

Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia Mecatrônica e de Sistemas
Mecânicos

ISSN 1517-3526
BT/PMR/0623

Algoritmo para Converter
Sólidos CSG em Sólidos B-REP

Murilo Antonio Salomão Garcia
Marcos de Salles Guerra Tsuzuki

Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

Diretor: Prof. Dr. Ivan Gilberto Sandoval Falleiros
Vice-Diretor: Prof. Dr. José Roberto Cardoso

Chefe do Departamento: Prof. Dr. Lucas Antonio Moscato
Suplente do Chefe do Departamento: Prof. Dr. Júlio César Adamowski

Conselho Editorial:

Prof. Dr. Lucas Antonio Moscato

Prof. Dr. Edison Gonçalves

Prof. Dr. Celso Massatoshi Furukawa

Prof. Dr. Nicola Getschko

Prof. Dr. Gilberto Francisco Martha de Souza

Prof. Dr. Marcelo Godoy de Simões

Prof. Dr. Marcos de Sales Guerra Tsuzuki

Coordenador Técnico:

Prof. Dr. Lucas Antonio Moscato

Esta é uma publicação da Escola Politécnica da USP/ Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, fruto de pesquisas realizadas por docentes e pesquisadores desta Universidade.

Iniciou-se a partir de outubro de 1999 a publicação do boletim técnico do Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, criado a partir do desdobramento do antigo Departamento de Engenharia Mecânica.
O presente trabalho é um resumo da dissertação de mestrado apresentada por Murilo Antonio Salomão Garcia: "Algoritmo para Converter Sólidos CSG em Sólidos B-Rep" sob a orientação do Prof. Dr. Marcos de Sales Guerra Tsuzuki, defendida em 17/04/2006, na EPUSP.

A íntegra da dissertação encontra-se à disposição com o autor, na Biblioteca de Engenharia Mecânica da Escola Politécnica/USP e no endereço <http://www.pmr.poli.usp.br/pmr/pBT.asp>

FICHA CATALOGRÁFICA

Garcia, Murilo Antonio Salomão

Algoritmo para converter sólidos CSG em sólidos B-Rep / Murilo Antonio Salomão Garcia, Marcos de Sales Guerra Tsuzuki. -- São Paulo : EPUSP, 2006.

12 p. – (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos ; BT/PMR/0623)

1. Modelagem sólida 2. Computação gráfica I. Tsuzuki, Marcos de Sales Guerra II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III. Título IV. Série
ISSN 1517-3526

Algoritmo para Converter Sólidos CSG em Sólidos B-Rep

MURILO ANTONIO SALOMÃO GARCIA
MARCOS DE SALES GUERRA TSUZUKI

Escola Politécnica da Universidade de São Paulo – Departamento de Engenharia Mecatrônica e de Sistemas
Mecânicos – Laboratório de Geometria Computacional
Av. Prof. Mello Moraes, 2231 – CEP 05508-900 – São Paulo, SP, Brasil
(murilo.garcia@poli.usp.br; mtsuzuki@usp.br)

Resumo

O objetivo deste trabalho é definir um novo algoritmo para a conversão de um modelo CSG para um modelo B-Rep. Normalmente isto é feito, percorrendo-se a árvore CSG e interpretando-a, ou seja: aplicando-se rotações, translações e escalamento (transformações afins) e determinando-se a união, interseção ou diferença (operações booleanas) entre dois modelos B-Rep. Nós definimos uma nova abordagem onde o espaço ocupado pelo sólido é determinado como uma representação volumétrica. Após isso, utilizando-se esta representação volumétrica um Modelo Sólido B-Rep é gerado. Um algoritmo assim poderá ser utilizado no futuro para a criação de Modelos Sólidos B-Rep a partir de Imagens Médicas Tridimensionais. Gerando desta forma um modelo sólido completo ao invés de uma Representação por Superfícies, que é como tradicionalmente se tem feito. O Modelo Sólido B-Rep tem a vantagem de permitir a fácil extração de propriedades relativas à massa, como: momento de inércia, centro de massa e volume. Além disso, a uma representação por superfícies não garante a geração de um volume fechado (o que é suficiente para a visualização), porém para o uso na engenharia como análise e manufatura por estereolitografia, é necessário um volume fechado. Um Modelo Sólido B-Rep é uma representação completa para o uso na engenharia. Por fim, posto que a construção de um Modelo Sólido é diferente da construção tradicional de uma Representação por Superfícies, nós tivemos que modificar o algoritmo Marching Cubes para obter o resultado desejado.

Palavras-chave: CSG, B-Rep, Modelos Sólidos, Marching Cubes, Algoritmo de Conversão.

Abstract

The purpose of this work is to define a new algorithm for converting a CSG representation into a B-Rep representation. Usually this conversion is done by walking through the CSG tree translating it, which means: applying rotations, translations and scaling (affine transformations) and determining the union, intersection or difference from two B-Rep solids. We will define another approach where the space used by the solids is determined as a volumetric representation. Then, using the volumetric representation a B-Rep Solid Model is created. Such an algorithm can be used in the future for creating B-Rep Solid Models from Three Dimensional Medical Images. Generating a full Solid Model instead of the traditional approach of only Surface Representations for 3D medical images has the advantage that mass properties are easily extracted from a Solid Model, such as: volume, moment of inertia and mass centre. The surface representation does not guarantee that a closed volume is created. It is enough for visualization. However, for engineering purposes as analysis or stereolithography manufacturing, a closed volume is necessary. A B-Rep Solid Model is a complete representation for engineering purposes. The construction of a Solid Model is different from the traditional construction of a surface model; thus we modified the marching cubes algorithm to reach this objective.

Keywords: CSG, B-Rep, Solid Model, Marching Cubes, Conversion Algorithm.

1. Introdução

A representação de Modelos sólidos por meio de Constructive Solid Geometry (CSG) permite aos usuários criar objetos sólidos 3D extremamente complexos de forma hierárquica combinando-se primitivas geométricas simples por meio de operações Booleanas e transformações afins [1]. Tradicionalmente as primitivas CSG são definidas por objetos analíticos simples, como cubos, cilindros e esferas. Alguns algoritmos CSG mais recentes dão suporte também a primitivas que são modelos sólidos quaisquer definidos pela superfície de suas fronteiras. Usando-se representações volumétricas baseadas em voxels, uma extensão maior a esses algoritmos pode incluir objetos extraídos de dados volumétricos usando-se limiares de intensidade. Estes dados volumétricos podem ser obtidos pela digitalização de objetos reais através de métodos como Tomografia Computadorizada, Ressonância Magnética, imagens de microscópios ou por meio de amostragem de funções implícitas ou procedurais. Estes modelos CSG expandidos são por vezes chamados de modelos CSG Volumétricos e são muito úteis em aplicações cirúrgicas, exames médicos baseados em imagens e modelagem de fenômenos amorfos [2]. Devido à falta de representação explícita das superfícies das fronteiras dos modelos sólidos, a exibição de CSG em tela não é suportada de forma nativa por dispositivos gráficos padrões. Ainda que muitos algoritmos de renderização CSG interativos tenham sido desenvolvidos no passado [3] eles não podem ser diretamente aplicados quando estão envolvidos dados volumétricos.

Uma possível solução para a renderização de um modelo CSG volumétrico é converter-se o modelo CSG em um modelo volumétrico baseado em voxel e aí construir-se o modelo sólido B-Rep e por fim renderizar-se este modelo B-Rep. Este método (conversão de um modelo CSG em um modelo volumétrico) é conhecido como Voxelização CSG, conceitualmente é um problema de se classificar um elemento como pertencente ou não a um conjunto em relação ao objeto CSG para todos os pontos de amostragem no volume espacial. Esta é a metodologia tomada como base para o desenvolvimento deste trabalho, ou seja utilizar-se a Voxelização CSG em seguida gerar-se o modelo B-Rep a partir do modelo de Voxel. O algoritmo que criamos baseia-se no *Marching Cubes* (vide Capítulo 4. *Algoritmo Marching Cubes*), porém existem três diferenças fundamentais no emprego do algoritmo, tradicionalmente os dados de entrada do algoritmo *Marching Cubes* são Imagens Médicas Tridimensionais, no nosso caso serão modelos CSG, no algoritmo tradicional todo o espaço tem que ser percorrido, no nosso novo algoritmo proposto apenas a fronteira do sólido é percorrida, e por fim os dados de saída do *Marching Cubes* são Representações por Superfícies, no nosso caso serão modelos B-Rep. Em função destas diferenças e com o objetivo de deixar o algoritmo mais eficiente propomos as alterações necessárias no Capítulo 3. *Representação B-Rep (Boundary Representation)*.

2. Representação CSG

Representações baseadas na construção do sólido capturam o processo de construção que define o sólido por uma seqüência de operações que instanciam ou combinam primitivas ou combinam o resultado de construções anteriores. Essas representações usualmente capturam a tentativa de criação de um design do usuário em um alto nível de representação que pode ser facilmente editado e parametrizado. Constructive Solid Geometry (CSG) é a representação por construção mais difundida. Suas primitivas são sólidos parametrizados, que podem possuir formas simples (como blocos, cones, cilindros e esferas) ou formas mais complexas apropriadas para um conjunto de aplicações particulares, como rasgos de chaveta e furos, no caso da engenharia mecânica. As primitivas podem ser instanciadas várias vezes e agrupadas hierarquicamente.

Instâncias de primitivas e grupos podem ser manipulados por meio de transformações afins visando mudar a forma original das primitivas possibilitando assim que o usuário possa chegar à forma final desejada. Estas transformações afins incluem (entre outras) as seguintes transformações espaciais: rotações, translações e escalamentos uniformes e não uniformes (esticando os eixos por um fator de escala constante) e reflexões invertendo objetos em relação a um eixo). Estas transformações estão ilustradas na Figura 1. Estas transformações (afins) possuem características em comum, como por exemplo mapear linhas em linhas. Note que algumas delas (translação, rotação e reflexão) preservam os comprimentos das arestas e os ângulos entre estas arestas. Outras (como escalamento uniforme), no entanto, preservam os ângulos mas não os comprimentos. Outras ainda (como escalamento não uniforme) não preservam nem os ângulos nem os

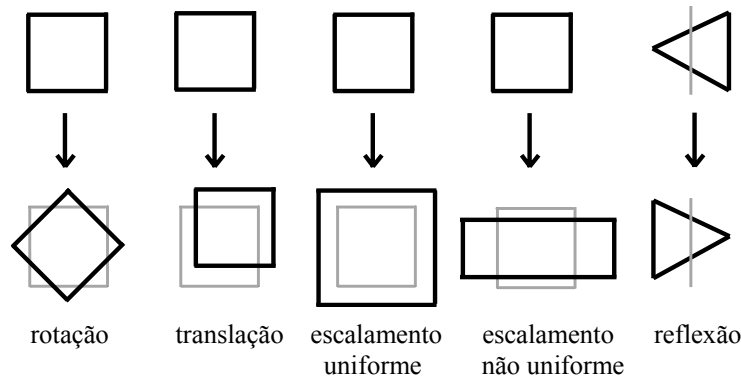


Figura 1 - Exemplos de transformações afins.

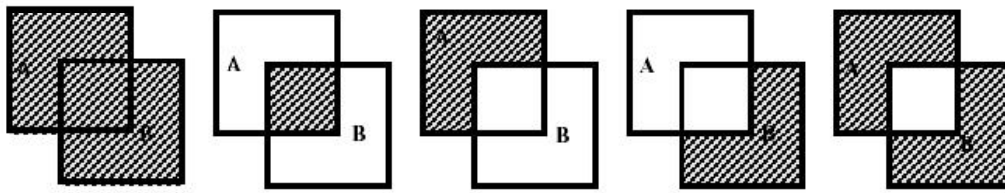


Figura 2. As cinco operações Booleanas não triviais para dois conjuntos (da esquerda para a direita): $A+B=\{a : a \in A \text{ ou } a \in B\}$, $A \cap B=\{a : a \in A \text{ e } a \in B\}$, $A-B=\{a : a \in A \text{ e } a \notin B\}$, $B-A=\{a : a \notin A \text{ e } a \in B\}$, e a diferença simétrica $(A-B)+(B-A)$.

comprimentos, porém ainda assim mapeiam linhas em linhas. Além disso as instâncias transformadas podem ser combinadas através de operações Booleanas regularizadas: união, intersecção e diferença. Estas operações Booleanas realizam o respectivo conjunto de operações Booleanas teóricas, e deste modo transformam o resultado em um conjunto r através da aplicação de operações que isolam o interior do sólido (excluindo a fronteira) seguido de um fechamento topológico (reconstruindo a fronteira do sólido final). Desta forma elas sempre retornam sólidos válidos (embora possivelmente vazios), eliminando possíveis faces internas ao sólido. Ainda que outras operações Booleanas possam ser oferecidas, estas três são convenientes e suficientes, pois entre as 16 combinações diferentes Booleanas entre dois conjuntos, A e B , 8 não possuem fronteira, 3 são triviais, e apenas 5 são úteis no âmbito da modelagem de sólidos, são elas: a união $A + B$, a intersecção $A \cap B$, as diferenças $A - B$ e $B - A$ e a diferença simétrica $(A-B)+(B-A)$, como mostrado na Figura 2.

A Figura 3 ilustra como uma sintaxe simples pode ser utilizada para especificar um sólido em CSG. A tradução desta sintaxe nos leva a um grafo com raiz, cujas folhas representam instâncias das primitivas e cujos nós internos representam transformações afins ou operações Booleanas que produzem sólidos. A raiz representa o sólido correspondente ao grafo CSG. As representações CSG são concisas, sempre válidas no conjunto r do domínio da modelagem, e facilmente parametrizadas e editadas. Muitos algoritmos de modelagem de sólidos trabalham diretamente nas representações CSG através da estratégia de divisão e conquista, onde resultados calculados nas folhas são transformados e combinados árvore acima de acordo com as operações associadas aos nós intermediários. As representações CSG não apresentam de forma explícita nenhuma informação a respeito de conectividade bem como da existência do sólido correspondente. Estas questões topológicas são mais bem abordadas por meio de alguma forma de avaliação da fronteira do sólido, onde um B-Rep total ou parcial é derivado por meio de algoritmo do modelo CSG.

2.1 Classificação de um Ponto para Sólidos CSG

A árvore CSG pode ser vista como uma descrição implícita da geometria do sólido modelado que deve ser avaliada para se seja possível gerar a saída gráfica ou para efeito de cálculos. Para isso é preciso que consigamos classificar um dado ponto do espaço como sendo interior ou exterior ao volume ocupado pelo sólido CSG. Como dito antes a representação CSG é uma árvore. Isso sugere o emprego de métodos de divisão

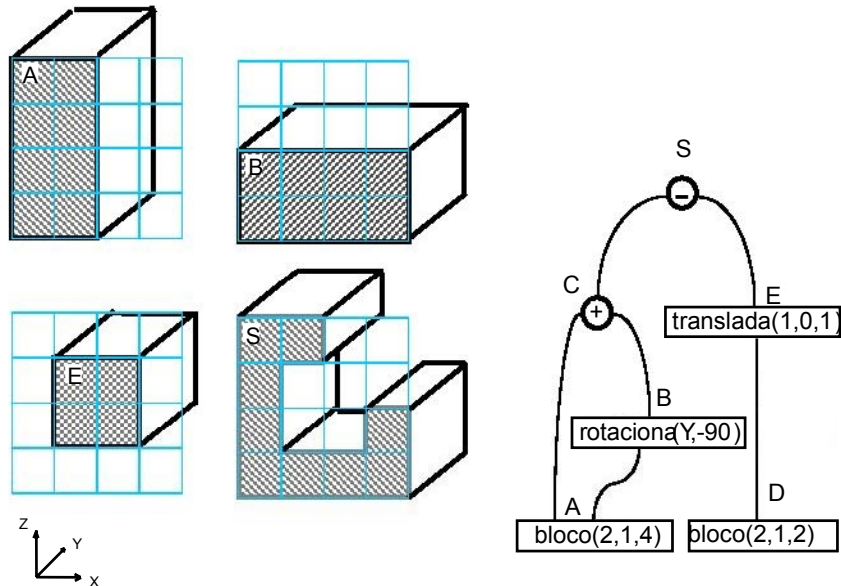


Figura 3. As instâncias A, B, e E são mostradas superpostas no mesmo grid de referência. O sólido S foi especificado pela seguinte seqüência de comandos: A = bloco(2,1,4); B = rotaciona(A,Y-eixo,90); C = A+B; D = bloco(2,1,2); E = translada(D,1,0,1); S = C-E; O correspondente grafo CSG (direita) possui 2 primitivas nas folhas, 2 nós com transformações, e 2 nós com expressões Booleanas regularizadas.

e conquista, ou descida em profundidade recursiva para calcular-se a classificação do ponto (como interior ou exterior ao sólido) [4].

3. Representação B-Rep (Boundary Representation)

Os modelos sólidos B-Rep derivaram dos modelos de poliedros utilizados em computação gráfica na representação de objetos e cenas para remoção de linhas e superfícies ocultas. Eles podem ser encarados como modelos gráficos avançados que tentam resolver alguns problemas através da inclusão de uma descrição completa da superfície da fronteira do objeto. Existem três entidades primitivas face, aresta e vértice, e a informação sobre a geometria anexada a elas forma a constituição básica dos modelos B-Rep. Além das informações sobre a geometria como equação da face e coordenadas do vértice, um modelo B-Rep deve representar também o relacionamento entre as faces, arestas e vértices. De acordo com Mäntylä [5], é costume agregar-se toda informação relativas à geometria das entidades sob o termo geometria do modelo de fronteira, e de forma similar, informações relativas a suas interconexões sob o termo topologia. Uma descrição detalhada sobre topologia pode ser encontrada em [6]. Pode-se dizer que a topologia funciona como uma goma onde as informações geométricas são aglutinadas; ou então que "as informações topológicas criam um vigamento no qual as informações geométricas são posicionadas". Com o objetivo de simplificar os algoritmos, principalmente na determinação do circuito de arestas ao redor de uma face a estrutura *half-edge* foi criada. Observou-se que a aresta na estrutura *winged-edge* [7] original tinha duas funções principais: representar o circuito de arestas ao redor de uma face e representar a aresta em si. O algoritmo para determinar o circuito de arestas ao redor de uma face era muito complexo com varias regras. Alguns pesquisadores observaram que se separando estas duas funções o algoritmo torna-se muito mais simples [8]. Desta forma, os modeladores de sólidos modernos têm uma entidade para representar a aresta em si e uma outra para representar o circuito de arestas ao redor da face. Conforme o modelador de sólidos se torna mais complexo, torna-se necessário adicionar uma nova primitiva: laço. A face pode possuir buracos interiores para representar protuberâncias ou depressões. Neste caso a face possui um laço externo e zero ou mais laços internos.

3.1 Operadores de Euler

Os Operadores de Euler foram introduzidos originalmente por Baumgart [7] juntamente com a estrutura de dados *winged-edge*. Para permitir a manipulação das entidades topológicas e ao mesmo tempo garantir a

Tabela 1. Nomenclatura dos Operadores de Euler.

Símbolo	Significado	Tradução
M	make	cria
K	kill	remove
V	vertex	vértice
E	edge	aresta
F	face	face
S	shell	shell
H	hole	furo
R	ring	laço

validade do modelo sólido, os operadores de Euler são utilizados satisfazendo a equação de Euler. A equação de Euler-Poincaré relaciona entidades primitivas que formam o sólido de maneira quantitativa:

$$v - a + f = 2(s - h) + r$$

Onde v é o número de vértices, a é o número de arestas, f é o número de faces, s é o número de *shells*, e r é o número de anéis (ou laços internos). Foi provado por Mäntylä [5], que os operadores de Euler formam um conjunto completo de primitivas para modelagem de sólidos *manifold*. Mais precisamente, todos os poliedros topologicamente válidos podem ser construídos a partir de um poliedro inicial por meio de uma seqüência finita de operadores de Euler. Eles permitem que a construção do sólido possa ser executada passo a passo, escondendo todos os detalhes de implementação da estrutura de dados. Isto torna os operadores de Euler muito poderosos no âmbito da modelagem de sólidos. Com a finalidade de facilitar a memorização, os Operadores de Euler estão definidos utilizando a nomenclatura da Tabela 1. Durante o processo de construção de um sólido pela utilização de Operadores de Euler, a validade topológica do mesmo é mantida observando-se a equação de Euler-Poincaré. Entretanto, é comum o agrupamento de Operadores de Euler em uma certa seqüência para mantermos também a geometria válida. É importante observar que não há como manter a geometria válida em todos os estágios da construção. Por isto, os Operadores de Euler devem ser agrupados em seqüências que possuam algum significado. Vários autores demonstram que seis operadores são suficientes para construir todos os objetos. Enquanto estes seis operadores podem ser escolhidos de várias maneiras, considerações de modularidade e independência criaram apenas pequenas variações na coleção encontrada na literatura. A seguir foram selecionados e estão descritos os seis operadores mais comumente utilizados na literatura, e para cada qual o inverso de seu par:

1. Criação de Sólidos: um vértice, face e sólido são criados <MVSF> - **Make Vertex Solid Face**.
2. Remoção de Sólidos: um vértice, face e sólido são removidos <KVSF> - **Kill Vertex Solid Face**.
3. Divisão de Vértices: um vértice é dividido em dois vértices conectados por uma aresta <MEV> - **Make Edge Vertex**.
4. União de Vértices: dois vértices vizinhos são unidos em um único e a aresta entre eles é removida. Essa operação é a inversa da divisão de vértices <KEV> - **Kill Edge Vertex**
5. Divisão de Faces: a face é dividida em duas por meio da adição de uma nova aresta entre dois vértices de seu laço externo <MEF> - **Make Edge Face**.
6. União de Faces: duas faces adjacentes são unidas através da remoção da aresta comum a elas. Essa operação é a inversa da divisão de faces <KEF> - **Kill Edge Face**.

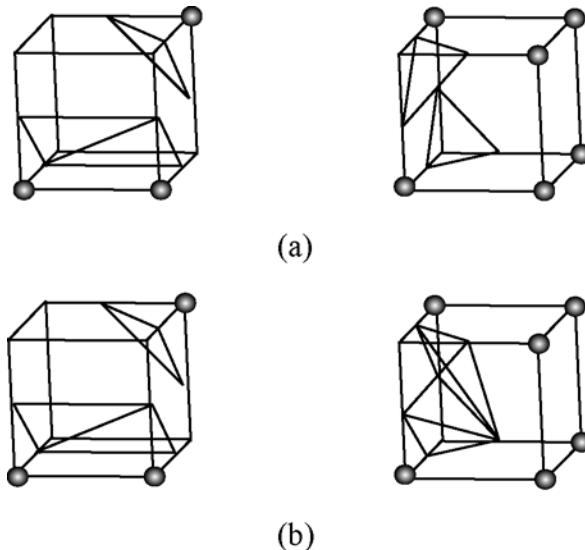


Figura 4. a) Exemplo do problema do furo de tipo A produzido pelo algoritmo convencional do marching cubes. (b) Solução para o problema.

7. Divisão de Laços: um laço é dividido em dois laços, um deles sendo uma nova fronteira interior do laço, através da remoção de uma aresta. Este operador, por conseguinte irá criar um novo laço interno <KEMR> - **Kill Edge Make Ring**.
8. União de Laços: dois laços, ao menos um sendo um laço interno, são unidos para produzir um só laço, desta forma, um laço interno é removido. Essa operação é a inversa da divisão de laços <MEKR> - **Make Edge Kill Ring**.
9. Criação de Furos: une duas faces, e cria um furo passante. Este é um operador necessário quando um toróide é criado <KFMRH> - **Kill Face Make Ring Hole**.
10. Remoção de Furos: cria uma nova face e remove o furo passante <MFKRH> - **Make Face Kill Ring Hole**.
11. Criação de um novo *shell*: Transforma o anel de uma face em uma nova face, e todo o conjunto de faces associadas à nova face constituirá um novo *shell* <MSFKR> - **Make Shell Face Kill Ring**.
12. Remoção de *shells*: Transforma uma face em um anel, e o *shell* a qual essa face pertencia é removido. <KSFMR> - **Kill Shell Face Make Ring**.

4. Algoritmo Marching Cubes

Marching Cubes é um algoritmo utilizado na renderização de isosuperfícies em dados volumétricos (Lorensen e Cline [9]). O princípio deste algoritmo é que podemos definir um *voxel* (cubo) avaliando-se os valores de pixels nos oito vértices do cubo. Se um ou mais pixels possuem valores menores do que o isovalor especificado pelo usuário, e um ou mais possuem valores maiores do que este, sabemos que este *voxel* deve contribuir para construção da isosuperfície. Se determinado quais arestas do cubo são intersectadas pela isosuperfície, podemos criar faces triangulares que dividem o cubo em regiões interiores e exteriores à isosuperfície. Conectando-se essas faces triangulares de todos os cubos da isosuperfície, obtemos a representação da superfície. Este algoritmo é usualmente usado para extrair-se a superfície de órgãos das imagens médicas. Este algoritmo provê um meio fácil e simples para obter-se uma representação por superfície tendo como ponto de partida uma série de secções de um objeto. Existem 256 possibilidades de configuração do cubo, em cada uma delas, a isosuperfície é triangulada. O algoritmo *weaving wall* [10] foi proposto junto com todas as 256 possibilidades definidas explicitamente. Este método, entretanto, é muito suscetível a erro. Na implementação convencional do marching cubes, o uso de simetria reduz o número de casos para 15. A simetria complementar é definida como a equivalência entre configurações complementares [9]. Duas configurações são definidas como complementar, se a ação do operador lógico NOT sobre todos os pontos e uma configuração se transformar na

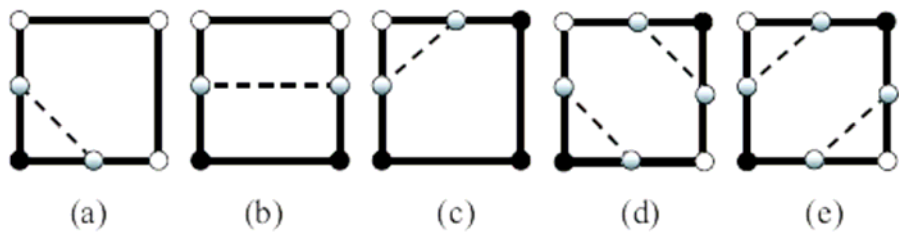


Figura 5. Isoponto localizado no ponto médio da aresta do cubo e as condições para existência (vértices em lados opostos da fronteira).

outra. Além da simetria complementar é possível rotacionar uma configuração por qualquer ângulo ao redor de qualquer dos três eixos primários e/ou aplicar reflexão em relação a um desses eixos à configuração. O uso da simetria que reduz o número de configurações de cubo pode produzir incoerências topológicas, ou furos em determinadas situações de cubos adjacentes conhecido como problema do furo de tipo A, ilustrado na Figura 4. A tabela com configurações pré-definidas de cubo é uma constante em praticamente todas as variações do algoritmo Marching Cubes. Porém o algoritmo proposto por Delibasis et al [11] não apresenta a tabela com as configurações pré-definidas. Para evitar a comparação com as configurações pré-definidas foi definido um algoritmo para ordenar as intersecções das arestas do cubo de modo a definir polígonos algorítmicamente. Note que a saída do algoritmo de Delibasis é uma superfície formada por polígonos e não triângulos como na implementação original.

5. Algoritmo de Conversão de CSG para B-Rep

O algoritmo proposto por Delibasis et al [11] constrói as configurações 2-D mostradas na Figura 5 (a), (b) e (c). No entanto a ambigüidade presente nos casos mostrados na Figura 5 (d) e (e) (esta é a ambigüidade que origina o furo tipo A) não é resolvida. O algoritmo proposto neste trabalho resolve esta ambigüidade, distinguindo quando é necessário criar-se um caso ou o outro, para tanto é necessário utilizar-se informações dos isocubos adjacentes.

5.1 Determinando-se a Existência de uma Aresta

Arestas são geradas conectando-se dois isopontos situados numa mesma face do cubo levando-se em consideração as seguintes regras ilustradas na Figura 5:

- {Condição 0.} Os dois isopontos devem estar situados na mesma face do cubo;
- {Condição 1.} Os dois isopontos devem compartilhar um vértice adjacente classificado com interno;
- {Condição 2.} Os dois isopontos devem possuir vértices adjacentes (um interno e o outro externo) de forma que um vértice adjacente classificado como interno deve ser adjacente a um classificado como interno e a outro classificado como externo;
- {Condição 3.} Os dois isopontos devem compartilhar um vértice adjacente externo e os outros três vértices da face devem ser classificados como internos.

Por fim, se a condição 0 associada com a condição 1, 2 ou 3 é uma sentença verdadeira a aresta é criada.

5.2 O Algoritmo

A parte inicial do algoritmo é similar ao algoritmo proposto por Delibasis [11], o espaço 3D é discretizado por uma malha de cubos de aresta unitária. Começamos então a marchar na malha de cubos a partir de um isocubo (cubo intersecionado pela fronteira do sólido), e marcharemos somente de um isocubo para outro. Desta maneira marcharemos apenas pela fronteira do sólido. Para sabermos se um cubo é um isocubo classificamos todos os vértices que formam esse cubo em relação ao sólido (interior ou exterior), se todos os

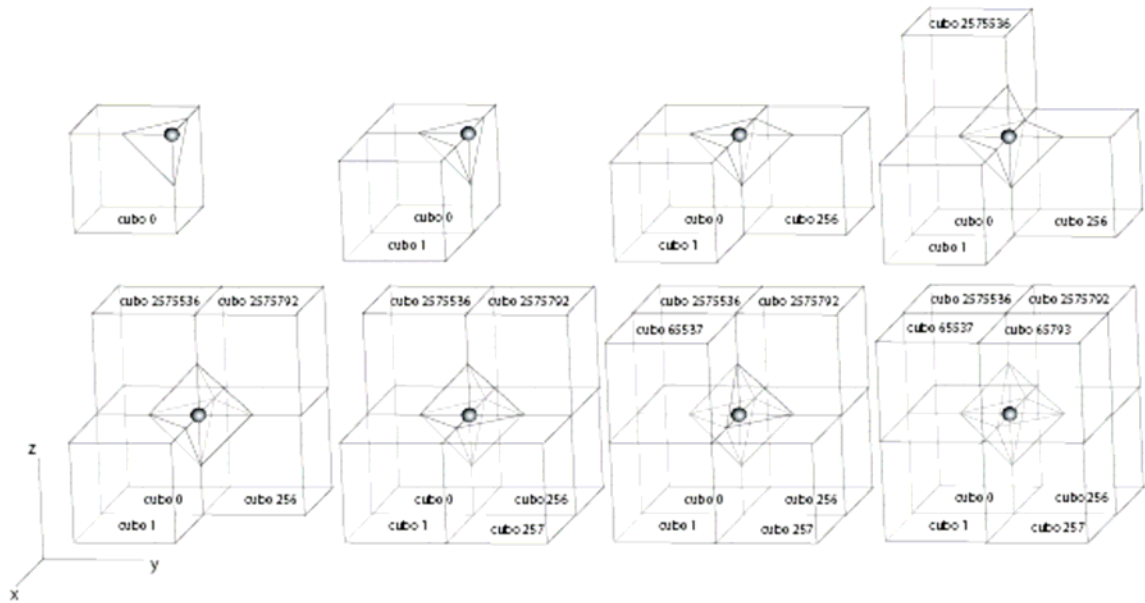


Figura 6. Marcha entre os cubos por meio das faces.

vértices possuírem a mesma classificação o cubo ou está totalmente dentro ou está totalmente fora do sólido. Se um dos vértices for classificado diferente de qualquer outro vértice desse mesmo cubo, o cubo é interseccionado pela fronteira do sólido e portanto é um isocubo. Para saber se um vértice é interior ou exterior ao cubo é necessário tomar-se as coordenadas do vértice e utilizar a árvore CSG (vide *Seção 2.1 Classificação de um Ponto para Sólidos CSG*). O algoritmo de Delibasis gera polígonos e não triângulos e estes polígonos nem sempre estão contidos num plano. Portanto se fez necessário inserir uma etapa de triangularização no algoritmo. O cerne do algoritmo aqui proposto é a maneira que se marcha de um cubo para outro. Esta marcha é realizada de forma ordenada passa-se de um cubo ao outro através de uma face comum entre eles onde uma aresta tenha sido gerada, desta maneira tira-se o maior proveito possível das informações de adjacência.

Assim que o cubo é processado os cubos para onde podemos marchar e as faces desses cubos que possuem arestas geradas são empilhados. A informação das faces que possuem arestas geradas é utilizada durante o processamento de cada cubo na parte de geração dos triângulos. Trabalhamos com duas pilhas de cubos: a pilha dos cubos que estão sendo processados (*pilha.cubos.atuais*) e a pilha dos que estão conectados às faces pelas quais a marcha é possível (*pilha.cubos.futuros*), toda vez que a *pilha.cubos.atuais* fica vazia ela é substituída pela *pilha.cubos.futuros* até que as duas pilhas estejam vazias, que é quando o algoritmo chega ao fim. Na Figura 6 podemos observar como se daria a marcha se tivéssemos um sólido que englobasse apenas um vértice do cubo. Podemos observar ainda as etapas de construção do sólido. Os números dos cubos são gerados automaticamente de acordo com sua posição espacial e são únicos.

5.3 Geração das Faces Triangulares no Modelo B-Rep

Os operadores de Euler (vide *Seção 3.1 Operadores de Euler*) serão utilizados para gerar as faces triangulares. O caso mais simples de geração de face triangular ocorre apenas uma vez, quando ainda não existe nada gerado, e isso ocorre logo após o processamento do primeiro cubo. Para o primeiro vértice deste triângulo utiliza-se o operador <MVSF> (Make Vertex Solid Face). Desta maneira o sólido é criado juntamente com seu primeiro vértice ($V1$) e sua primeira face, note que esta face só será fechada no final da criação do sólido. Agora se faz necessário criar o segundo vértice ($V2$) e a primeira aresta ($V1V2$), conectando-se o primeiro vértice ao segundo. Para tanto o operador <MEV> (Make Edge Vertex) é empregado. Após isso podemos criar o terceiro vértice ($V3$) e conectá-lo ao segundo ($V2$) por meio de uma segunda aresta ($V2V3$). Para isso empregamos novamente o operador <MEV> (Make Edge Vertex). Por último precisamos criar a aresta ($V3V1$) que conectará

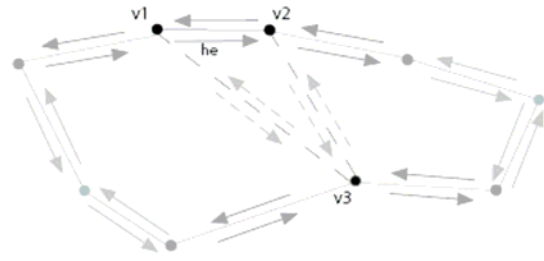


Figura 7. Criação de uma face triangular com três vértices já criados Caso I.

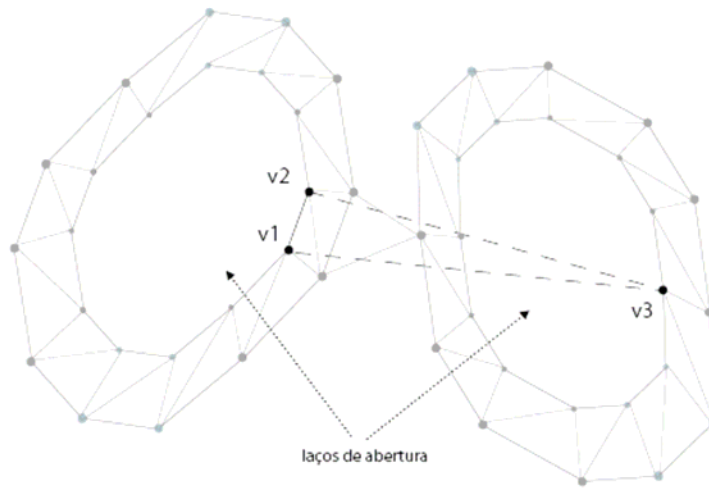


Figura 8. Criação de uma face triangular com três vértices já criados Caso II.

o terceiro vértice ao primeiro e finalmente criar a face do triângulo para tanto devemos utilizar <MEF> (Make Edge Face). De forma resumida temos a seguinte seqüência de operadores:

- <MVSF> Make Vertex Solid Face;
- <MEV> Make Edge Vertex;
- <MEV> Make Edge Vertex;
- <MEF> Make Edge Face;

Mostramos em detalhes na Figura 7 como fica a disposição das *half-edges* (*he*) com a geração da face, o laço formado pelas *half-edges* *he1*, *he5* e *he3*, é o que chamamos de Laço de Abertura do sólido. É a partir desse laço que o sólido "cresce", ou seja, as *half-edges* contidas nesse laço são usadas na geração das novas faces do sólido. Em alguns casos o sólido pode possuir mais do que um laço de abertura numa etapa intermediária da construção do sólido, porém como essa situação é temporária e com o avanço da marcha estes laços serão unidos, voltando a existir apenas um laço de abertura.

Geração de Faces Triangulares com Uma Aresta Já Criada

Neste caso a face que temos que criar possui uma aresta em comum com uma face já criada. Existem duas situações onde isso pode ocorrer: estamos criando um polígono que foi previamente triangularizado por possuir mais do que três vértices, ou estamos gerando uma face que compartilha uma aresta com um cubo vizinho que já foi visitado. Para os dois casos a maneira que utilizamos para saber que estamos nesta situação é a mesma. A *half-edge* *he* pertence ao laço de abertura e é recuperada por meio da aresta $V1V2$, isto é possível para o caso (b) pois estamos supondo que a marcha se deu do *cubo 1* para o *cubo 3*, e nesse processo a aresta foi recuperada; no caso (a) temos a aresta pois acabamos de construir a *face 1*. Em posse de *he* verificamos se as seguintes expressões são verdadeiras, onde LA é o conjunto de todos os laços de abertura: $he \rightarrow vtx() = V1$, $he \rightarrow nxt() \rightarrow vtx() = V2$, $v3 \notin LA$, onde $he \rightarrow vtx()$ é o vértice associado à *he* e $he \rightarrow nxt()$ é a *half-edge* seguinte à *he*.

Desta forma já temos dois vértices ($V1$ e $V2$) e uma aresta $V1V2$ criados, portanto podemos criar o terceiro vértice ($V3$) e conectá-lo ao segundo por meio de uma segunda aresta ($V2V3$). Para isso empregamos o operador <MEV> (Make Edge Vertex). Por último precisamos criar a aresta ($V3V1$) que conectará o terceiro vértice ao primeiro e finalmente criar a face do triângulo para tanto devemos utilizar <MEF> (Make Edge Face). De forma resumida temos a seguinte seqüência de operadores:

- <MEV> Make Edge Vertex;
- <MEF> Make Edge Face;

Geração de Faces Triangulares com Duas Arestas Já Criadas

A única maneira de esta situação ocorrer é quando dois cubos adjacentes ao cubo que está sendo processado já foram visitados e tiveram cada um uma face criada que compartilha uma aresta com o cubo em processamento. Estas arestas pertencem à face a ser criada, e os três vértices já foram criados, faltado apenas a criação de uma última aresta para fechar a face para tanto devemos utilizar <MEF> (Make Edge Face). A maneira que utilizamos para saber que estamos nesta situação é a seguinte, a *half-edge* he pertence ao laço de abertura e é recuperada por meio da aresta $V1V2$, isto é possível pois estamos supondo que a marcha se deu do *cubo 1* para o *cubo 3*, e nesse processo a aresta foi recuperada. Em posse de he verificamos se as seguintes expressões são verdadeiras: $he \rightarrow vtx() = V1$, $he \rightarrow nxt() \rightarrow vtx() = V2$, $he \rightarrow prv() \rightarrow vtx() = V3$, onde $he \rightarrow prv()$ é *half-edge* a anterior a he . Na Figura 9 as três faces pertencem a cubos diferentes, suponha que a *face 1* e a *face 2* já foram criadas e portanto a *face 3* já possui duas arestas criadas $V1V2$ e $V3V1$ para tanto o *cubo 1* e o *cubo 2* necessariamente já foram visitados. Concluimos que para a *face 3* ser construída falta somente a criação da última aresta ($V2V3$) e da face em si. Abaixo destacamos o operador utilizado:

- <MEF> Make Edge Face;

Geração de Faces Triangulares com Três Arestas Já Criadas

Existem situações em que todas as arestas da face são compartilhadas com cubos adjacentes, no caso de todos estes cubos já terem sido visitados e processados, todas as três arestas da face já terão sido criadas. Neste caso nada deve ser feito pois a face já foi indiretamente criada, normalmente isso ocorre quando a última face é adicionada ao sólido. A maneira que utilizamos para saber que estamos nesta situação é a seguinte, a *half-edge* he pertence ao laço de abertura e é recuperada por meio da aresta $V1V2$, isto é possível pois estamos supondo que a marcha se deu do *cubo 1* para o *cubo 3*, e nesse processo a aresta foi recuperada. Em posse de he verificamos se as seguintes expressões são verdadeiras: $he \rightarrow nxt() \rightarrow nxt() \rightarrow nxt() = he$. Ou seja já existe um laço formado pelos vértices $V1$, $V2$ e $V3$, conseqüentemente as arestas e a face já estão criadas.

Geração de Faces Triangulares Três Vértices Já Criados - Caso I

Existe uma segunda situação onde já existem três vértices criados, porém não se verifica a expressão que rege o caso anterior. A maneira que utilizamos para saber que estamos nesta situação é a seguinte, a *half-edge* he (Figura 11) pertence ao laço de abertura e é recuperada por meio da aresta $V1V2$. Em posse de he verificamos se as seguintes expressões são verdadeiras, onde l é o laço de abertura que contém $V1$ e $V2$: $he \rightarrow vtx() = V1$, $he \rightarrow nxt() \rightarrow vtx() = V2$, $he \rightarrow prv() \rightarrow vtx() \neq V3$, $V3 \in l$. Ou seja já existem os vértices $V1$, $V2$ e $V3$ de modo que pertençam ao mesmo laço de abertura, porém não existem as arestas $V2V3$ e $V3V1$. Desta forma já temos três vértices e uma aresta ($V1V2$) criados, portanto podemos conectar o segundo vértice ao terceiro vértice por meio de uma segunda aresta ($V2V3$). Para isso empregamos o operador <MEF> (Make Edge Face). Por último precisamos criar a aresta ($V3V1$) que conectará o terceiro vértice ao primeiro e finalmente criar a face do triângulo para tanto devemos utilizar novamente <MEF> (Make Edge Face). Esta opção cria um novo laço de abertura. De forma resumida temos a seguinte seqüência de operadores:

- <MEF> Make Edge Face;
- <MEF> Make Edge Face;

Geração de Faces Triangulares Três Vértices Já Criados - Caso II

Existe uma terceira situação onde já existem três arestas criadas, porém também não se verifica as expressões que regem os casos anteriores. A maneira que utilizamos para saber que estamos nesta situação é a seguinte, a *half-edge* he (Figura 12) pertence ao laço de abertura e é recuperada por meio da aresta $V1V2$. Em posse de he

verificamos se as seguintes expressões são verdadeiras, onde l é o laço de abertura que contém $V1$ e $V2$: $he \rightarrow vtx() = V1$, $he \rightarrow nxt() \rightarrow vtx() = V2$, $he \rightarrow prv() \rightarrow vtx() \neq v3$, $v3 \notin l$. Ou seja já existem os vértices $V1$, $V2$ e $V3$ de modo que $V1$ e $V2$ pertençam ao mesmo laço de abertura, porém $V3$ pertence a um outro laço de abertura e não existem as arestas $V2V3$ e $V3V1$. Desta forma já temos três vértices e uma aresta ($V1V2$) criados, porém estamos em uma situação em que o sólido em construção é topologicamente igual a um toróide ou um tubo, por exemplo, nos quais foram criados mais do que um laço de abertura. Nesse caso em particular os operadores <KFMRH> e <MEKR> precisam ser executados. O segundo operador (<MEKR>) irá criar a segunda aresta ($V2V3$). Por último precisamos criar a aresta ($V3V1$) que conectará o terceiro vértice ao primeiro e finalmente criar a face do triângulo para tanto devemos utilizar novamente <MEF> (Make Edge Face). Esta operação transforma dois laços de abertura em um único. De forma resumida temos a seguinte seqüência de operadores:

- <KFMRH> Kill Face Make Ring Hole;
- <MEKR> Make Edge Kill Ring;
- <MEF> Make Edge Face;

6. Resultados

As etapas da construção do sólido, classificação dos vértices, geração dos isopontos e geração dos polígonos triangularizados, podem ser observadas na Figura 9. Durante a criação do sólido, mais de um laço de abertura pode ser criado para depois serem unidos, na Figura 10 podemos observar esse processo. Sólidos mais complexos também foram usados nos testes como o ilustrado na Figura 11.

7. Conclusões

Um novo algoritmo para converter sólidos CSG em sólidos B-Rep foi proposto e implementado. Esse novo algoritmo é uma combinação de voxelização CSG e marching cubes. Uma vantagem deste algoritmo em relação ao marching cubes é o fato de marchar-se apenas na fronteira do sólido, esta alteração reduziu a complexidade do algoritmo marching cubes de polinomial de primeira ordem para uma complexidade constante. Outra alteração é a utilização de arestas geradas em um dado cubo no cubo adjacente, desta forma constrói-se um sólido válido, pois as possíveis ambigüidades encontradas em uma face são tratadas de maneira igual em cubos adjacentes evitando problemas como furos. Uma possível melhoria a ser implementada no futuro é a alteração da entrada de dados para que possam ser usados dados volumétricos obtidos pela digitalização de objetos reais através de métodos como Tomografia Computadorizada, Ressonância Magnética, imagens de microscópios ou por meio de amostragem de funções implícitas ou procedurais.

Referências

- [1] HOFFMANN, C. M., Geometric and Solid Modeling: An Introduction. Morgan Kaufmann Publishers, 1989.
- [2] FANG, S.; LIAO, D., Fast CSG Voxelization by Frame Buffer Pixel Mapping, em Proceedings of the 2000 IEEE symposium on Volume visualization, (Salt Lake City, Utah, United States), p. 43-48, 2000.
- [3] THIBAUT, W. C.; NAYLOR, B. F., Set Operations on Polyhedra Using Binary Space Partitioning Trees, SIGGRAPH'87, v. 21, p. 153-162, Julho 1987.

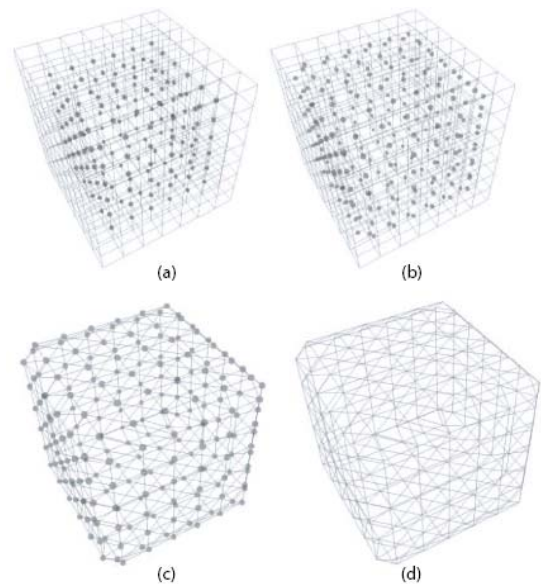


Figura 9 – Em (a) temos os cubos e os vértices interiores ao sólido, em (b) temos os isopontos nas arestas do cubo, em (c) temos os isopontos conectadas pelas arestas das faces triangulares e em (d) temos as faces do sólido final.

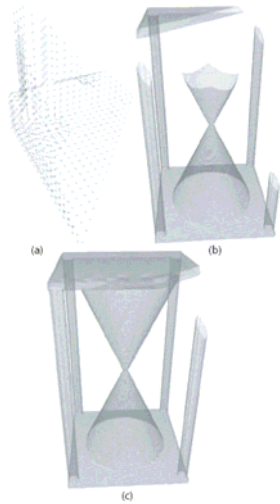


Figura 10 – Em (a) ampliamos a imagem do início da construção do sólido com apenas 1 laço de abertura criado, em (b) temos uma etapa intermediária da construção do sólido onde 5 laços de abertura foram criados e em (c) temos uma etapa mais avançada da criação do sólido onde parte dos laços de abertura que tinham sido criados foram unidos restando apenas 2 laços de abertura.



Figura 11 – Dado formado apenas por um paralelepípedo e algumas esferas. Foram processados 123140 cubos e gerados 246272 triângulos.

- [4] REQUICHA, A. A. G., Representation for Rigid Solids: Theory, Methods and Systems, Computing Surveys, v. 12, p. 437-464, Dezembro 1980.
- [5] MÄNTYLÄ, M., An Introduction to Solid Modeling. New York, NY, USA: Computer Science Press, 1988.
- [6] SÉGONNE, F., Segmentation of Medical Images under Topological Constraints. PhD thesis, Massachusetts, Dezembro 2005, Department of Electrical Engineering and Computer Science, Massachusetts Institute Of Technology.
- [7] BAUMGART, B. G., A Polyhedron Representation for Computer Vision, em AFIPS Conf. (AFIP PRESS, A., ed.), v. 44, (Va), p. 589-596, 1975.
- [8] TORIYA, H.; CHIYOKURA, H., 3D CAD Principles and Applications. Berlin: Springer-Verlag, 1991.
- [9] LORENSEN, W. E.; ECLINE, H., Marching Cubes: A High Resolution 3-D Surface Construction Algorithm, Computer Graphics, v. 21, p. 163-169, Julho 1987.
- [10] H.H. Baker, Building Surfaces of Evolution: The Weaving Wall, International Journal of Computer Vision, v. 3, p. 51-71, Maio 1989.
- [11] DELIBASIS, K.; MATSOPOULOS, G.; MOURAVLIANSKY, N.; NIKITA, K., A Novel and Efficient Implementation of The Marching Cubes Algorithm, Computerized Medical Imaging and Graphics, v. 25, p. 343-352, Julho 2001.

BOLETINS TÉCNICOS

TEXTOS PUBLICADOS

Foram publicados até outubro de 1999, 152 boletins técnicos do PMC relacionados a temas que incluem assuntos de pesquisa do atual Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

- BT/PMR/9901 – Análise por Elementos Finitos de Transferência de Calor em Elementos Combustíveis Nucleares com Diversos Formatos de Seção Transversal – LUCIANO ABE, JOÃO BATISTA DE AGUIAR
- BT/PMR/9902 – Modelagem de Transdutores Piezelétricos de Potência – MARCELO YASSUNORI MATUDA, JULIO CEZAR ADAMOWSKI
- BT/PMR/9903 – Estudo sobre Escoamentos Críticos em Microorifícios e Capilares – KAZUTO KAWAKITA, MARCOS TADEU PEREIRA
- BT/PMR/9904 – Desenvolvimento e Ensaio de Transdutor Multicomponentes para Medição de Força e Momento – MIGUEL JAIME SANDOVAL RODRÍGUEZ, TARCISIO ANTONIO HESS COELHO
- BT/PMR/9905 – Sistema de Posicionamento Pneumático Rotativo – MARCOS COSTA HUNOLD, EDUARDO LOBO LUSTOSA CABRAL
- BT/PMR/0001 – Análise Dinâmica de Sistemas de Produção Flutuante de Petróleo Através de Redes Neurais Artificiais – JHONNY LEONIDAS MERMAS TIQUILLOCA, MARCELO GODOY SIMÕES, HÉLIO MITIO MORISHITA
- BT/PMR/0002 – Aplicação do Método da Expansão em Funções Hierárquicas na Solução das Equações de Navier-Stokes para Fluidos Incompressíveis – GAIANÉ SABUNDJIAN, EDUARDO LOBO LUSTOSA CABRAL
- BT/PMR/0003 – Aplicação de Redes de Petri Interpretadas na Modelagem de Sistemas de Elevadores em Edifícios Inteligentes – GLADYS D. BASTIDAS GUSTIN, PAULO EIGI MIYAGI
- BT/PMR/0004 – Método de Modelagem e Análise de Sistemas de Manufatura Baseado no Modelo dos Recursos e no Modelo Funcional – ADINAN DE SOUZA, LUCAS ANTONIO MOSCATO
- BT/PMR/0005 – Metodologia para Modelagem de Sistemas Ar Condicionado em Edifícios Inteligentes – EMÍLIA VILLANI, PAULO EIGI MIYAGI
- BT/PMR/0006 – Análise Experimental da Cravação de Âncoras de Carga Vertical – PHILIP VON PRITZELWITZ, EDISON GONÇALVES
- BT/PMR/0007 – Método de Calibração de Câmeras para Criação de Ambiente Virtual – SILVANA APARECIDA GRAMINHO DA SILVA, JUN OKAMOTO JÚNIOR
- BT/PMR/0008 – Análise de Processos de Separação dos Resíduos de Corrosão do Cobre em Meio Aquoso – ANDERSON DE OLIVEIRA FERNANDES, EDISON GONÇALVES
- BT/PMR/0009 – Mancal Magnético do Tipo Atração com Controle Uniaxial – ISAIAS DA SILVA, OSWALDO HORIKAWA
- BT/PMR/0010 – Aplicação dos Conceitos de Confiabilidade à Análise de Falhas e Elaboração de Políticas de Manutenção – IDÉLCIO ALEXANDRE PALHETA CARDOSO, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0011 – Sistema Baseado em DSP para Controle em Tempo Real de Motor Elétrico – SILVIO SZAFIR, MARCELO GODOY SIMÕES
- BT/PMR/0012 – Modelagem Não-Linear do Ensaio de Tração Empregando o MEF – DANIEL BENÍTEZ BARRIOS, EDISON GONÇALVES, JAVIER GARCÍA DE LA FIGAL COSTALES
- BT/PMR/0013 – Sistema de Informação: Estudo de Caso no Fluxo de Materiais para a Fabricação de Blanks – JOSÉ JEAN-PAUL ZANLUCCHI DE SOUZA TAVARES, JOSÉ REINALDO SILVA
- BT/PMR/0101 – Análise do Processo Usinagem Usando o Método dos Elementos Finitos – MELCHOR RODRÍGUEZ MADRIGAL, JOÃO BATISTA AGUIAR
- BT/PMR/0102 – Medição de Viscosidade de Líquidos por Ultra-Som – FLÁVIO BUIOCHI, JULIO CEZAR ADAMOWSKI
- BT/PMR/0103 – Um Ambiente de Ultra Baixas Temperaturas e Campos Magnéticos Super Intensos – RAMON VALLS MARTIN, CARLOS CHIEN CHING TU, NEI FERNANDES DE OLIVEIRA JR.
- BT/PMR/0104 – Formato e Estabilidade do Balão Têxtil no Processo de Torção “Dois por Um” Descendente – JOÃO PAULO PEREIRA MARCICANO, CARLOS CHIENG CHING TU
- BT/PMR/0105 – Monitoramento de Desgaste em Fresas de Topo Através da Medição de Torque – VINICIUS LEAL CARNEIRO DA CUNHA, LUCAS ANTONIO MOSCATO
- BT/PMR/0106 – Análise da Resistência Mecânica de uma Roda de Alumínio – FLÁVIO ARCANJO SANTANA, GILBERTO FRANCISCO MARTHA DE SOUZA

- BT/PMR/0107 – Modelagem Orientada a Processo e Objeto para Sistemas Autônomos de Transporte – FABRÍCIO JUNQUEIRA, PAULO EIGI MIYAGI
- BT/PMR/0108 – Balanceamento Adaptativo e Completo de Mecanismos Planos, Articulados, Cadeia Cinemática Aberta, de Dois Graus de Liberdade – LIANG YONG, TARCÍSIO ANTONIO HESS COELHO
- BT/PMR/0109 - Análise de Resistência ao Escoamento de Vaso de Pressão Autoportante para o Transporte Rodoviário de Produtos Perigosos a Granel – ORLANDO MERSCHMANN JR., GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0201 – Recuperação de Formas de Objetos Através de Curvas Senoidais – RÉGIS PASINI, OSWALDO HORIKAWA
- BT/PMR/0202 – Metodologia de Análise da Influência das Tensões Residuais no Comportamento à Fratura – DANIEL BENÍTEZ BARRIOS, EDISON GONÇALVES
- BT/PMR/0203 – Aritmética Intervalar Aplicada em um Modelador de Sólidos B-rep – MARCELO SHIMADA, MARCOS DE SALES GUERRA TSUZUKI
- BT/PMR/0204 – Análise do Comportamento Biomecânico da Interação Prótese-Osso Humano – GILBERTO GARCÍA DEL PINO, EDISON GONÇALVES
- BT/PMR/0205 – Desenvolvimento de um Sistema de Visão Omnidirecional – VALDIR GRASSI JUNIOR, JUN OKAMOTO JUNIOR
- BT/PMR/0206 – Estudo e Implementação de um *Rainbow Range Finder* – LEIJI NISHIDA, OSWALDO HORIKAWA
- BT/PMR/0207 – Análise da Influência da Deformação Plástica sobre a Taxa de Corrosão do Cobre Eletrolítico em Água Pura – BYUNG SIK KIM, EDISON GONÇALVES
- BT/PMR/0208 – Projeto de Mecanismos Flexíveis usando o Método de Otimização Topológica – CÍCERO RIBEIRO DE LIMA, EMILIO CARLOS NELLI SILVA
- BT/PMR/0209 – Análise Experimental e Numérico-Computacional da Cravação de Âncoras em Solos Argilosos – GUSTAVO RIBEIRO TESTA, EDISON GONÇALVES
- BT/PMR/0210 – Caracterização de Líquidos por Ultra-Som – RICARDO TOKIO HIGUTI, JULIO CEZAR ADAMOWSKI
- BT/PMR/0211 – Desenvolvimento Holografia Digital com Aplicação na Análise de Deslocamento em Condições Submersas – JOSÉ LUIS VALIN RIVERA, EDISON GONÇALVES
- BT/PMR/0212 - Metodologia para Detecção e Tratamento de Falhas em Sistemas de Manufatura Através de Redes de Petri -LUI ALBERTO MARTINEZ RIASCOS, PAULO EIGI MIYAGI
- BT/PMR/0213 – Sistema para Flexão de Joelho Mecanizada em Órteses de Membros Inferiores – MARKO ACKERMANN, FÁBIO G. COZMAN
- BT/PMR/0214 – Projeto, Construção e Testes de um Redutor Planetário de Duplo Estágio a Tração – FABRÍCIO SUNAHARA NAGAHASHI, EDUARDO LOBO LUSTOSA CABRAL
- BT/PMR/0215 – Desenvolvimento de Técnicas de Caracterização de Transdutores Piezelétricos – GILDER NADER, JULIO CEZAR ADAMOWSKI
- BT/PMR/0301 – Análise de Confiabilidade de um Processo de Manufatura – Aplicação para um Processo de Furação de Precisão – DIEGO MANREZA TOMAZ, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0302 – Aplicação de Técnicas de Confiabilidade para Avaliação do Desempenho da Estufa de um Sistema de Pintura – RICHARD TUVACRK, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0303 – Confiabilidade na Estruturação de Carrocerias de Veículos – DENIS LUQUE MARTINHS, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0304 – Análise de Confiabilidade de Sistemas Hidráulicos Empregados em Prensas: Aplicação para uma Prensa de Repuxo de 250 Toneladas – ÁLVARO CAMARGO PRADO, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0305 – Sistematização do Projeto do Controle de Sistemas Produtivos – FRANCISCO YASTAMI NAKAMOTO, DIOLINO JOSÉ DOS SANTOS FILHO
- BT/PMR/0306 – Análise da Influência dos Carregamentos Cíclicos sobre o Comportamento Mecânico das Camadas Poliméricas dos Cabos Submarinos Eletro Ópticos – NELSON CÁRDENAS OLIVIER, EDISON GONÇALVES
- BT/PMR/0307 – Geração de Redes Bayesianas Uniformemente Distribuídas – JAIME SHINSUKE IDE, FABIO GAGLIARDI COZMAN
- BT/PMR/0308 – Qualidade Aplicada ao Processo de Colagem de Vidros em Veículos Automotivos – RICARDO CAMILLO, JOÃO PAULO PEREIRA MARCIANO
- BT/PMR/0309 – Desenvolvimento de um Sistema Computacional para Avaliação Probabilística da Vida em Fadiga de Componentes Mecânicos – FERNANDO DE FREITAS, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0310 – Redes Neurais CMAC Paramétricas : Fundamentos e Aplicações em Processamento de Sinais e Controle de Processos – PAULO EDUARDO MACIEL DE ALMEIDA, MARCELO GODOY DE SIMOÕES

- BT/PMR/0311 – Desenvolvimento de um Sistema para Análise Automática do Padrão de Franjas Fotoelásticas Isocromáticas – ANTONIO FRANCISCO GENTIL FERREIRA JUNIOR, OSWALDO HORIKAWA
- BT/PMR/0312 – Simulated Annealing Aplicado ao Planejamento de Caminhos para Robôs – VINICIUS RODRIGUES DE MORAES, MARCOS DE SALES GUERRA TSUZUKI
- BT/PMR/0313 – Investigações sobre o Acabamento Superficial de Usinagens com Altíssima Velocidade de Corte – IAN FACCIO, MARCO STIPKOVIC FILHO
- BT/PMR/0314 – Classificação de Defeitos em Tubos de Gerador de Vapor de Plantas Nucleares Utilizando Mapas Auto-Organizáveis – ROBERTO NAVARRO DE MESQUITA, EDUARDO LOBO LUSTOSA CABRAL
- BT/PMR/0315 – Especificando Requisitos de Sistemas com Redes de Petri³ - ESTON ALMANÇA DOS SANTOS, JOSÉ REINALDO SILVA
- BT/PMR/0316 – Estudo do Processo de Furação em Aço com Brocas Helicoidais de Aço Rápido – ÉD CLAUDIO BORDINASSI, MARCO STIPKONIC FILHO
- BT/PMR/0317 – Projeto e Implementação de uma Plataforma de Testes Aplicada ao Desenvolvimento de Veículos Submarinos não Tripulados – FLÁVIO JOSÉ AGUIAR SOARES, ETTORE APOLONIO DE BARROS
- BT/PMR/0318 – Modelagem e Controle de Veículos não Tripulados – ERIC CONRADO DE SOUZA, NEWTON MARUYAMA
- BT/PMR/0319 – Projetos de Atuadores Piezelétricos Flexensionais Usando o Método de Otimização Topológica – RONNY CALIXTO CARBONARI, EMILIO CARLOS NELLI SILVA
- BT/PMR/0320 – Avaliação de Risco de Sistemas Mecânicos: Aplicação para o Sistema Centralizado de Suprimento de Oxigênio em Estabelecimentos Assistenciais de Saúde – CARLOS ROBERTO BURRI, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0321 – Development of Dynamically Tuned Gyroscope – DTG – FERNANDO DE CASTRO JUNQUEIRA, ETTORE APOLONIO DE BARROS
- BT/PMR/0322 – Contribuição para a Otimização da Usinagem de Cavidades 2 ½ D – LEANDRO COSTA DE OLIVEIRA, MARCOS DE SALES GUERRA TSUZUKI
- BT/PMR/0323 – Desenvolvimento de uma Máquina de Compressão-Torção para Medição do Atrito em Processos de Conformação de Chapas Metálicas – CARLOS EDDY VALDEZ SALAZAR, GILMAR FERREIRA BATALHA
- BT/PMR/0324 – Reconhecimento Automático de Locutor em Modo Independente de Texto por Self-Organizing Maps – ALEXANDRE TEIXEIRA MAFRA, MARCELO GODOY SIMÕES
- BT/PMR/0325 – Selecting Exact and Approximate Algorithms in Decomposed Bayesian Networks – FABIO T. RAMOS, FABIO G. COZMAN
- BT/PMR/0326 – Desenvolvimento de um Sistema Acústico de Posicionamento Submarino – FREDERICO VINES FARIA DE LIMA, CELSO MASSATOSHI FURUKAWA
- BT/PMR/0327 – Estimação de Coeficientes Hidrodinâmicos de um Veículo Submarino Semi-Autônomo – JUAN P. JULCA AVILA, JULIO CESAR ADAMOWSKI
- BT/PMR/0328 – Simulação Numérica via Elementos Finitos da Conformação de Chapas Metálicas por Pressão Hidráulica (BULGE TEST) – JORGE PAIVA ABRANTES, GILMAR FERREIRA BATALHA
- BT/PMR/0329 – Avaliação do Potencial de Utilização de um Mecanismo Paralelo como Máquina Ferramenta – GUILHERME RASZL, TARCÍSIO ANTONIO HESS COELHO
- BT/PMR/0401 – Desenvolvimento de um Sistema Robótico para Cirurgias Minimamente Invasivas – WALTER DE BRITTO VIDAL FILHO, LUCAS ANTONIO MOSCATO
- BT/PMR/0402 - Representação de Curvas e Superfícies em Modeladores de Sólido B-Rep – WANG CONGLI, MARCOS DE SALES GUERRA TSUZUKI
- BT/PMR/0403 – Controle de um Sistema de Teleoperação Mestre-Escravo com Retorno de Força Aplicado à Robótica Cirúrgica – LUIZ FELIPE DE CASTRO E SILVA VIDAL, LUCAS ANTONIO MOSCATO
- BT/PMR/0404 – A Petri-Net Based Object-Oriented Approach for the Modelling and Analysis of Hybrid Systems – EMILIA VILLANI, PAULO EIGI MIYAGE
- BT/PMR/0405 – Plastic Buckling of Open Shells Under Impact Loads – SÉRGIO SEIJI TERAMOTO, MARCILIO ALVES
- BT/PMR/0406 – Construção de Mapas de Ambiente para Navegação de Robôs Móveis com Visão Omnidirecional Estéreo – CLÁUDIA CRISTINA GHIRARDELLO DECCÓ, JUN OKAMOTO JUNIOR
- BT/PMR/0407 – Modelos Constitutivos de Atrito Bi-Dimensionais para Condições de Deslizamento Isotrópicas, Cinemáticas e Anisotrópicas – RENATO TEIXEIRA VARGAS, JOÃO BATISTA AGUIAR
- BT/PMR/0408 – Critério de Fadiga Multiaxial para Análise de Fadiga de Alto Ciclo – ACCACIO MORAES NETO, GILBERTO FRANCISCO MARTHA DE SOUZA

- BT/PMR/0409 – Axiomatizing Noisy-OR – FABIO GAGLIARDI COZMAN
- BT/PMR/0410 – Uma Introdução aos Filtros de Partículas – VICTOR AUGUSTO FERNANDES DE CAMPOS, NEWTON MARUYAMA
- BT/PMR/0411 – Proposta de Utilização da Metodologia E-MFG/PFS para a Sistematização do Planejamento de Dispositivos de Fixação – MIGUEL NAVARRO MENA, DIOLINO JOSÉ DOS SANTOS FILHO
- BT/PMR/0412 – Manutenção Baseada em Confiabilidade: Ciência e Prática – IDÉLCIO ALEXANDRE PALHETA CARDOSO, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0413 - Design of a Split Hopkinson Pressure Bar - FELIPE GALLINA, MARCILIO ALVES
- BT/PMR/0414 - Análise Numérico-Computacional das Tensões Residuais Induzidas pelo Jateamento com Granalha - MIGUEL ANGEL CALLE GONZALES, EDISON GONÇALVES
- BT/PMR/0415 – Desenvolvimento de um Novo Conceito de Produto: Sensor de Nível de Combustível para Tanques de Motocicletas – NELSON SHOJI NISHIMURA, LUCAS ANTONIO MOSCATO
- BT/PMR/0416 - Estudo do Processo de Hidroconformação de Tubos - CRISTIANE SANTONI, GILMAR FERRERIRA BATALHA
- BT/PMR/0417 - Análise Experimental da Confiabilidade da Ferramenta no Processo de Furação - CARMEN ELENA PATIÑO RODRIGUEZ, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0418 - Método de Degeneração para o Projeto do Controle de Sistemas Produtivos - JULIO ARAKAKI, PAULO EIGI MIYAGI
- BT/PMR/0501 – Projeto de Sistemas de Controle Modulares e Distribuídos – ANDRÉ CÉSAR MARTINS CAVALHEIRO, DIOLINO JOSÉ DOS SANTOS FILHO
- BT/PMR/0502 – Formal Verification of the Rafale Landing System –EMILIA VILLANI, PAULO EIGI MIYAGI
- BT/PMR/0503 – Ambiente Integrado de Análise de Sistemas de Manufatura – WILSON M. ARATA, PAULO EIGI MIYAGI
- BT/PMR/0504 – Modelagem de Sistemas de Controle Distribuídos e Colaborativos de Sistemas Produtivos – CRISTINA TOSHIE MOTOHASHI MATSUSAKI, DIOLINO JOSÉ DOS SANTOS FILHO
- BT/PMR/0505 – Metodologia para a Modelagem de Sistemas C³I como SEDS – ROGER VALENCIA REYES, FABRÍCIO JUNQUEIRA, PAULO EIGI MIYAGI
- BT/PMR/0506 – Aprendizado Semi-Supervisionado de Classificadores Bayesianos Utilizando Testes de Independência – MARCELO CESAR CIRELO, FABIO GAGLIARDI COZMAN
- BT/PMR/0507 – Recuperação de Forma de Peças Estampadas por Luz Estruturada – JULIO TABEGNA, OSWALDO HORIKAWA
- BT/PMR/0508 – Construção de uma Rede Bayesiana Aplicada ao Diagnóstico de Doenças Cardíacas – ANDRÉ HIDEAKI SAHEKI, FABIO GAGLIARDI COZMAN
- BT/PMR/0509 – Grades de Evidência com Visão Omnidirecional Estéreo – FABIANO ROGÉRIO CORRÊA, JUN OKAMOTO JUNIOR
- BT/PMR/0510 - Leis de Similaridade Aplicadas a estruturas Sob Impacto – ROBERTO EIKI OSHIRO, MARCÍLIO ALVES
- BT/PMR/0511 - Algoritmos para Inferência em Redes Credais – JOSÉ CARLOS FERREIRA DA ROCHA, FÁBIO GAGLIARDI COZMAN
- BT/PMR/0512 - Desenvolvimento de Método Holográfico para Inspeção de Estruturas Submersas – RODRIGO DE SOUZA DIAS, EDISON GONÇALVES
- BT/PMR/0513 – Proposta de Procedimento para Análise de Fadiga de Molas Planas – DANIEL MARETTI DE CARVALHO, EDISON GONÇALVES
- BT/PMR/0514 – Análise Teórico-Experimental da Corrosão do Cobre em Sistemas de Refrigeração: Modelagem Eletroquímica e Numérico-Computacional – GELSON GOMES FALCÃO FILHO, EDISON GONÇALVES
- BT/PMR/0601 – Algoritmo de Reconstrução de Imagens Tridimensionais Através da Tomografia por Impedância Elétrica Baseado no Método de Otimização Topológica – LUIZ AUGUSTO MOTTA MELLO , EMILIO CARLOS NELLI SILVA
- BT/PMR/0602 – Derivadas em Redes Bayesianas Usando Eliminação de Variáveis- MATHIAS JUAN PERAZZO, FÁBIO GAGLIARDI COZMAN
- BT/PMR/0603 – Machine Overhaul and/or Retrofitting: A Reliability Approach – GUILHERME EDUARDO DA CUNHA BARBOSA, GILBERTO FRANCISCO MARTHA DE SOUZA
- BT/PMR/0604 – Desenvolvimento de Processos de Conformação de Chapas usando Simulação Computacional como Ferramenta Integrada no Desenvolvimento de Carrocerias Automáticas- GLEITON LUIZ DAMOULIS, GILMAR FERREIRA BATALHA
- BT/PMR/0605 – Interferometria Holográfica e Processamento de Imagens Aplicados à Análise de Deformações – GILVAN BARBOSA DA SILVA, EDISON GONÇALVES
- BT/PMR/0606 – Proposta de Algoritmo para a Redução da Malha de Polígonos em Modelos Sólidos B-Rep- WANG GONGSHI, MARCOS DE SALLES GUERRA TSUZUKI

- BT/PMR/0607 – Inteligência de Exames Aplicada na Síntese de Estratégias de Caça em ambiente Tridimensional – EMILIANO GONÇALVES DE CASTRO, MARCOS DE SALLES GUERRA TSUZUKI
- BT/PMR/0608 – Estimacão de Trajetórias Terrestres Utilizando Unidade de Medição Inercial de Baixo Custo e Fusão Sensorial- DOUGLAS DANIEL SAMPAIO SANTANA, CELSO MASSATOSHI FURUKAWA
- BT/PMR/0609 – An open Distributed Systems Approach for Building Automation – GLADYS DEIFAN BASTIDAS GUSTÍN, PAULO Eigi Miyagi
- BT/PMR/0610 – Projeto Otimizado de Micromotor um Eletrostático – HUMBERTO FERREIRA VINHAIS, EMILIO CARLOS NELLI SILVA
- BT/PMR/0611 – Fundamentos da Realimentação das Informações pela manufatura Informada – JOSÉ JEAN-PAUL ZANLUCCHI DE SOUZA TAVARES, JOSÉ REINALDO SILVA
- BT/PMR/0612 – Intra and Intermodel Consistency in the Representation of the Semantics of Heterogeneous Discrete Event Dynamic System Models- WILSON MUNEMASSA ARATA, PAULO EIGI MIYAGI
- BT/PMR/0613- Análise Numérica de um Transdutor Piezelétrico de Potência para Processamento de Termoplásticos Têxteis – JOÃO BATISTA DA SILVA, JULIO CEZAR ADAMOWSKI
- BT/PMR/0614 - Detecção de Falhas em Sistemas Flexíveis de Montagem Através de Rede de Petri- DAVI LIRA NUNEZ, PAULO EIGI MIYAGI
- BT/PMR/0615 – Extensão Sintática a Modelagem E-MFG com Comunicadores- GUY CLIQUET DO AMARAL FILHO, DIOLINO DOS SANTOS FILHO
- BT/PMR/06/16- Projeto de MEMS Eletrotermomecânicos Usando Otimizaçãp Topológica- WILFREDO MONTEALEGRE RÚBIO, EMILIO CARLOS NELLI SILVA
- BT/PMR/06/17- Modelagem das Tensões Residuais no Processo de Torneamento Duro de um Aço DIN 100CrMn6 e suas Correlações com os Esforços de Corte- SERGIO DELIJAICOV , JOÃO BATISTA DE AGUIAR
- BT/PMR/0618 – Mancais Magnéticos Híbridos do Tipo Atração com Controle Uniaxial – ISAIAS DA SILVA, OSWALDO HORIKAWA
- BT/PMR/0619 – The Inferencial complexity of Credal Networks – CASSIO POLPO DE CAMPOS, FÁBIO GAGLIARDI COZMAN
- BT/PMR/0620 –IPE and L2U: Approximate Algorithms for Credal Networks– JAIME SHINSUKE IDE, FÁBIO GAGLIARDI COZMAN
- BT/PMR/0621- Análise Elastoplástica do Colapso de Elementos Tubulares – MIGUEL JAIME SANDOVAL RODRIGUEZ-, JOÃO BATISTA DE AGUIAR
- BT/PMR/0622 – Determinação das Constantes Elásticas de Materiais Compostos por Ultra-Som – DANIEL VERGA BOERI, JULIO CEZAR ADAMOWSKI

Escola Politécnica da USP - Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos
Edifício de Engenharia Mecânica/Naval - Av. Prof. Mello Moraes, 2231
Cidade Universitária - CEP 05508-030 - Caixa Postal 61548 - São Paulo - SP- Fone: (0 XX 11) 3091-5337
Fax: (0 XX 11) 3091-5461