



APLICAÇÃO DE ALGORITMOS DE OTIMIZAÇÃO NA DETECÇÃO DE CONTATO ENTRE SUPERQUÁDRICAS CONVEXAS

Rônaldy Sousa Oliveira

Programa de Pós-Graduação em Ciência da Computação, UFPI,
ronaldy@ifpi.edu.br

Paulo Sérgio Marques dos Santos

Departamento de Matemática, UFPI,
psergio@ufpi.edu.br

André Macedo Santana

Departamento de Computação, UFPI,
andremacedo@ufpi.edu.br

RESUMO

Apresentamos um algoritmo usado para detectar o contato e encontrar a distância entre dois objetos convexos. Esta distância é definida como o comprimento do caminho mais próximo, ligando um objeto ao outro. O problema é formulado como um problema de otimização convexo, com restrições não lineares, que é resolvido por um método de penalidades. Relatamos experiências numéricas que ilustram o comportamento do método proposto.

PALAVRAS CHAVE. Programação Não Linear, Problema de Otimização, Detecção de Contato.

Área Principal. PM - Programação Matemática.

ABSTRACT

We present an algorithm used to detect contact and find the distance between two convex objects. This distance is defined as the length from the nearest road linking an object to another. The problem is formulated as a convex optimization problem with nonlinear constraints is solved by a method of penalties. We describe experiments that illustrate the numerical behavior of the proposed method.

KEYWORDS. Nolinear Programming, Optimization Problem, Contact Detection.

Main Area. PM - Mathematical Programming.

1. Introdução

Detectar colisões tem sido de grande importância na simulação de movimentos realistas. Determinar a distância mínima entre dois objetos convexos é um aspecto importante na área da robótica, processamento de imagens, sistemas CAD e outras áreas de processamento de informação que lidam com dados geométricos. O planejamento de trajetória automatizado necessita da capacidade do cálculo da distância, para detectar se houve colisão ou não, reduzindo o tempo total necessário para maioria dos algoritmos de planejamento de percursos. Em geral, um algoritmo de planejamento de trajetória precisa saber não só se ocorreu colisão, mas também, caso não tenha ocorrido, o quanto está próximo de ocorrer. Um exemplo seria um tipo de algoritmo de planejamento de caminho, "generate-and-test", que requer testes para saber se a configuração e a trajetória são livres de colisão, segundo Burckley (1989), ou a medida de quanto uma parte do robô está de colidir com um obstáculo.

Objetos rígidos não devem se intercer e, quando colidem, a resposta precisa ser tratada adequadamente. Detectar contato é crucial para jogos de computador 3D, animação por computador, etc, ver Choi et al. (2009). Um dos principais requisitos para modelar a geometria de um objeto em 3D, neste caso, a superfície exterior de um corpo rígido, é o uso de uma descrição matemática que proporciona uma elevada representatividade geométrica, unida a um compacto, controlável e intuitivo conjunto de parâmetros.

De acordo com Barr (1981), superfícies superquádricas são entidades geométricas que fornecem tal descrição para uma variedade de formas. Para os sistemas mecânicos que apresentam formas livres, superfícies superquádricas também podem ser aplicadas por encontrarem o melhor ajuste para o conjunto de pontos que pertence à superfície, segundo Li (2007) e Bardin et al. (1995).

Este artigo apresenta um algoritmo para detecção de contato utilizando superquádricas. Ele é aplicável a qualquer tipo de formas convexas e oferece uma relação entre os pontos da superfície e os vetores normais. O contato é detectado quando a distância mínima é igual a zero e positiva quando as superfícies estiverem separadas.

A escolha deste trabalho deve-se, principalmente, à possibilidade de melhorar o tempo de resposta dos algoritmos apresentados em Lopes et al. (2010), encontrando resultados bastante satisfatórios ao ser comparado com o citado na literatura.

2. Superfícies Superquádricas

Superquádricas são uma família de formas paramétricas que foram descobertas pelo dinamarquês Peit Hein, ver Gardiner (1965), como uma extensão de quádricas básicas e sólidas. Elas têm sido utilizadas, ou propostas, para serem usadas como primitivas na representação de formas em computação gráfica, segundo Barr (1981), e visão computacional, segundo Pentland (1986), Solina e Bajcsy (1990), Terzopoulos e Metaxas (1991). Superquádricas desempenham o papel de partes de protótipos e podem ser ainda mais deformadas e coladas umas às outras em modelos de aparência realista, ver Pentland (1986).

Segundo Barr (1981), uma superfície superquádrica é definida por uma equação implícita chamada *inside-outside* apresentada na Equação (1):

$$h(x, y, z) = 0 \Leftrightarrow F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_1}{\epsilon_2}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} \quad (1)$$

Os parâmetros a_1 , a_2 e a_3 especificam as dimensões das superquádricas. Os expoentes ϵ_1 e ϵ_2 controlam a perpendicularidade das superquádricas no plano x, y e direção z , respectivamente. Cada ponto (x, y, z) , que satisfaz $F(x, y, z) = 1$, está na superfície da superquádrica;

se $F(x, y, z) < 1$, então o ponto está no interior da superquádrica e se $F(x, y, z) > 1$, então o ponto está fora da superquádrica. Além disso, segundo Barr (1981), é possível descrever a superfície de uma superquádrica na forma parametrizada:

$$r(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega \leq \pi \end{matrix}$$

O vetor \vec{r} é originado no centro da coordenada e varre uma superfície fechada no espaço quando os dois parâmetros independentes, ângulos η e ω , mudam nos intervalos indicados. ω é o ângulo entre o eixo x e a projeção do vetor \vec{r} no plano $x-y$, enquanto η é o ângulo entre o vetor \vec{r} e sua projeção no plano $x-y$. Os parâmetros η e ω correspondem aos ângulos de latitude e longitude do vetor \vec{r} .

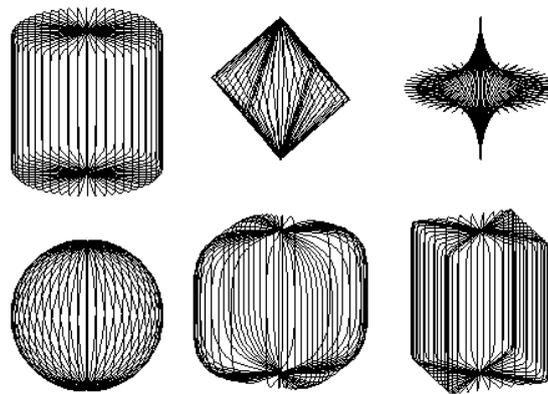


Figure 1: Exemplos de superfícies superquádricas.

3. Formulação do Problema

Dados dois objetos convexos S_A e S_B , cujas posições globais e orientações são conhecidas, apresentamos uma formulação para calcular o problema da distância mínima entre esses corpos. Para calcular a distância mínima entre dois pontos, precisamos encontrar um par de pontos, um em cada objeto, tal que a distância entre os dois pontos seja igual ou menor a distância entre qualquer outro par. Em nossa abordagem, descrevemos as superfícies dos objetos como superquádricas e um objeto não está completamente inscrito no outro. A distância entre duas superquádricas é então definida como a menor distância entre os dois pontos A e B, onde A está em S_A e B está em S_B , conforme a seguinte relação:

$$d = \min d(A, B) \tag{2}$$

tal que

$$A \in S_A \text{ e } B \in S_B \tag{3}$$

onde d é a distância entre os objetos S_A e S_B .

O vetor distância \vec{d}_{AB} , que liga os pontos das superfícies, é definido como a distância euclidiana entre os pontos A e B, como representado na Figura 2. Assim, o vetor distância é escrito como:

$$\vec{d}_{AB} = \| A - B \| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2} \tag{4}$$

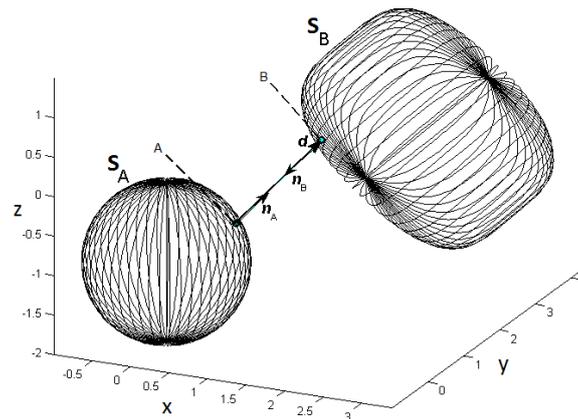


Figure 2: Duas superquádricas S_A e S_B com os pontos A e B, os vetores normais \vec{n}_A e \vec{n}_B e o vetor distância \vec{d} .

O problema da distância mínima consiste de determinar os pontos das superfícies que compartilham uma normal e onde a distância entre as superfícies é mínima. As normais indicam as condições necessárias para que dois pontos formem um par de contato: os vetores normais, nestes pontos, devem ser colineares um ao outro e colineares ao vetor distância que liga os dois pontos. O vetor normal a cada ponto, \vec{n}_A e \vec{n}_B , é obtido a partir da derivada parcial da equação implícita da superfície. Se a representação da superfície implícita, dada na Equação (1), é considerada, então, o vetor normal, a um determinado ponto, é o gradiente da superfície avaliado nesse ponto, ver Lopes et al. (2010). A formulação matemática para o contato está relacionada à representação da superfície. A representação da superfície implícita e paramétrica é descrita da seguinte forma:

$$S_A = \{(x, y, z) \in \mathbb{R}^3 \mid F_A(x, y, z) = 1\} \quad (5)$$

onde F_B é definida de módulo análogo, e são funções escalares implícitas que definem onde os pontos que pertencem às superfícies S_A e S_B estão localizados. Em geral, três são as equações não lineares usadas para definir um conjunto de restrições geométricas para o cálculo da distância mínima:

1. restrição ortogonal (relativa às restrições entre dois vetores perpendiculares);
2. restrição colinear (relativa às restrições entre dois vetores paralelos);
3. restrição *locus* (garante que os pontos permaneçam sobre a superfície de contato).

Dada a representação implícita da superfície (Equação 1), o vetor normal de uma superfície superquádrica é a sua derivada parcial. O vetor normal é, portanto, dado por:

$$\vec{n}(x, y, z) = \nabla h(x, y, z) = (n_x, n_y, n_z) \quad (6)$$

Como já informado, o cálculo da distância mínima, entre as superfícies, consiste em encontrar a localização do par de pontos A e B, em que o vetor distância \vec{d}_{AB} está alinhado com os vetores normais, \vec{n}_A e \vec{n}_B , e possui um valor mínimo da norma euclidiana. A condição de colinearidade, entre esses vetores, pode ser escrita como duas equações do produto vetorial, relacionando os vetores \vec{d}_{AB} , \vec{n}_A e \vec{n}_B , ver Pombo et al. (2007). Essas equações são definidas:

i) Os vetores normais \vec{n}_A e \vec{n}_B , dos candidatos a pontos de contato, tem que ser paralelos:

$$\vec{n}_A \times \vec{n}_B = 0 \quad (7)$$

$$\vec{n}_A \cdot \vec{n}_B = -1 \quad (8)$$

ii) O vetor \vec{d}_{AB} , que representa a distância entre os candidatos a pontos de contato, tem que ser paralelo ao vetor normal \vec{n}_A ou \vec{n}_B . Esta condição pode ser escrita da seguinte forma:

$$\vec{n}_A \cdot \vec{d}_{AB} = 1 \quad (9)$$

$$\vec{n}_B \cdot \vec{d}_{AB} = -1 \quad (10)$$

A condição de intersecção estabelece que o contato entre as superfícies acontece e, portanto, a condição de contato é calculada a partir da projeção seguinte:

$$\vec{d}_{AB} \cdot \vec{n}_A \leq 0 \quad (11)$$

3.1. Detecção de Contato

O vetor distância, de magnitude d , é calculado como a distância euclidiana do vetor \vec{d}_{AB} , portanto, considerando $\mathbf{d} = \|\cdot\| \geq 0$. Em algum momento, a magnitude d indicará uma das três possíveis situações, de acordo com o valor da distância mínima, conforme demonstrada na Figura 3.

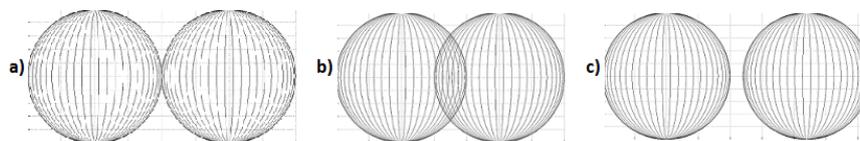


Figure 3: Situações de contato das superfícies: a) Contato em um único ponto; b) Contato em mais de um ponto; c) Sem contato.

3.2. Problema de Otimização

Sendo duas superquádricas, S_A e S_B , o problema de otimização tem como objetivo encontrar os pontos $A \in S_A$ e $B \in S_B$, que minimizam a distância \vec{d} (3), entre essas superfícies, sujeito às restrições (5), (7), (8), (9) e (10).

A restrição (5) garante que os pontos estejam na superfície das superquádricas, enquanto as restrições (7), (8), (9) e (10), em caso de contato, garantem que a solução indica a interferência máxima entre os corpos.

Após definidas as restrições e a função objetivo, o problema de otimização é modelado, segundo Portal et al. (2010), da seguinte forma:

$$\begin{aligned} &\text{minimizar} && \|\mathbf{A} - \mathbf{B}\|^2 \\ &\text{Sujeito a:} && h_A(x_A, y_A, z_A) = 0 \\ & && h_B(x_B, y_B, z_B) = 0 \\ & && \vec{n}_A \times \vec{n}_B = 0 \text{ ou } \vec{n}_A \cdot \vec{n}_B = -1 \\ & && \vec{n}_A \cdot \vec{d}_{AB} = 1 \text{ ou } \vec{n}_B \cdot \vec{d}_{AB} = -1 \end{aligned} \quad (12)$$

4. Descrição do Algoritmo

Apresentamos um algoritmo simples e rápido para detectar colisão entre objetos convexos. O algoritmo encontra uma distância separando dois corpos, caso não estejam em colisão. Além de o algoritmo verificar se há um plano separador entre as superfícies, caso não encontre, determina se as superfícies envolvidas estão em contato.

O método consiste na transformação do original restrito em um problema sem restrições. Para isso, definimos uma nova função objetivo, adicionando uma função auxiliar à função objetivo. Esse método gera uma sequência de pontos infactíveis, cujo limite tende à solução ótima do problema. A partir do problema (12) definimos:

$$g_{c,\alpha}(A, B) = c_1 \|h_A(A)\|^{\alpha_1} + c_2 \|h_B(B)\|^{\alpha_2} + c_3 \|\vec{n}_A \times \vec{n}_B\|^{\alpha_3} + c_4 \|\vec{A}B_A \cdot \vec{n}_A\|^{\alpha_4} \quad (13)$$

E consideramos o problema:

$$\begin{aligned} &\text{minimizar} && \|A - B\|^2 \\ &\text{Sujeito a:} && cg_\alpha(A, B) = 0, \text{ onde } c \geq 0. \end{aligned} \quad (14)$$

O método(14) consiste em uma estratégia de penalidades, ver Bazarra et al. (1993), e nosso algoritmo é mostrado a seguir:

1. No experimento, estabelecer o erro de convergência ($\epsilon > 0$), o ponto inicial x^0 é obtido a partir de (15) e (16), o parâmetro de penalidade $c^0 = (c_1, c_2, c_3, c_4)$, o fator de incremento de penalidade $\beta > 0$ e $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, $k = 0$.
2. Resolver o problema utilizando um método de minimização irrestrita:

$$\text{minimizar } f(A, B) + c^k g_\alpha(A, B)$$

obtendo então x^{k+1} .

3. Se $c^k g_\alpha(A, B) < \epsilon$, pare, pois a solução ótima foi encontrada, caso contrário, vá para o passo 4.
4. Fazer $c^{k+1} = \beta c^k$
5. $k = k + 1$, voltar ao passo 2.

Verificamos, à medida que $c \rightarrow \infty$ e $g_\alpha(A, B) \rightarrow 0$, que sequência formada pelas soluções do problema penalizado converge para a solução do problema original. É importante ressaltar que a escolha inicial do fator de penalidade e do parâmetro de penalidade afeta a convergência do método.

Para garantir que a solução adequada seja determinada, deve-se estimar uma posição inicial, aproximada, que esteja perto da solução desejada, ver Lopes et al. (2010). Esta estimativa está nos vetores que ligam as origens das duas superfícies, desviando-se de cada origem um certo percentual dado pelo fator δ_A ou δ_B , respectivamente, ver Portal et al. (2010). As posições iniciais para as superfícies A e B são escritas em Portal et al. (2010), conforme as equações (15) e (16).

$$\begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} = \begin{bmatrix} p_{x_A} + \delta_A(p_{x_B} - p_{x_A}) \\ p_{y_A} + \delta_A(p_{y_B} - p_{y_A}) \\ p_{z_A} + \delta_A(p_{z_B} - p_{z_A}) \end{bmatrix} \quad (15)$$

e

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} p_{x_B} - \delta_B(p_{x_B} - p_{x_A}) \\ p_{y_B} - \delta_B(p_{y_B} - p_{y_A}) \\ p_{z_B} - \delta_B(p_{z_B} - p_{z_A}) \end{bmatrix} \quad (16)$$

5. Resultados Experimentais

A métrica utilizada para avaliar o método de Penalidade consistiu em aplicar o método em um conjunto de superquádricas convexas. Para esse conjunto de superquádricas, observaram-se avaliações da função objetivo, o tempo despendido para encontrar a distância e os valores dos pontos das superfícies dessas superquádricas. Utilizaram-se diferentes parâmetros para o método. O código de detecção de contato foi desenvolvido em MATLAB 7.10.0 (R2010A), sendo executado em um computador com processador Core(TM) 2 Duo da Intel(R) 2.20 e 4 GB de memória RAM.

5.1. Resultados Numéricos

Tabela 1: Propriedades das Superquádricas e Posições, casos de 1 a 6. (Observar que $i=1,2,3$ e $j=1,2$)

casos	$(p_x, p_y, p_z)_A$	$(a_i)_A$	$(2/\epsilon_j)_A$	$(e_i)_A$
	$(p_x, p_y, p_z)_B$	$(a_i)_B$	$(2/\epsilon_j)_B$	$(e_i)_B$
1	(0.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
	(4.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
2	(4.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
	(5.8, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
3	(0.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
	(0.0, 0.0, 4.0)	(1.2, 1.4, 1.8)	(2.6, 3.1)	(0.0, 0.0, 0.0)
4	(0.0, 0.0, 4.0)	(1.2, 1.4, 1.8)	(2.6, 3.1)	(0.0, 0.0, 0.0)
	(2.0, 1.5, 3.0)	(1.5, 1.0, 1.0)	(4.0, 2.2)	(0.0, 0.4, 0.25)
5	(4.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
	(2.0, 1.5, 3.0)	(1.5, 1.0, 1.0)	(4.0, 2.2)	(0.0, 0.4, 0.25)
6	(0.0, 0.0, 0.0)	(1.0, 1.0, 1.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
	(2.0, 1.5, 3.0)	(1.5, 1.0, 1.0)	(4.0, 2.2)	(0.0, 0.4, 0.25)

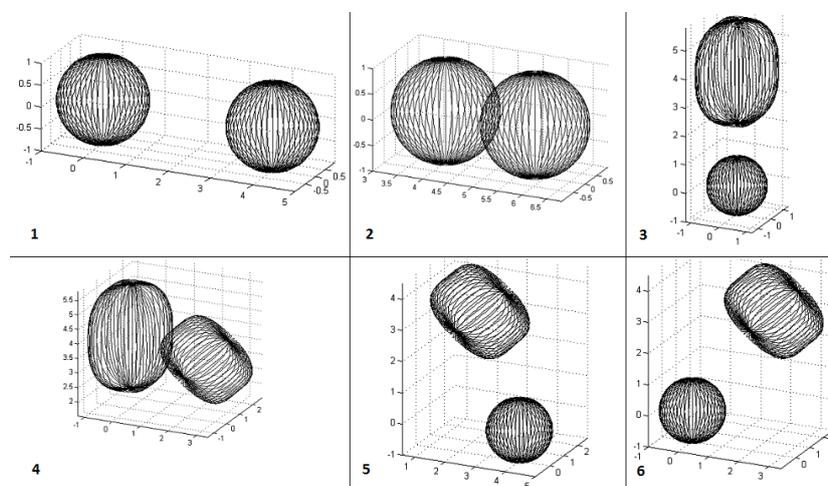


Figure 4: Pares de superquádricas referentes aos casos de 1 a 6.

Tabela 2: Testes dos pares de contato para o cálculo da distância mínima ou penetração máxima entre superfícies superquádricas convexas para os casos de 1 a 6. Parâmetros: $c^0 = (0.001, 0.001, 0.001, 0.001)$, $\beta = 0.2$ e $\alpha = (0.5, 0.5, 0.5, 0.5)$.

casos	rest.	PI	PQS	MP
		$d(\mathbf{p}, \mathbf{q})$ -tempo[ms]-iter.	$d(\mathbf{p}, \mathbf{q})$ -tempo[ms]-iter.	$d(\mathbf{p}, \mathbf{q})$ -tempo [ms]-iter.
1	6	2.000(d)-815-0	2.000(d)- 8	1.950(d)-684-1
	6,8	2.000(d)-823-0		1.950(d)-679-1
	6,8,10	2.000(d)-679-0		1.950(d)-683-1
2	6	0.200(c)-1.005-0	0.200(c)- -1	0.039(d)-668-3
	6,8,10	0.000(d)-1.337-24		0.039(d)-685-3
	6,10,11	0.000(d)-1.456-56		0.039(d)-689-3
3	6	1.200(d)-824-0	1.200(d)- -3	1.150(d)-676-1
	6,9	1.200(d)-810-0		1.150(d)-672-1
	6,9,11	1.200(d)-816-0		1.150(d)-675-1
4	6	0.381(c)-1.120-0	0.381(c)- -11	0.318(c)-684-1
	6,8	0.410(c)-1.084-5		0.290(c)-686-2
	6,10,11	0.061(d)-1.985-127		0.263(c)-684-3
5	6	1.590(d)-1.121-8	1.587(d)- -4	1.620(d)-670-1
	6,9,11	1.589(d)-2.190-133		1.513(d)-683-3
	6,10,11	1.589(d)-1.170-18		1.513(d)-683-3
6	6	1.821(d)-1.113-8	1.821(d)- -8	1.836(d)-674-1
	6,9,10	1.820(d)-1.284-32		1.805(d)-682-2
	6,9,11	1.820(d)-1.140-15		1.805(d)-684-2

Tabela 3: Propriedades das Superquádricas e Posições referentes aos casos das Tabelas 4,5 e 6. (Observar que $i=1,2,3$ e $j=1,2$)

casos	$(p_x, p_y, p_z)_A$	$(a_i)_A$	$(2/\epsilon_j)_A$	$(e_i)_A$
	$(p_x, p_y, p_z)_B$	$(a_i)_B$	$(2/\epsilon_j)_B$	$(e_i)_B$
Tab 04	(-1.0, 2.0, 4.0)	(1.0, 1.25, 1.6)	(4.0, 2.0)	(-1.0, -0.2, 0.14)
	(0.5, 1.5, -3.0)	(2.0, 2.0, 2.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
Tab 05	(-1.0, 2.0, 1.0)	(1.0, 1.25, 1.6)	(4.0, 2.0)	(-1.0, -0.2, 0.14)
	(0.5, 1.5, -1.0)	(2.0, 2.0, 2.0)	(2.0, 2.0)	(0.0, 0.0, 0.0)
Tab 06	(3.0, -3.0, 2.5)	(1.0, 1.25, 1.6)	(4.0, 2.0)	(-1.0, -0.2, 0.14)
	(0.0, 0.0, 4.0)	(1.5, 1.0, 1.0)	(4.0, 2.2)	(0.0, 0.4, 0.25)

Esta seção compara o método proposto neste trabalho com os métodos demonstrados por Lopes et al. (2010). Os métodos PI (Método de Pontos Interiores) e PQS (Programação Quadrática Sequencial) são apresentados por Lopes et al. (2010) e MP (Método de Penalidade).

A Tabela 1 detalha as propriedades geométricas das superquádricas, mostradas na Figura 4, para os testes de contato da Tabela 2. Enquanto a Tabela 3 informa as propriedades das superquádricas para os testes das Tabelas de 4 a 6, representadas na Figura 5.

Podemos observar na Tabela 2, que o método proposto (MP) obteve bons resultados ao ser comparado com os outros dois métodos. Apesar de possuir quantidade de iteração

Tabela 4: Testes de contato com os parâmetros $c^0 = (0.02, 0.02, 0.01, 0.01)$, $\beta = 0.3$ e $\alpha = (0.5, 0.5, 2.5, 2.5)$ para o Método de Penalidade.

restrições	Métodos	$d(\mathbf{p}, \mathbf{q})$	tempo[ms]	iterações
6	PI	3.523192	1.132	13
	MP	3.531054	0.756	5
6,8	PI	3.528226	1.009	7
	MP	3.531054	0.676	5
6,8,10	PI	3.523607	2.147	132
	MP	3.531054	0.652	5
6,8,11	PI	3.523193	2.106	121
	MP	3.531054	0.620	5
6,9	PI	3.523192	1.053	21
	MP	3.531054	0.620	5
6,9,10	PI	3.523192	1.082	20
	MP	3.531054	0.668	5
6,9,11	PI	3.523192	1.125	27
	MP	3.531054	684	5
6,10,11	PI	3.523192	1.188	18
	MP	3.531054	0.757	5

Tabela 5: Testes de contato com os parâmetros $c^0 = (0.01, 0.01, 0.01, 0.01)$, $\beta = 0.2$ e $\alpha = (2.5, 2.5, 2.5, 2.5)$ para Método de Penalidade para o mesmo par de superquádricas.

restrições	Métodos	$d(\mathbf{p}, \mathbf{q})$	tempo[ms]	iterações
6	PI	0.000000 (contato)	1.165	15
	MP	1.071021 (contato)	690	1
6,8	PI	1.107645 (contato)	1.115	7
	MP	1.064035 (contato)	707	2
6,8,10	PI	0.000004 (distante)	1.684	56
	MP	1.062875 (contato)	694	3
6,8,11	PI	0.000040 (distante)	1.678	54
	MP	1.062875 (contato)	717	3
6,9	PI	1.092449 (contato)	1.297	32
	MP	1.071021 (contato)	709	1
6,9,10	PI	0.000035 (distante)	1.513	52
	MP	1.060393 (contato)	696	5
6,9,11	PI	1.084320 (contato)	1.002	1
	MP	1.060393 (contato)	698	5
6,10,11	PI	0.048047 (distante)	1.899	134
	MP	1.060401 (contato)	699	5

maior do que o PI, em alguns casos, o tempo gasto para convergência no MP foi inferior ao do PI. Nota-se, no caso 4, que nosso algoritmo detectou o contato em todos os conjuntos de restrições, não acontecendo o mesmo com o algoritmo PI; já no caso 2, não conseguimos identificar a colisão. Os dados da distância do PQS são vistos em Lopes et al. (2010). Devido à indisponibilidade do código de acesso, não tivemos como fazer a comparação em tempo de execução e o cálculo da distância em todos os conjuntos de restrições.

Tabela 6: Testes de contato utilizando parâmetros $c^0 = (0.1, 0.1, 0.1, 0.1)$, $\beta = 0.15$ e $\alpha = (0.1, 0.1, 0.1, 0.1)$ no Método de Penalidade para o mesmo par de superquádricas.

restrições	Métodos	$d(\mathbf{p}, \mathbf{q})$	tempo[ms]	iterações
6	PI	1.768108	1.127	14
	MP	1.755794	692	5
6,8	PI	1.787740	1.080	6
	MP	1.755796	699	5
6,8,10	PI	1.768108	2.061	116
	MP	1.755798	695	5
6,8,11	PI	1.768108	2.214	128
	MP	1.755798	694	5
6,9	PI	1.768108	1.128	15
	MP	1.755796	693	5
6,9,10	PI	1.768108	1.136	18
	MP	1.755797	695	5
6,9,11	PI	1.768108	1.179	19
	MP	1.755797	691	5
6,10,11	PI	1.768108	1.152	15
	MP	1.755797	698	5

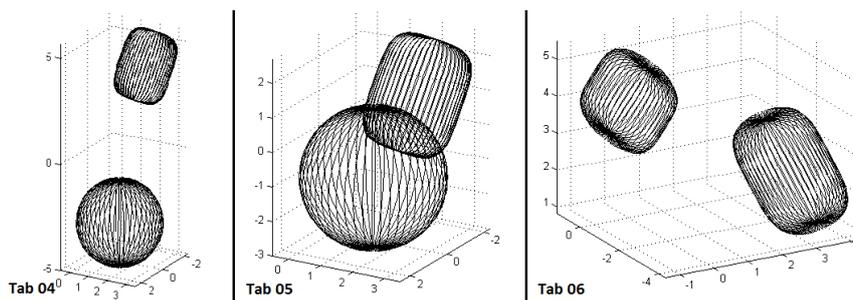


Figure 5: Pares de superquádricas dos casos das Tabelas 04, 05 e 06.

Até o momento, foram apresentados apenas casos mostrados por Lopes et al. (2010). As Tabelas 4, 5 e 6 referem-se a 3 novos testes, cujas propriedades dos objetos são informadas na Tabela 3, e visualizadas na Figura 5.

Os resultados da Tabela 4 possibilitam identificar as proximidades dos valores encontrados nos métodos comparados, os algoritmos convergiram encontrando uma distância entre as superquádricas. Em todos os casos o tempo e o número de iterações no MP foi menor. Na Tabela 5, o algoritmo proposto continua retornando bons resultados. O tempo de resposta é menor que o tempo retornado pelo PI e o MP convergiu encontrando contato em todas as linhas, não acontecendo o mesmo com o algoritmo comparado, que encontrou colisão apenas na metade dos testes.

Os resultados encontrados na Tabela 6 são bastante semelhantes aos apresentados na Tabela 4, ambos tiveram o mesmo comportamento, encontrando uma distância entre os corpos, mas os valores da distância mudam a cada conjunto de restrições, não obtendo precisão no cálculo. Mas podemos observar que o MP continua tendo um bom ganho de tempo e iterações.

6. Considerações Finais

Neste trabalho, apresentou-se a implementação de um algoritmo, baseado no método de Penalidades, como solução ao problema de encontrar a distância entre duas superquádricas convexas. Por meio destes estudos de caso, observou-se o comportamento do método proposto e comparou-se o seu desempenho com um método citado na literatura. Os dados apresentados demonstram a boa performance do método proposto, principalmente em termos de tempo de resposta.

Como principais vantagens em relação aos algoritmos comparados, nossa abordagem possui menor tempo de resposta e menor número de iterações, encontrando valores aceitáveis para o cálculo da distância. A nossa aproximação foi testada com 9 cenários diferentes, demonstrando resultados bastante promissores que colocam o método estudado como uma alternativa viável para a detecção de contato entre objetos convexas.

O incremento dos experimentos numéricos e o aprofundamento da análise dos resultados obtidos são tema de nossa pesquisa em curso.

Referências

- Bardinet, E., Cohen, L.D., Ayach, N.** (1995), Superquadrics and free-form deformations: a global model to fit and track 3D medical data, *Lecture Notes in Computer Science*, 905, 319-326.
- Barr, A. H.** (1981), Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Applications*, 1(1), 11-23.
- Bazarra, M., Sherali, H. e Shetty, C.** (1993), Nonlinear programming, theory and algorithms, Wiley, Hoboken, New Jersey.
- Buckley, C.E.** (1989), A foundation for the flexible-trajectory approach to numeric path planning, *International Journal of Robotics Research*, 8(3), 44-64.
- Choi, Y.K., Chang, J.W., Wang, W., Kim, M.S., Elber, G.** (2009), Continuous collision detection for ellipsoids., *IEEE Trans. Vis. Comput. Graph.*, 15(2), 311-325.
- Gardiner, M.** (1965), The superellipse: A curve that lie between the ellipse and the rectangle, *Sci. Amer.*, 213, 222-234.
- Li, C.L.** (2007), Part segmentation by superquadric fitting - a new approach towards automatic design of cooling system for plastic injection mould, *Int. J. Adv. Manuf. Technol.*, 35, 102-114.
- Lopes, D.S., Silva, M.T., Ambrósio, J.A. e Flores, P.** (2010), A mathematical framework for rigid contact detection between quadrics and superquadrics surfaces, *Multi-body Syst Dyn*, 255-280.
- Pentland, A. P.** (1986), Perceptual organization and the representation of natural form, *Artificial Intelligence*, 28(2), 293-331.
- Pombo, J., Ambrósio J. e Silva, M.** (2007), A new wheel-rail contact model for railway dynamics, *Vehicle System Dynamic*, 45(2), 165-189.
- Portal, R.J.F., Dias, J.M.P e Sousa, L.A.G.** (2010), Contact detection between convex superquadric surfaces, *The Archive of Mechanical Engineering*, LVII, 165-186.
- Solina, F., Bajcsy, R.** (1990), Recovery of parametric models from range images: the case for superquadrics with global deformations, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 131-147.
- Terzopoulos, D., Metaxas, D.** (1991), Dynamic 3D models with local and global deformations: deformable superquadrics, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, 703-714.