

**PROGRAMA PARA DETERMINAÇÃO DE CONTORNOS  
DE OBJETOS A PARTIR DE SUAS PROJEÇÕES  
PROGRAMA DE SIMULAÇÃO DE DADOS DE PROJEÇÃO**

Maria Luiza Reis  
Nilson Costa Roberty  
Virginia Reis Crispin  
PEN/COPPE/Universidade Federal  
do Rio de Janeiro  
CP 68509, 21945 - Rio de Janeiro

**SUMARIO:**

Este programa foi elaborado com o objetivo de desenvolver uma técnica mais eficiente e precisa na determinação de contornos de objetos dispostos separadamente a partir de projeções ao redor deste conjunto.

Como resultado dessa análise, que usa todas as projeções disponíveis conjuntamente, tem-se os contornos dos objetos descritos como figuras poligonais convexas.

As aplicações desta técnica, são aquelas que envolvem problemas de identificação bidimensionais a partir de projeções tomadas em diversos ângulos, tais como visualização da forma de objetos a partir de sinais de radar, identificação de desvios de posição e forma originais de tubulações ou de varretas em um elemento combustível de um reator nuclear empregando radiação.

Dentro desse mesmo programa, existe a opção de simulação de dados de projeção de cilindros dispostos no interior de um domínio definido pelo usuário.

Os contornos dos objetos são apresentados na tela do monitor sobre os desenhos dos objetos simulados. No canto inferior da tela, também é apresentado um gráfico da curva de projeção para cada ângulo.

O programa foi escrito em linguagem MS/Quick-BASIC versão 4.0 e foi executado em PC/AT com placa EGA.

## PROGRAMA DE IDENTIFICAÇÃO DE CONTORNOS:

O objetivo deste programa é o de fornecer o contorno de objetos usando para isso as projeções tomadas ao redor do domínio em ângulos diversos. Ainda que esses dados contenham informação de densidade dos objetos, o programa só determinará o seu contorno sem assinalar diferenças no interior das peças.

O princípio básico deste método é o fato de que onde não há objeto o valor da projeção é nulo. Este fato permite que cada projeção seja dividida em faixas onde aparecem sinais dos objetos. Os contornos que encontramos são os limites das interseções das faixas dos diversos ângulos.

Os dados de projeção que são lidos no início do programa são valores obtidos da discretização da imagem das peças tomadas a cada ângulo.

Para melhor compreensão do resultado, junto com a representação dos contornos dos objetos, o usuário tem a opção de ver as curvas de cada projeção.

As regiões de interseção das faixas não vazias de cada ângulo são encontradas através de uma busca na qual é feito um teste onde se verifica se um conjunto de faixas de ângulos diferentes realmente se interceptam. Essa verificação é realizada por uma subrotina que chamamos poligonal que, a partir de parâmetros das retas que limitam as faixas, determina os vértices ordenados da figura poligonal encontrada. A própria subrotina poligonal traça os lados da figura obtida na tela.

Como já foi dito os testes são feitos para um conjunto de faixas combinadas. Todas as combinações que são possíveis devem ser testadas e são representadas por um vetor. Um exemplo de combinação é mostrada abaixo.

$$\vec{v} = (b_1, b_2, b_3, b_4) = (4, 2, 1, 1)$$

Esse vetor representa a permutação de quatro faixas que são: a quarta faixa do primeiro ângulo, a segunda do segundo ângulo e a primeira tanto do terceiro quanto no quarto ângulo.

O vetor mostrado é preenchido de forma que a cada passagem o número de combinações testadas possa ser escrito como:

$$N_{\text{comb.}} = \sum_{i=1}^J b_i \cdot f_i^{i-1}$$

onde J é o número de projeções

$f_i$  é o número de faixas na projeção i

Numericamente, isso pode ser obtido de forma simples como é mostrado abaixo em uma subrotina em BASIC.

#### PERMUTAÇÕES:

```

FOR i = 1 TO J
  IF b(i) = f(i) THEN
    b(i) = 0
  ELSE
    b(i) = b(i) + 1
  RETURN
END IF
NEXT i
RETURN

```

#### POLIGONAL:

A subrotina descrita aqui constitui a peça chave para a solução do problema de reconstrução de contorno usado nesse programa.

A subrotina poligonal tem como parâmetros de entrada as retas que definem o contorno numeradas de 1 até 2. J no sentido horário, de modo que retas paralelas tem numeração iguais a n e n+J, onde a reta n é a de maior valor ao longo da reta orientada que passa pela origem e que é perpendicular ao ângulo que define a vista. Um terceiro parâmetro (pois a reta contribui com dois: ângulo e distância) é o número total de projeções J.

Inicialmente, é feita uma busca para definir a reta que servirá para compor o primeiro lado da poligonal. Nesta busca, calculam-se as interseções entre esta e todas as outras retas até encontrar uma interseção que pertença à região definida pelas faixas assim: região poligonal =  $\{(x,y) \in \mathbb{R}^2 \mid n_k < x \cdot \cos \theta_k < m_k, \forall k \ 1 < k < J\}$ .

onde  $n_k$  é o limite mínimo da faixa da vista  $k$   
 $\theta_k$  é o ângulo da vista  $k$   
 $m_k$  é o limite máximo da faixa da vista  $k$

O primeiro vértice pode não ser encontrado na reta de número um. Assim, esta etapa deve efetuar o teste para todas as retas até o primeiro vértice ser encontrado. A reta testada que contiver este ponto é chamada de primeira reta e é ela que vai determinar o fechamento da poligonal, quando for encontrado um segundo vértice sobre ela.

A segunda etapa se inicia chamando a reta com a qual a primeira se interceptou, da reta testada e reiniciando a busca agora deixando de testar a primeira reta que também passa a se chamar reta anterior. Encontrado um outro vértice, esta segunda etapa pára, passando a reta testada a se chamar reta anterior e a outro que contém o segundo vértice da reta testada passa a ser chamada de reta testada continuando o ciclo até que, como já foi dito, o segundo vértice da primeira reta seja encontrado.

#### PROGRAMA DE SIMULAÇÃO DE DADOS:

Esse programa pode ser chamado diretamente do Programa de Identificação de Contornos. Ele desenha na tela do monitor os círculos, representando a seção transversal de um cilindro, que são definidos pelo usuário através da posição do centro, do raio e da densidade - estes dados também servem para programas de reconstrução de densidade.

Depois que os círculos são desenhados, o programa aguarda um comando que confirme ou não o desenho feito. Caso não seja confirmado, o usuário pode redefinir os dados que considerar incorretos.

Após a confirmação, o programa calcula os dados de projeção para cada ângulo, segundo a discretização desejada. Isso é feito dividindo cada círculo em áreas pertencentes a diferentes segmentos de discretização que são multiplicadas pela densidade e somadas ao valor de projeção já atribuído ao segmento correspondente.

#### CONCLUSÕES:

A qualidade de definição do contorno de um objeto depende basicamente de dois fatores: quanto e quais são os ângulos de projeção. Quando existe

apenas um objeto, tem-se que quanto maior o número de projeções, melhor é a definição deste, pois ele é envolvido por uma poligonal convexa mais bem ajustada. Entretanto, quando há diversas peças em uma mesma imagem, a escolha dos ângulos usados é da maior importância pois a disposição dos objetos no plano pode favorecer determinados ângulos no trabalho de identificação. Isto se deve ao que chamamos de sombreamento, que leva ao aparecimento de contornos onde não há objeto algum.

A figura 2 mostra os contornos encontrados quando usadas quatro projeções equiespaçadas. Na figura 3 os mesmos objetos são envolvidos usando seis projeções. Pode ser observado que a ausência dos ângulos de  $45^{\circ}$  e  $135^{\circ}$  provocou o aparecimento de um sombreamento mais intenso.

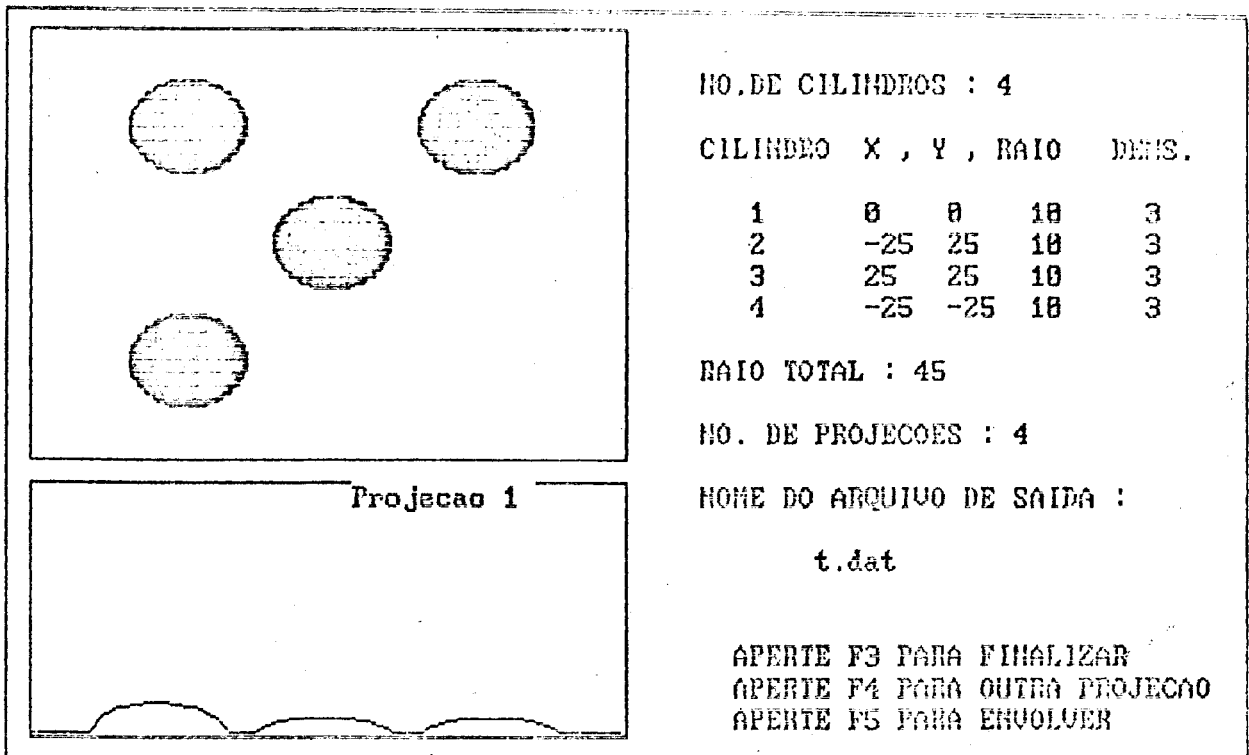


Figura 1

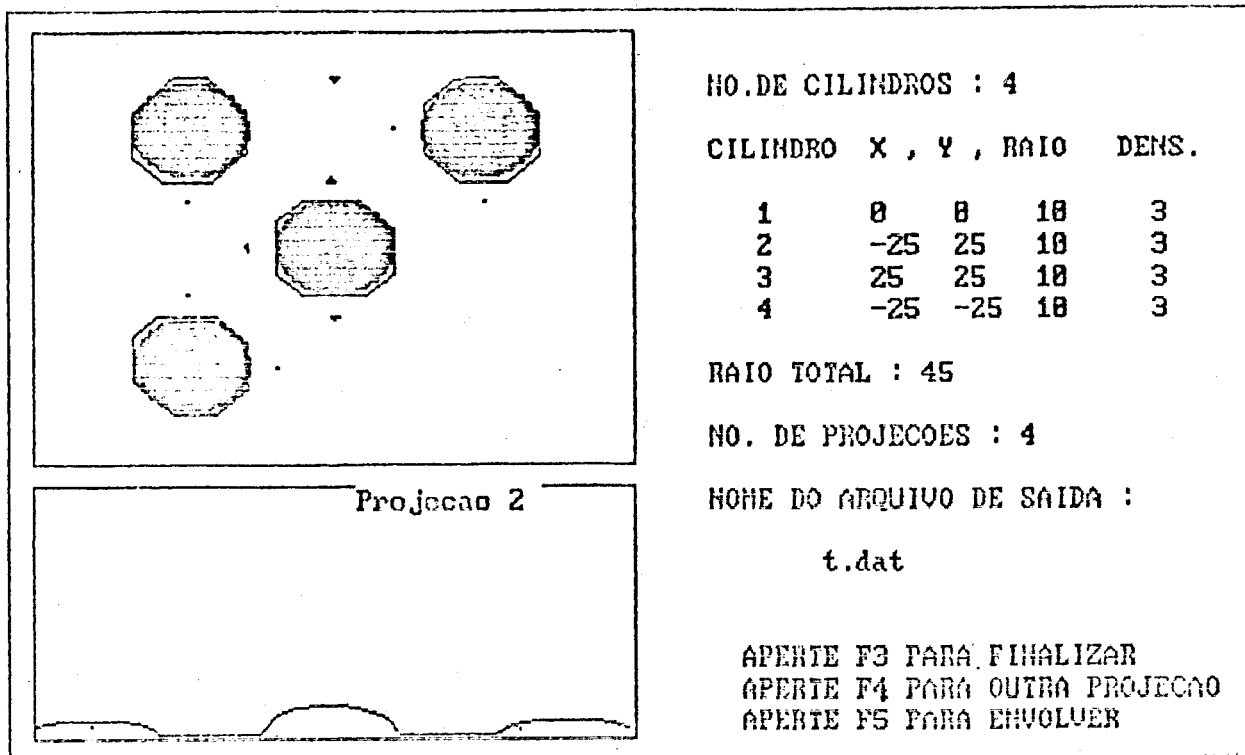


Figura 2

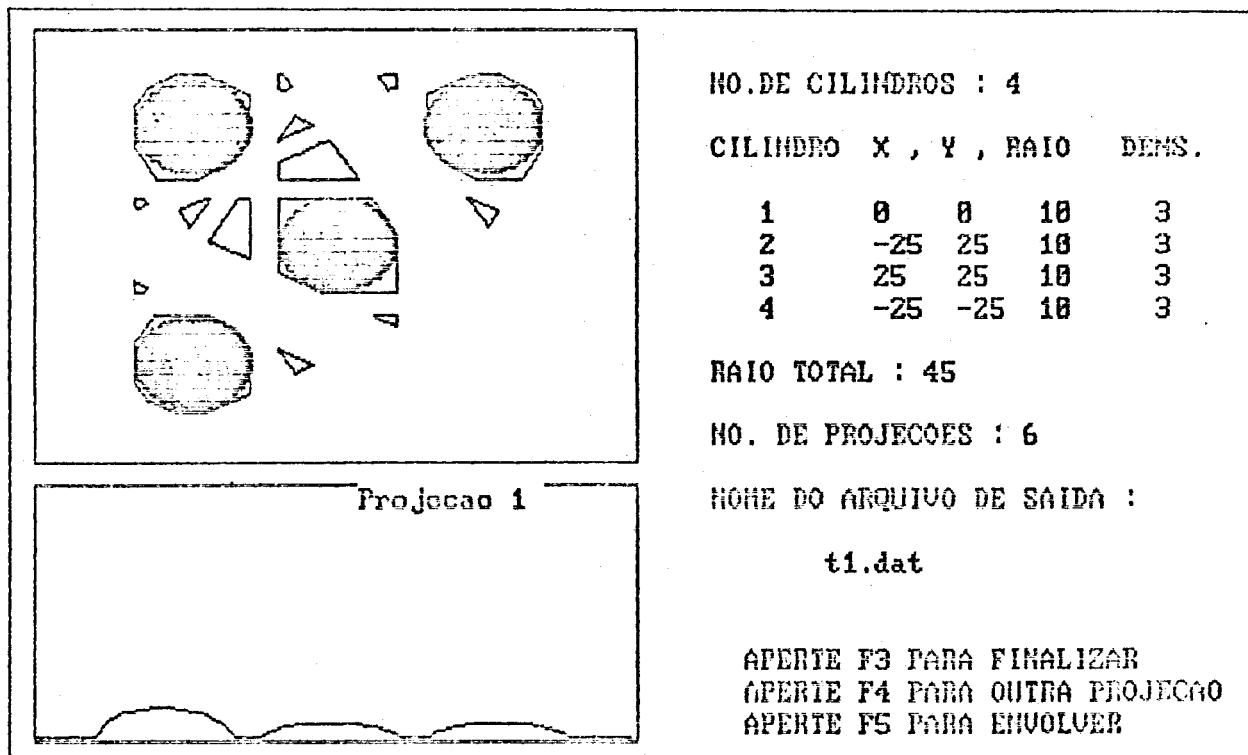


Figura 3