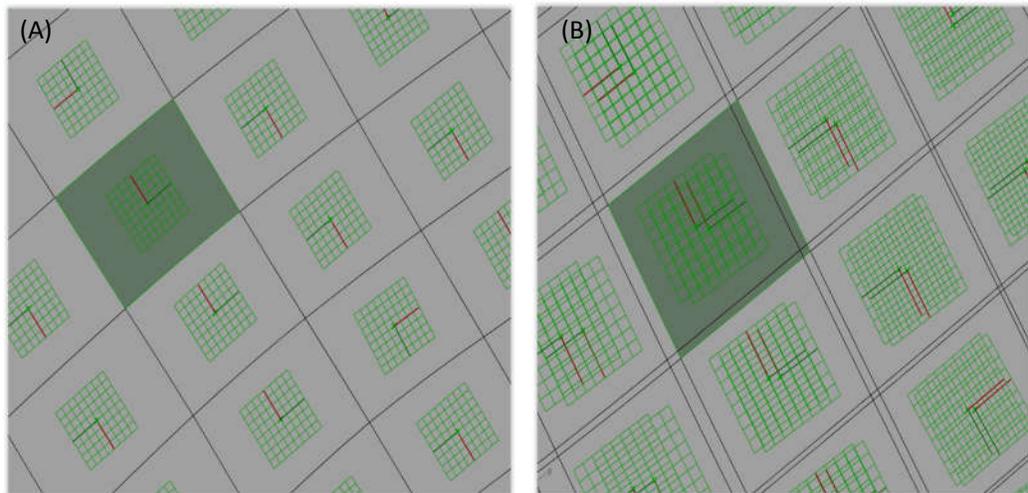
Para todo o processo funcionar adequadamente, todas as superfícies individuais equivalentes devem estar perfeitamente alinhadas. Temos a superfície plana e a superfície destino formadas de várias superfícies individuais independentes, que foram originadas de seus respectivos polígonos individuais. Então a mesma superfície correspondente deve ser equivalente nas suas coordenadas *UVW*, que são coordenadas equivalentes às coordenadas tridimensionais *XYZ*, mas utilizadas em mapas 2D (MURDOCK, 2012). Como o mapeamento 2D não tem a profundidade representada pelo eixo *Z*, a coordenada *W*, sua equivalente, serve para mostrar a orientação, nesse caso, se é para dentro ou para fora do objeto. Essas coordenadas são representadas visualmente por um *frame*. Na Figura 8, vemos algumas superfícies das topologias. Observa-se que os frames das superfícies equivalentes estão exatamente na mesma posição em todas as superfícies individualmente após a aplicação do algoritmo.

Figura 7: (A) Subdivisão dos polígonos em outros quatro e curvas NURBS. (B) Conversão para superfícies NURBS a partir das curvas NURBS.



Fonte: Elaborado pelos autores.

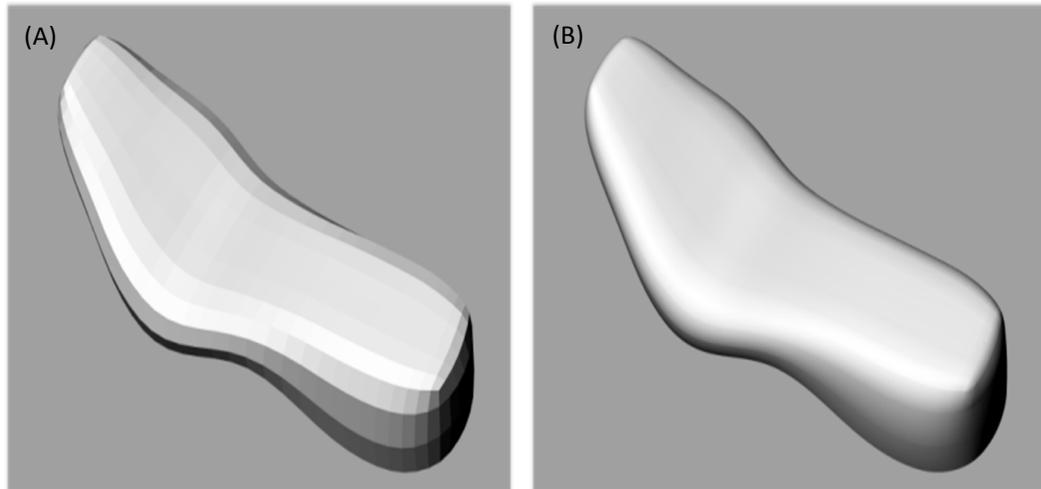
Figura 8: (A) Frames nas superfícies individuais. (B) Frames das superfícies equivalentes estão na mesma posição em todas as superfícies individualmente após a aplicação do algoritmo.



Fonte: Elaborado pelos autores.

Na Figura 9, vemos a volumetria destino original com modelagem poligonal e após a sua conversão para várias superfícies *NURBS*. Observa-se que o modelo ficou sem ângulos acentuados. Apesar de não ser aparente o modelo final em *NURBS* é composto de 434 superfícies *NURBS* individuais, por isso chamado de *polysurfaces*.

Figura 9: (A) Volumetria original confeccionada com modelagem poligonal. (B) Mesma volumetria convertida para várias superfícies NURBS (434 superfícies individuais formam esse modelo).

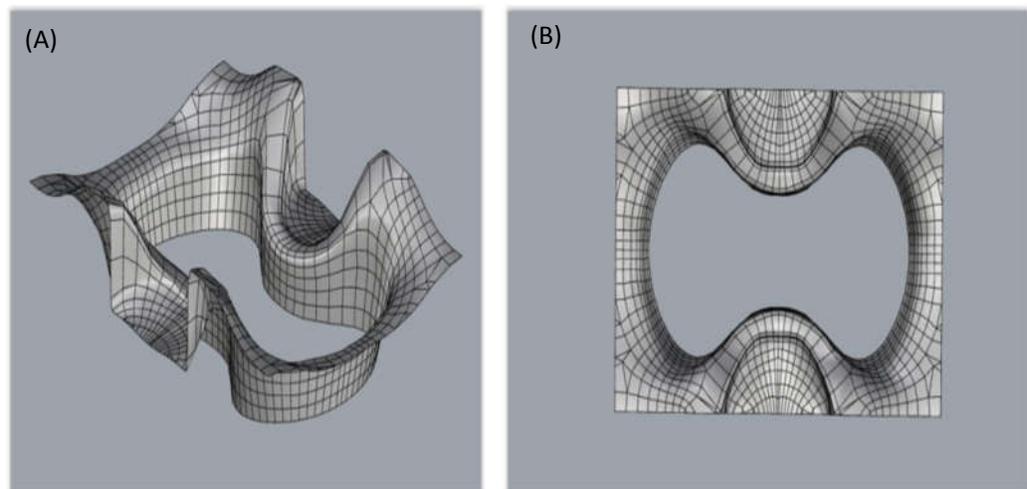


Fonte: Elaborado pelos autores.

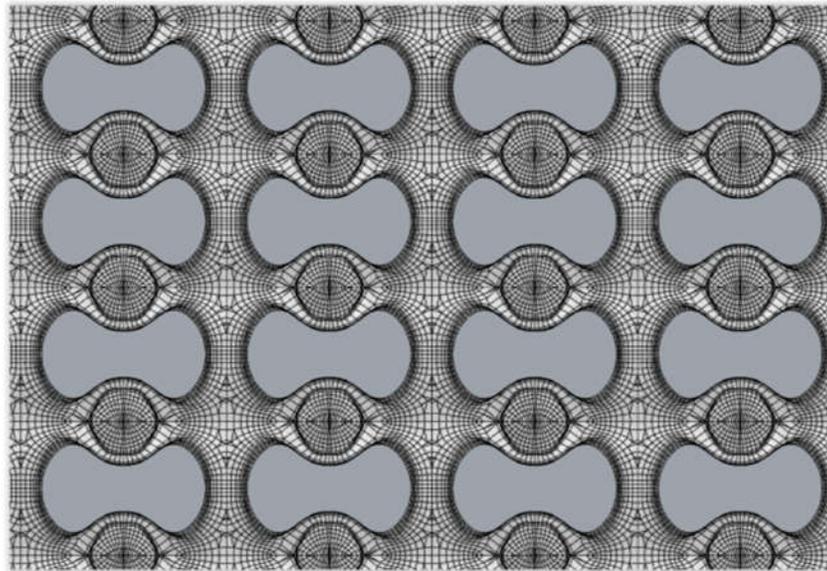
2.2. Recorte da Volumetria inicial Segundo Topologia Planificada

Na Figura 10, vemos o módulo poligonal utilizado como padrão para geração de uma volumetria distribuída sobre a superfície plana através da sua repetição (Figura 11). Cabe ressaltar que, para esse trabalho, poderia ser qualquer outra forma poligonal, não necessitando serem módulos. O fator limitante é a quantidade de polígonos, o que pode tornar lento o trabalho dependendo das configurações de *hardware*. A espessura dessa volumetria original não é muito importante, pois o próprio algoritmo a adequa à espessura pré-configurada da volumetria final.

Figura 10: Objeto poligonal que será utilizado como módulo de repetição. (A) Módulo em perspectiva com polígonos aparentes. (B) Módulo vista superior.



Fonte: Elaborado pelos autores.

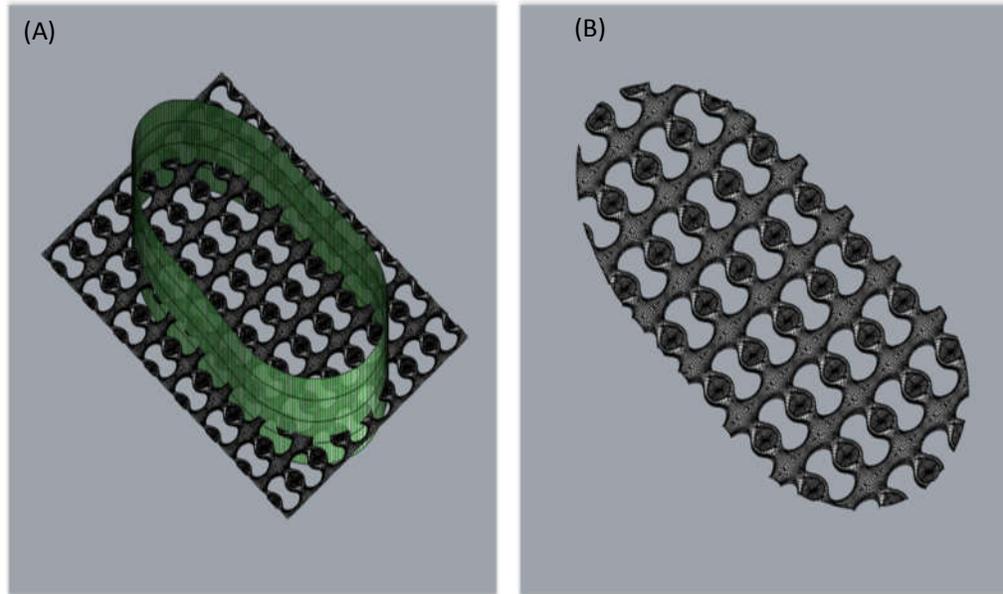
Figura 11: Detalhe dos módulos repetidos gerando um grande *mesh*.

Fonte: Elaborado pelos autores.

Após termos gerado o *mesh* como se fosse uma padronagem, temos de excluir tudo que não está contido no espaço gerado pela topologia planificada. Todo o princípio deste trabalho é fazer uma relação das formas contidas sobre o espaço plano composto da forma da superfície planificada e sua correspondente transposição para o espaço gerado na forma tridimensional destino. Esse conceito é a essência deste trabalho. Todos os procedimentos giram em torno disso. Foram experimentadas várias opções de operações *booleanas* para recortar os objetos. Operações *booleanas* são procedimentos que podem unir, subtrair ou realizar outras operações, entre duas formas na computação gráfica. A estratégia adotada foi a de "explodir" o *mesh*, ficando somente triângulos individuais, mas sempre com o cuidado em manter a estrutura de dados e a identificação individual dos módulos. Trabalhando com os triângulos do objeto - menor polígono possível - o recorte das formas foi facilitado e não apresentou defeitos. Com o contorno extrudado da superfície planificada seccionaram-se os polígonos. Verifica-se, em cada triângulo, o que está contido dentro da área formada pelas superfícies planificadas e exclui-se o restante (Figura 12). Temos então como resultado a nossa "textura" que será transposta acompanhando exatamente o formato da superfície planificada.

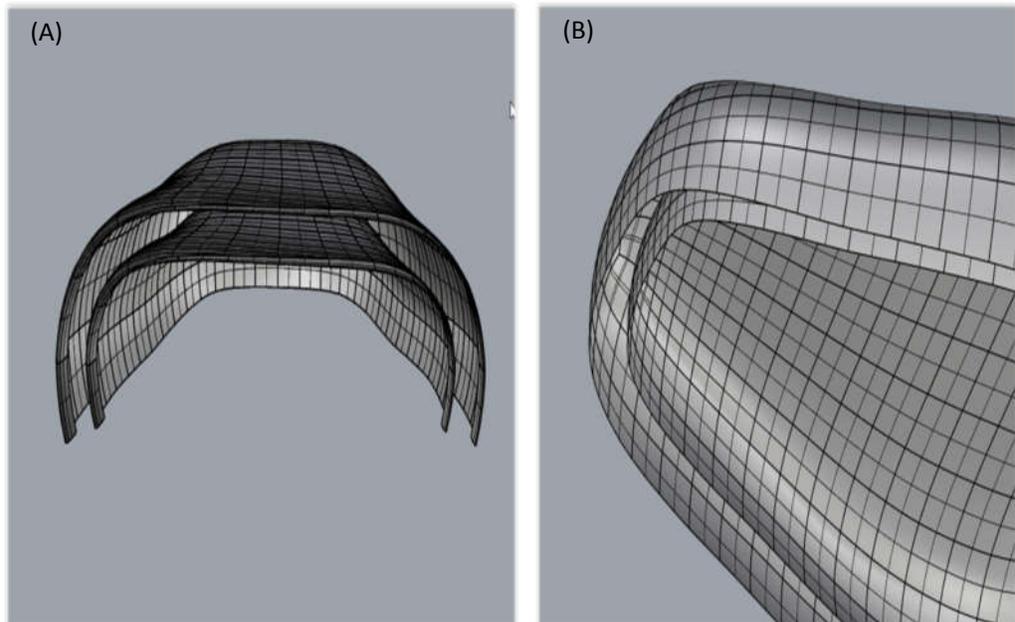
Na superfície destino foi aplicada uma extrusão formando um espaço vazio entre as duas superfícies. Esse espaço corresponde à espessura do objeto final e é onde as peças serão alocadas. Vemos na Figura 13 o espaço criado entre as superfícies destino com o objeto recortado para melhor visualização. Na realidade, a volumetria original com modelagem poligonal foi extrudada, formando outro objeto que sobrepõe o original. Após, esses dois objetos foram convertidos para *NURBS* formando duas volumetrias formadas de *polysurfaces*. Esse procedimento é muito importante pois deixa as junções entre todas as superfícies perfeitamente alinhadas, não ficando aparente a "emenda" entre as várias superfícies constituintes. Aplicar comandos como *offset*, ou outros, diretamente nas *polysurfaces* provocaria pequenos desalinhamentos, pois são várias superfícies e não somente uma inteira, comprometendo o trabalho final.

Figura 12: Estratégia adotada para executar a operação *booleana*. (A) Contorno extrudado da superfície planificada para cortar polígonos triangulares individuais. (B) Após o corte e separação das peças restando somente o que esta sobre a superfície planificada.



Fonte: Elaborado pelos autores.

Figura 13: Espaço gerado entre volumetrias convertidas para *NURBS*. O objeto foi cortado para melhor compreensão. Esse espaço seria a espessura do objeto final e é onde os módulos transpostos ficarão localizados. (A) Vista do centro para frente com o objeto recortado para permitir visualização do espaço formado. (B) Frente do objeto vista inferior.



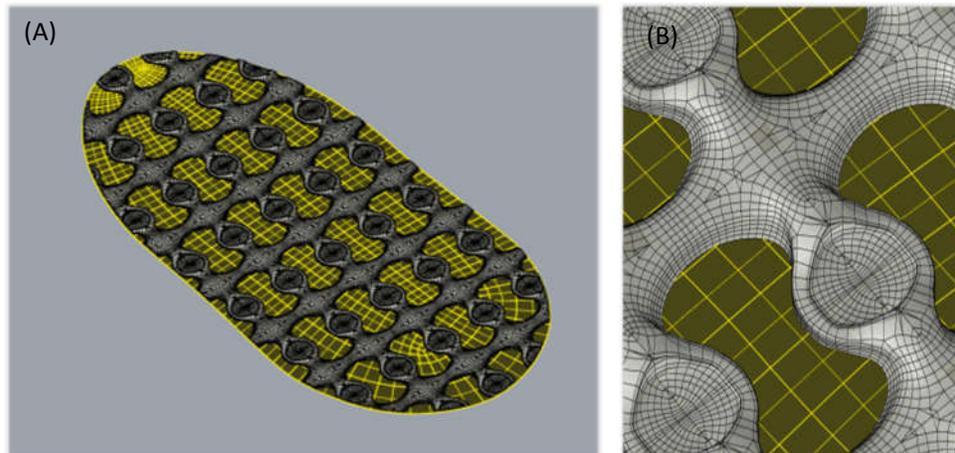
Fonte: Elaborado pelos autores.

2.3. Transposição Volumétrica.

O procedimento adotado para transposição pode ser resumido em separar a estrutura de dados do objeto *mesh* em relação a seus vértices e faces constitutivos. Após, esses vértices são analisados num espaço plano e transpostos para o espaço normalmente curvo entre as superfícies do objeto destino. Todo procedimento resume-se entre reconhecer a localização de cada vértice do *mesh* que será transposto sobre a topologia planificada e sua localização correspondente entre as topologias destino. Para isso o algoritmo faz uma análise complexa da árvore de dados sobre a localização em qual superfície cada vértice está e em que altura pré-configurada entre superfícies destino que ele se encontra. O objeto é decomposto em vértices e faces e reconstruído no novo espaço entre as superfícies destino. É uma tarefa que demanda um poder de processamento grande, mas depois de criado o algoritmo funciona perfeitamente.

Na Figura 14, vemos a volumetria formada de módulos *mesh* sobre a topologia planificada a partir do objeto destino. É importante ressaltar que foi escolhida uma das três topologias planificadas. Poderia ser qualquer outra. O que mudaria, seria a menor ou maior distorção entre proporções iniciais em relação à distribuição final.

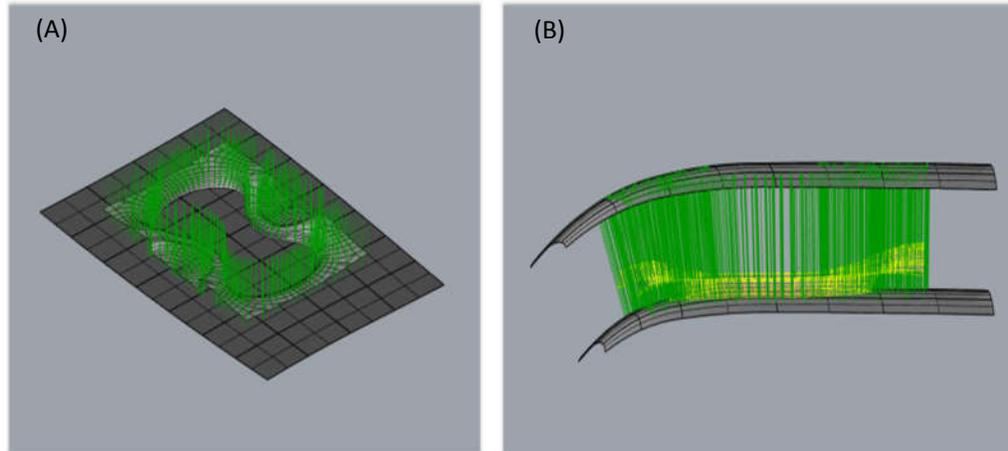
Figura 14: (A) Topologia planificada em amarelo com os objetos *meshes* recortados. (B) Detalhe.



Fonte: Elaborado pelos autores.

Na Figura 15 (A), vemos somente um dos módulos sobre as superfícies *NURBS* planificadas que esse módulo abrange. O procedimento que o algoritmo desenvolve é localizar cada vértice individualmente sobre uma das superfícies, sempre mantendo a árvore de dados em relação aos módulos. Isso é importante para a posterior reconstrução do objeto *mesh* em relação à sua estrutura de vértices e faces constituintes. É analisada também a localização da altura relativa dos vértices em relação à altura pré-configurada do espaço nas superfícies destino. E, na Figura 15 (B), vemos o módulo sendo transposto e recriado segundo o seu novo espaço envoltório. Vemos as linhas da altura em verde e, em amarelo, o módulo reconstruído com seus vértices correspondentes. As linhas verdes são geradas pela ligação da localização de cada vértice na superfície inferior e superior do espaço formado que será a espessura do objeto final.

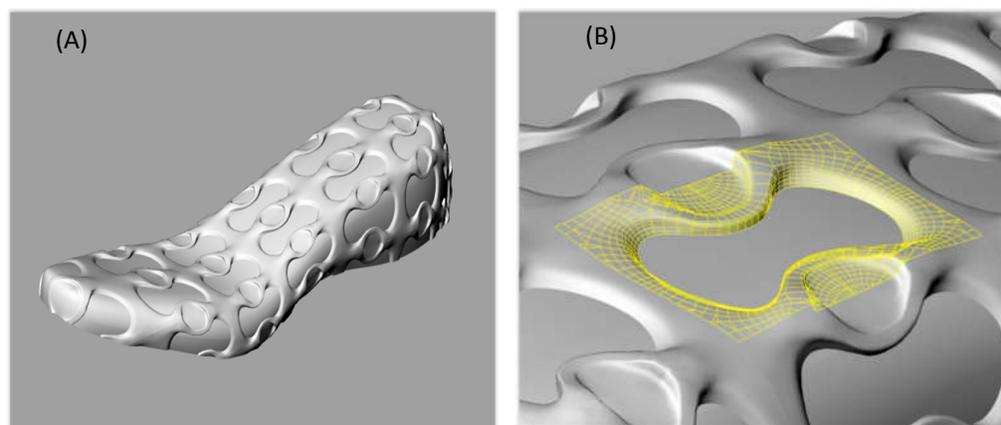
Figura 15: Localização dos vértices em relação à altura nas superfícies *NURBS* individuais. As linhas verdes são geradas pela ligação da localização de cada vértice na superfície inferior e superior. (A) um módulo na superfície planificada. (B) mesmo módulo em amarelo no espaço gerado entre as duas superfícies destino.



Fonte: Elaborado pelos autores.

E, finalmente na Figura 16, vemos o resultado final da transposição dos módulos sobre a superfície destino. É importante frisar que a localização da distribuição dos módulos não necessariamente seria sobre a superfície destino original. Poderia ser abaixo dela ou em qualquer posição relativa. Para isso seria necessário apenas mudar alguns parâmetros no algoritmo, desde que o resultado final acompanhe a volumetria original.

Figura 16: (A) Resultado final do trabalho conservando a volumetria original. (B) Detalhe de um módulo em amarelo para melhor visualização.

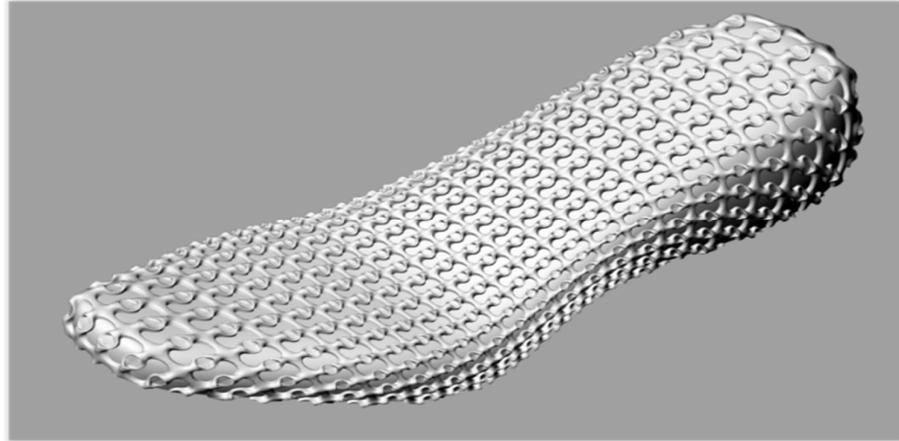


Fonte: Elaborado pelos autores.

Temos, na Figura 17, o mesmo trabalho da Figura 16, mas com maior quantidade de módulos. O algoritmo já prevê problemas de falta de memória RAM para a grande quantidade de polígonos. Com a diminuição do tamanho dos módulos em relação ao objeto destino, a quantidade de polígonos que o *software* tem de lidar ao mesmo tempo é muito grande.

Normalmente o *Grasshopper*, pela forma como foi concebido, trabalharia com todos os polígonos ao mesmo tempo. Então foi criada uma estrutura de *loop*, o que não é usual no *Grasshopper*, que calcula a transposição individualmente e já executa o *bake* (materialização na tela do Rhinoceros dos códigos do *Grasshopper*) e não acumula os dados na memória RAM.

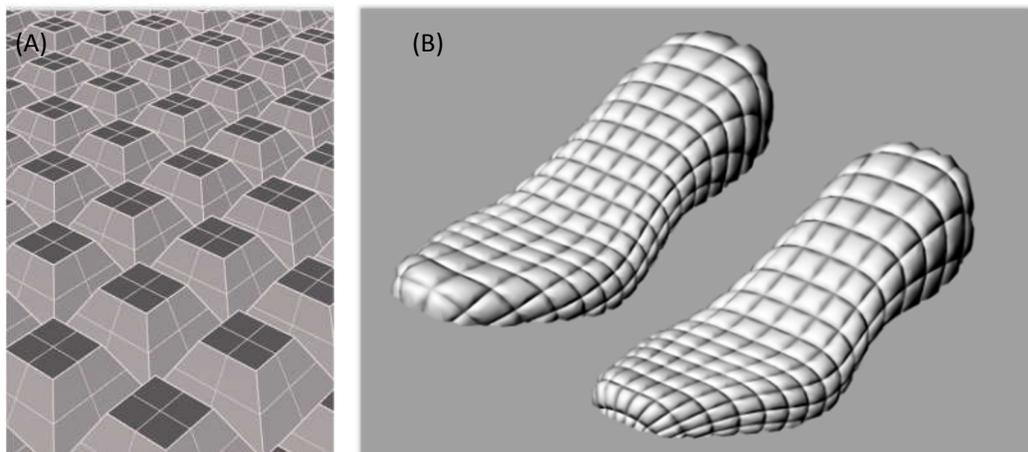
Figura 17: Mesmo módulo mas em quantidade bem superior.



Fonte: Elaborado pelos autores.

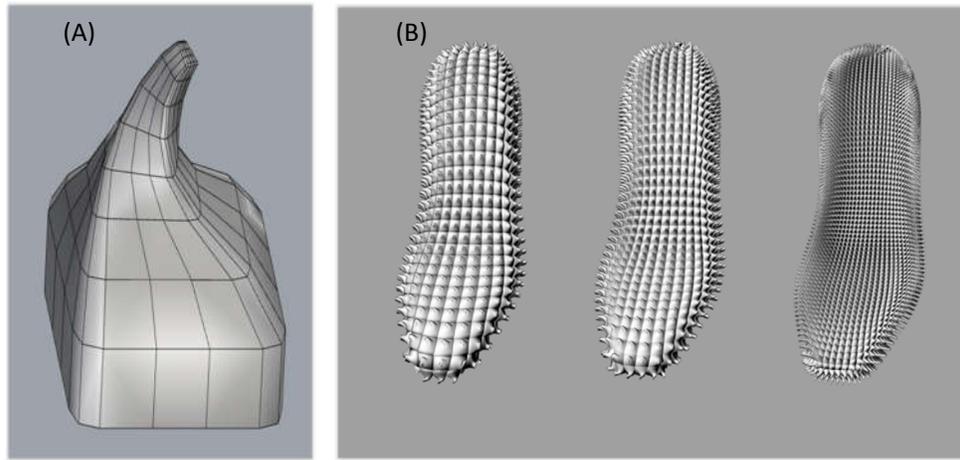
Para termos testes com diferentes *meshes*, foram executados outros trabalhos. Na Figura 18, vemos um módulo em forma de tronco de pirâmide, o que ajuda na visualização da forma distribuída sobre a volumetria destino. E, na Figura 19, um *mesh* mais exótico referenciado numa textura de jaca para confirmar a validade dos procedimentos adotados em situações bem distintas. Com um aumento ao extremo na quantidade de módulos foram testados problemas possíveis de memória e outros aspectos. Não discutimos aqui a validade de um "solado" com essa textura, que, no mínimo, teria de ser confeccionado com material flexível. Buscamos somente demonstrar a versatilidade possível na utilização desse algoritmo.

Figura 18: (A) Outro módulo *mesh* inicial. (B) Transposição com quantidades distintas de módulos.



Fonte: Elaborado pelos autores.

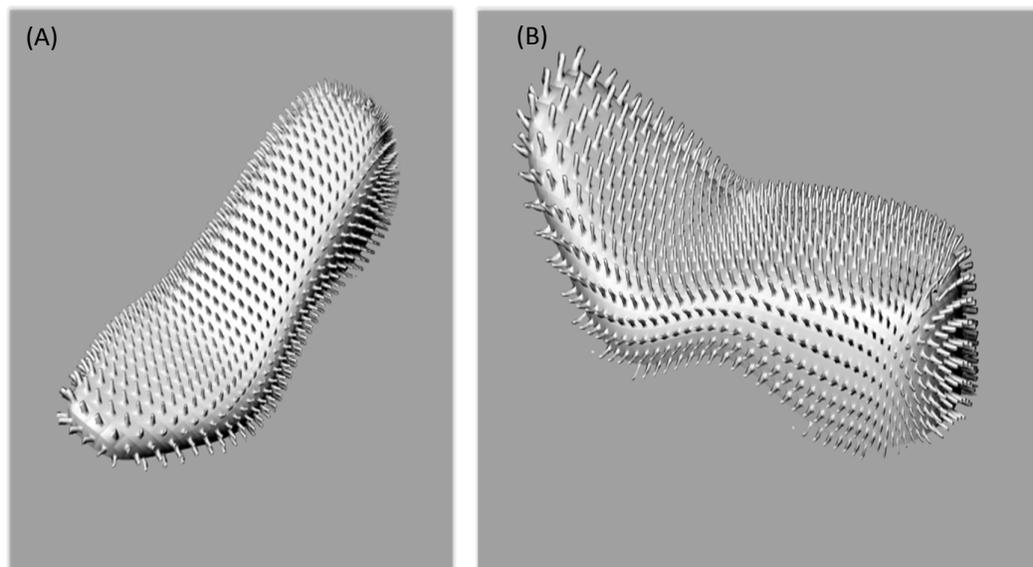
Figura 19: (A) Módulo Inicial. (B) Transposição com quantidades distintas de módulos.



Fonte: Elaborado pelos autores.

Na Figura 20, vemos um resultado diferente daquele da Figura 19, mas o módulo usado foi o mesmo. O que resultou na diferença foram mudanças de configurações na forma como a transposição entre alturas de superfícies dos objetos é tratada. O módulo fica com seu corpo mais achatado e só a haste final fica mais evidente. A modelagem paramétrica permite uma série de configurações, mudando facilmente o resultado final. E, novamente, sem discutir o mérito desse modelo em relação à sua usabilidade, o material de confecção teria necessariamente de ser flexível.

Figura 20: Alterando-se parâmetros das configurações na forma como o algoritmo trata a altura entre superfícies há uma variação no resultado da distribuição do módulo da Figura 19. (A) Perspectiva frente. (B) Perspectiva costas.



Fonte: Elaborado pelos autores.

3. Considerações Finais

Este artigo apresenta uma solução para problemas de transposição de volumetrias. Abrange casos quando não é possível converter *polysurfaces* ou superfícies poligonais em somente uma superfície *NURBS* sem grandes deformações. As estratégias desenvolvidas solucionam esse problema em uma gama grande de casos onde outros procedimentos não seriam possíveis. Isso apresenta-se como uma alternativa viável para uma série de objetos, mas nem sempre seria possível aplicar essas estratégias apresentadas. Dependendo da forma que for construída a volumetria destino, como, por exemplo, se tiver partes sobrepostas, estas teriam de ser resolvidas separadamente.

Foram estruturados vários procedimentos metodológicos através de estratégias generativas na criação de possibilidades inovadoras no desenvolvimento de superfície e volumetrias de objetos que poderão ter uso em áreas como o *design*, a arquitetura e as artes visuais. Vários estudos de processos técnicos foram necessários a fim de viabilizar as soluções desenvolvidas para o desenvolvimento de modelos 3D, visando a criação de protótipos adequados à fabricação digital. No desenvolvimento deste trabalho, muitas possibilidades foram surgindo. Evidentemente que é necessário definir objetivos e acentuar um foco para sistematizar o estudo, e isso foi feito, mas a vontade de ir além foi muito grande. Pode-se afirmar que os caminhos a serem percorridos estão no início e que esses estudos abriram possibilidades enormes de continuação e ampliação dos procedimentos que envolvem superfícies e volumetrias.

Referências

AZEVEDO, Eduardo; CONCI, Aura. **Computação Gráfica: Teoria e Prática**. Rio de Janeiro: Campus, 2003.

CARDOSO, Cilene Estol. **Desenvolvimento de um método de controle de distorções para aplicação em um problema de design de superfícies de formas tridimensionais não planificáveis**. Dissertação (Mestrado em Design). UFRGS, Porto Alegre, 2009.

CATMULL, E.; CLARK, J. Recursively Generated B-Spline Surfaces On Arbitrary Topological Meshes. *Computer Aided Design*, 10(6):350–355, 1978. Disponível em: <https://people.eecs.berkeley.edu/~sequin/CS284/PAPERS/CatmullClark_SDSurf.pdf> Acesso: Mar. 2017.

GOMES, Jonas; VELHO, Luiz. **Fundamentos da Computação gráfica**. Rio de Janeiro: IMPA, 2008

ISSA, Raja. **Essential Mathematics for computational Design**. Robert McNeel & Associates, 2nd Edition, 2013. Disponível em: <<http://www.rhino3d.com/download/rhino/5.0/essentialmathematicsthirdedition/>> Acesso: Nov. 2013.

MURDOCK, Kelly L. **Autodesk 3ds Max 2013 Bible**. Indianapolis, Indiana. Editora John Wiley & Sons, 2012.